# HARNESSSHAP: EXPLAINING LLM AGENT TRAJECTORIES WITH TASK-ADAPTIVE SHAPLEY ATTRIBUTION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

When a large language model (LLM) agent succeeds or fails on a multi-step task, developers need to identify which parts of the trajectory contributed to the outcome. Existing attribution methods often rely on fixed judgment prompts, which may miss task-specific success criteria, and naive suppression can leave downstream actions dependent on removed steps. We introduce HarnessSHAP, a trajectory attribution method that learns a task-adaptive harness for a frozen LLM judge. The harness contains role-analysis and counterfactual-judgment prompts that estimate outcomes after dependency-consistent suppression while allowing downstream replanning. HarnessSHAP then applies Shapley attribution over dependency-valid orderings to assign contributions to evidence, reasoning steps, tool calls, and actions. The harness is optimized by XAI-Grad, a textual prompt optimizer that uses replay-derived endpoint and marginal errors to update the prompt pair without changing model parameters. We evaluate HarnessSHAP against a Fixed harness and Generic TextGrad on API-Bank, HotpotQA, ToolSandbox, and Who&When. XAI-Grad improves counterfactual outcome agreement and attribution quality across evaluated settings. On API-Bank, HotpotQA, and ToolSandbox, XAI-Grad achieves Spearman correlations of 0.74, 0.72, and 0.79, with endpoint MAE values of 0.103, 0.111, and 0.096, respectively. The ablation study shows that combining endpoint and marginal feedback reduces attribution error to 0.124. On Who&When, HarnessSHAP evaluates static failure localization separately from replay-based attribution.

## 1 INTRODUCTION

A final task score summarizes success without identifying the contributions of individual parts of the agent's recorded behavior. Large language model (LLM) agents produce trajectories, meaning event sequences containing observations, evidence, exposed reasoning, tool calls, tool results, and actions. We call a part of the record selected for analysis a trajectory unit. Attribution estimates each unit's contribution to the outcome. Reliable attribution can support failure analysis, audits, and targeted revision of agent behavior.

Existing methods address retrieved context (Cohen-Wang et al., 2024), tool use (Horovicz, 2025), annotated failure locations (Zhang et al., 2025), and outcomes after an intervention. We call a change to permitted agent behavior an intervention. Executing the agent under that change yields a counterfactual outcome, a process called replay. Static localization identifies annotated failure locations without executing an intervention, while direct replay can be expensive or unavailable. A language model with unchanged parameters predicts outcomes without environment replay. This model is the frozen judge for outcome prediction. Blocking selected recorded units while allowing the agent to regenerate later behavior defines suppression. A fixed judge prompt may misread task success or the role of a suppressed unit.

HarnessSHAP proposes task-adapted trajectory attribution by separating a fixed attribution layer from a learned external pair of analysis prompts. The fixed layer parses a trajectory and records which units require other units in a prerequisite graph. The layer applies suppression that respects these prerequisites. The layer stores independent outcome judgments for contexts, the sets of units exempt



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.

from suppression. The layer then averages the score changes from adding each unit over orders that respect the prerequisites. The method uses XAI-Grad, a textual optimizer for attribution, to optimize the two prompts while keeping the judge model fixed. The role prompt $p_{\text{role}}$ describes each unit's function, while the outcome prompt $p_{\text{judge}}$ predicts the outcome under a specified suppression context. Model parameters and the fixed attribution layer remain unchanged throughout prompt pair optimization and explanation.

HarnessSHAP uses a dependency-constrained trajectory attribution game with suppression and downstream replanning. Its fixed layer specifies trajectory decomposition, dependency rules, suppression semantics, the factual endpoint, context caching, and Faigle-Kern aggregation (Faigle & Kern, 1992). Within this fixed attribution game, HarnessSHAP contributes task-adaptive optimization of the external analysis harness and prompt pair.

Optimization supervised by replay and explanation without replay both use the recorded factual score when all units are exempt from suppression.

Our contributions are as follows.

- We present HarnessSHAP, a task-adaptive trajectory attribution framework that learns external role-analysis and counterfactual-judgment prompts while keeping the attribution game and judge model fixed.
- We develop XAI-Grad, which organizes replay-derived endpoint and marginal errors into prompt-specific feedback, alternates updates to the two prompts, and selects candidates on held-out trajectories.
- We evaluate HarnessSHAP against the Fixed harness and Generic TextGrad under a shared protocol, separating replay fidelity, execution validation, static failure localization, and cost.

## 2 Related Work

**Agent trajectories and failure attribution.** Agent frameworks, benchmarks, and explanation methods address tool use, evidence, and failure localization. ReAct is an agent framework that interleaves exposed reasoning with actions (Yao et al., 2023). API-Bank (Li et al., 2023) and ToolSandbox (Lu et al., 2025) are benchmarks for tool use. HotpotQA is a dataset for questions requiring multiple reasoning steps (Yang et al., 2018). Who&When is a benchmark for identifying the responsible agent and step in static logs with labeled failures (Zhang et al., 2025). AgenTracer is a system for locating failures, with an annotation pipeline, a trained tracer, and evidence for accuracy and downstream repair (Zhang et al., 2026). Localization labels identify failure sites, while HarnessSHAP estimates unit contributions under a fixed suppression policy.

**Attribution under structure.** Shapley attribution methods differ in the orders used to average each unit's marginal contribution, meaning the score change from adding that unit. Classical Shapley values (Shapley, 1953) average over all complete unit orders, called permutations (Lundberg & Lee, 2017). Faigle-Kern values use linear extensions of a prerequisite order, meaning complete unit orders in which prerequisites precede their dependents (Faigle & Kern, 1992). Asymmetric Shapley values (Frye et al., 2020) and causal Shapley values (Heskes et al., 2020) are related attribution methods that encode direction. ContextCite attributes generated text to its supplied context (Cohen-Wang et al., 2024). HarnessSHAP retains a fixed game and Faigle-Kern aggregation, while adapting context judgments through external prompts.

**Prompt optimization for trajectory attribution.** TextGrad optimizes textual variables through language-model feedback while keeping model parameters frozen (Yuksekgonul et al., 2025). Its computation graph connects those variables to update feedback. Within a fixed dependency-constrained game, HarnessSHAP specializes this feedback for trajectory attribution. XAI-Grad routes endpoint and marginal errors to alternating role-analysis and counterfactual-judgment prompt updates and selects candidates under a shared evaluation contract. The matched Generic TextGrad comparison evaluates this task-adaptive prompt optimization.

# 3 METHOD

## 3.1 TASK AND OUTPUT

HarnessSHAP assigns one contribution to each recorded unit in a trajectory produced by a fixed agent execution setup. The Who&When application evaluates HarnessSHAP for localization separately from estimating contributions to executed counterfactual outcomes. The HarnessSHAP game evaluates suppression and downstream regeneration using typed trajectory units and a fixed suppression policy. Let $x = (e_1, \ldots, e_\ell)$ denote that trajectory, with $\ell$ events indexed by position $j$ and $e_j$ denoting each event. The scoring rule $R$ maps a trajectory to its outcome score, with factual score $R(x) \in [0, 1]$. A deterministic, versioned parser maps $x$ to the unit set $\mathcal{U}_x = \{u_1, \ldots, u_n\}$, where $n$ counts units and $u_i$ denotes unit $i$. The parser also builds the prerequisite graph $D_x$, a directed graph without cycles that records dependencies between those units. Each unit retains a stable identifier, source event, and provenance linking the unit to its source. The output attribution vector $\widehat{\phi}_x$ contains contribution $\widehat{\phi}_i$ for each unit. The output also includes the graph, sampled orders, cached values, and selected prompt pair version.

The prompt pair $h = (p_{\text{role}}, p_{\text{judge}})$ contains the two text variables optimized for the attribution task. The role prompt $p_{\text{role}}$ describes the fixed units and graph in the role table $\Theta_x$. The outcome prompt $p_{\text{judge}}$ predicts the outcome of a suppression context using the recorded trajectory and that table. The judge weights, decoding configuration, parser, graph rules, suppression policy, context sampler, cache key, and aggregation remain fixed.

## 3.2 FIXED GAME WITH DEPENDENCY CONSTRAINTS

Parser details are provided in Appendix A.1.

A context $S$ is a set of recorded units exempt from suppression. A valid context includes all prerequisites of its units as recorded by the parser. An edge $(u_j, u_i) \in D_x$ requires that exemption of $u_i$ also exempts $u_j$. Sets containing every prerequisite of each included unit are closed under prerequisites. These sets form the valid context family $\mathcal{I}_x$

$$\mathcal{I}_x = \{S \subseteq \mathcal{U}_x : u_i \in S \wedge (u_j, u_i) \in D_x \Rightarrow u_j \in S\}. \tag{1}$$

A proper context $S \in \mathcal{I}_x$ exempts fewer than all recorded units. The intervention $\mathcal{H}_x(S)$ targets every unit in $\mathcal{U}_x \setminus S$. The intervention preserves the recorded events before the earliest target, called the factual prefix. It restores the environment at that point before regenerating the remaining trajectory. The controller regenerates the remaining trajectory, called the suffix, under the execution setup that produced the recorded trajectory. Exempted units need not recur unchanged during this regeneration. The controller must block the targeted call, result, or state effect rather than delete a completed log entry. At $S = \mathcal{U}_x$, the intervention is the identity on the recorded trajectory and launches no new replay.

Accepted replays define the reference outcome for a proper context under the fixed suppression policy. Let $X_x(S, \xi)$ be a replay attempt, where $\xi$ collects its randomness. Acceptance means a valid context, completed execution, a defined score, absent targets, and no reintroduction of equivalent suppressed content. The indicator $a_x(S, \xi)$ equals one for an accepted attempt and zero otherwise. Let $q_x(S)$ be the replay acceptance probability and $v_x(S)$ the mean score among accepted replays. With $\Pr$ denoting probability and $\mathbb{E}$ expectation over replay randomness, contexts with positive acceptance probability have

$$q_x(S) = \Pr[a_x(S, \xi) = 1], \qquad v_x(S) = \mathbb{E}[R(X_x(S, \xi)) \mid a_x(S, \xi) = 1]. \tag{2}$$

The reference game $w_x(S)$ uses the recorded factual score for the full context and the mean among accepted replays for proper contexts with positive acceptance probability.

$$w_x(S) = \begin{cases} R(x), & S = \mathcal{U}_x, \\ v_x(S), & S \subsetneq \mathcal{U}_x,\ q_x(S) > 0. \end{cases} \tag{3}$$

A fresh stochastic rerun does not replace the recorded factual score. If $q_x(S) = 0$, the reference game is undefined at that context. A finite collection of replay attempts, called a replay bank, with no accepted attempt does not establish that $q_x(S) = 0$.

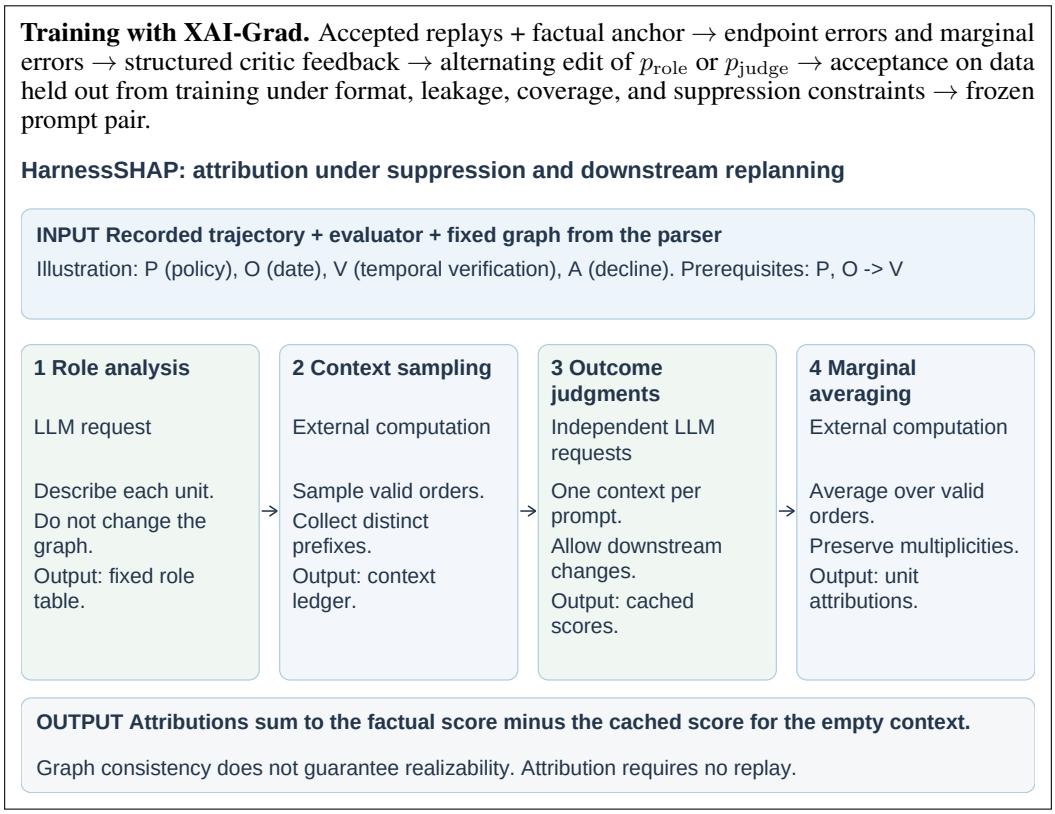


Figure 1: **Optimization and trajectory attribution.** The HarnessSHAP architecture separates optimization supervised by replay from explanation without replay. The lower panel uses the fixed attribution layer with the selected frozen prompts.

The factual endpoint is anchored in both games, and contexts exempt units from suppression without forcing the regenerated suffix. The refund trace is a constructed example with policy evidence P, an observation of the order date O, temporal verification V, and decline action A. Arrows indicate the reading order within each pipeline and the prerequisite relation P, O to V in the example. Background colors group stages without encoding a numerical quantity. LLM denotes large language model, while $p_{\text{role}}$ and $p_{\text{judge}}$ denote the role prompt and outcome prompt, respectively.

For a proper context, the reference estimate $\widetilde{w}_x(S)$ is the mean outcome of accepted attempts, with total and accepted attempt counts recorded. The estimate at the factual endpoint is fixed to the recorded outcome, $\widetilde{w}_x(\mathcal{U}_x) = R(x)$. A proper context with no accepted replay has a missing reference estimate and is never assigned zero. Replay records and adherence checks are detailed in Appendix A.2.

### 3.3 Prompt pair supervision

The outcome prompt predicts context outcomes from recorded information and the suppression specification, without executing an intervention. The role prompt first produces $\Theta_x$ once per trajectory for use in those predictions. The outcome prompt receives $x$, $\Theta_x$, the scoring rule $R$, the suppression specification, and one context $S$. The scoring rule is distinct from the observed score $R(x)$. The predicted score $\widehat{v}_x^h(S) \in [0,1]$ and a structured rationale form the prompt output for prompt pair $h$. These context predictions define the values of the judgment game. The factual endpoint is assigned the recorded score without prediction.

$$\widehat{v}_x^h(\mathcal{U}_x) := R(x). \tag{4}$$

Training and explanation use the same $\mathcal{H}_x$, and labels from a different suppression policy cannot supervise the prompt pair.

For a context with an available reference estimate, define the endpoint error $e_v(x, S)$ as the predicted score minus that estimate.

$$e_v(x, S) = \widehat{v}_x^h(S) - \widetilde{w}_x(S). \tag{5}$$

An adjacent pair $(S, S \cup \{u_i\})$ contains the contexts immediately before and after adding unit $u_i$ along a linear extension. Both contexts require reference values, including the factual score when the second context contains every recorded unit. The marginal error $e_\Delta(x, S, i)$ subtracts the reference score change from the predicted score change for that addition.

$$e_\Delta(x, S, i) = \left[\widehat{v}_x^h(S \cup \{u_i\}) - \widehat{v}_x^h(S)\right] - \left[\widetilde{w}_x(S \cup \{u_i\}) - \widetilde{w}_x(S)\right]. \tag{6}$$

The marginal error equals the difference between adjacent endpoint errors, so endpoint feedback and marginal feedback use identical replay labels.

$$e_\Delta(x, S, i) = e_v(x, S \cup \{u_i\}) - e_v(x, S). \tag{7}$$

A common prediction offset cancels between proper contexts, but the anchored endpoint always has $e_v(x, \mathcal{U}_x) = 0$.

Feedback separates errors in outcome judgment from errors in unit descriptions to direct updates to the relevant prompt. Endpoint feedback identifies overestimation or underestimation, the task condition that was misread, and any suppressed information used by the judge. Marginal feedback identifies differences between endpoint errors even when one endpoint appears accurate alone. Feedback for $p_{\text{judge}}$ changes how the outcome is judged from $\Theta_x$, the recorded trace $x$, and the suppression specification for $S$. Feedback for $p_{\text{role}}$ groups marginal errors by unit and corrects missing or ambiguous descriptions of each unit's function. The feedback cannot add units, merge units, or edit graph edges.

### 3.4 XAI-GRAD PROMPT OPTIMIZATION

Lineage splits, reference support, candidate selection, stopping rules, and budgets are detailed in Appendix A.

XAI-Grad adapts TextGrad prompt optimization to attribution under a narrower set of evaluation rules. The method in Section 3 groups errors, alternates prompt updates, and selects candidates under common replay, coverage, and leakage rules. These differences require direct ablation against Generic TextGrad under matched optimization limits. XAI-Grad alternates edits to the role prompt and outcome prompt while keeping the other prompt fixed in each round. Optimization starts from initial prompt pair $h_0$, with round index $t$ beginning at zero and increasing after each round. Even rounds propose edits to $p_{\text{role}}$, while odd rounds propose edits to $p_{\text{judge}}$ using the relevant error records. The editor produces versioned candidates for the scheduled prompt update. Textual feedback alone cannot justify accepting a candidate because the feedback can be wrong.

Candidate selection requires valid outputs and improved validation loss on a fixed population with reference support. Let $\rho$ denote the loss applied to an error and $\lambda$ the weight of the marginal loss. The endpoint loss $L_v(h)$ averages transformed endpoint errors, while the marginal loss $L_\Delta(h)$ averages transformed marginal errors. The proposed objective $L(h)$ combines these losses for prompt pair $h$.

$$L(h) = L_v(h) + \lambda L_\Delta(h), \qquad L_v = \operatorname{mean} \rho(e_v), \quad L_\Delta = \operatorname{mean} \rho(e_\Delta). \tag{8}$$

The endpoint loss uses supported proper contexts, while the marginal loss uses adjacent occurrences with both reference values and retains their multiplicities within each trajectory.

The order multiset $\Pi$ is the collection of retained orders, with duplicates and their multiplicities preserved.

The context ledger $\mathcal{C}$ stores distinct sets of units preceding each position in retained orders. These sets are the order prefixes, whose context judgments can be reused.

### 3.5 EXPLANATION AND AGGREGATION

Explanation averages contributions over linear extensions so that every unit enters a context after its recorded prerequisites. Let $\mathcal{L}(D_x)$ denote the set of linear extensions of $D_x$, and let $\pi \in \mathcal{L}(D_x)$ denote one such complete order. The predecessor set $\operatorname{Pre}_\pi(i)$ contains the units before $u_i$ in $\pi$, and both contexts adjacent to its addition are closed under prerequisites.

**Algorithm 1 HarnessSHAP training and explanation.**

**Input:** Lineage-separated training/validation sets $T, V$; replay bank $B$; initial harness $h_0$
Fixed objective, support/budget/stopping rules; explanation trajectory $x$
**Training**
Freeze(parser, graphs, interventions, evaluator, judge settings, attribution rule)
$\mathcal{S} \leftarrow \mathrm{FixReferenceSupport}(B)$; $\mathrm{AnchorFactualScores}(B)$
$h \leftarrow h_0$; $t \leftarrow 0$
**repeat**
  $(\Theta, \widehat{v}) \leftarrow \mathrm{PredictRolesAndValues}(T, h, \mathcal{S})$
  $(e_v, e_\Delta) \leftarrow \mathrm{EndpointAndMarginalErrors}(\widehat{v}, B, \mathcal{S})$
  $p \leftarrow p_{\mathrm{role}}$ if $t$ is even, else $p_{\mathrm{judge}}$
  $f \leftarrow \mathrm{RouteTrainingFeedback}(e_v, e_\Delta, p)$
  $H \leftarrow \mathrm{VersionedEdits}(h, p, f, \mathrm{freeze_other} = \mathrm{True})$
  $\mathrm{ChargeAllCalls}(\mathrm{shared_budget}, \mathrm{audits} = \mathrm{True}, \mathrm{validation} = \mathrm{True})$
  $E \leftarrow \mathrm{RejectContractFailures}(H)$
  $\ell \leftarrow \mathrm{EvaluateOnFixedSupport}(E, V, \mathcal{S})$
  $h' \leftarrow \mathrm{BestEligibleCandidate}(E, \ell, \mathrm{require_all_constraints} = \mathrm{True})$
  $h \leftarrow h'$ **if** $h'$ exists and improves the objective; **else** retain $h$
  $\mathrm{LogDecision}()$; $t \leftarrow t + 1$
**until** $\mathrm{RegisteredStop}()$
$h^* \leftarrow \mathrm{Freeze}(h, \text{audit trail})$
**Explanation**
$(\mathcal{U}_x, D_x) \leftarrow \mathrm{Parse}(x)$; $\Theta_x \leftarrow \mathrm{RoleTable}(x, h^*)$
$\Pi \leftarrow \mathrm{EnumerateOrUniformlySample}(\mathcal{L}(D_x), \text{fixed budget})$
$\mathcal{C} \leftarrow \mathrm{DistinctPrefixes}(\Pi)$; $\mathrm{cache}[\mathcal{U}_x] \leftarrow R(x)$
**for** each $S \in \mathcal{C} \setminus \{\mathcal{U}_x\}$ **do**
  $\mathrm{cache}[S] \leftarrow \mathrm{JudgeOncePerCacheRealization}(x, \Theta_x, S, h^*)$
**end for**
$\widehat{\phi}_x \leftarrow \mathrm{Aggregate}(\Pi, \mathrm{cache}, \mathrm{rule} = \text{Eq. 9}, \mathrm{preserve_multiplicity} = \mathrm{True})$
**return** $\widehat{\phi}_x, D_x, \Theta_x, \Pi, \mathrm{cache}, \mathrm{Version}(h^*)$

With the selected prompt pair $h^*$ frozen, estimate each unit's contribution by averaging its score change across retained orders.

$$\widehat{\phi}_i = \frac{1}{|\Pi|} \sum_{\pi \in \Pi} \left[ \widehat{v}_x^{h^*}(\mathrm{Pre}_\pi(i) \cup \{u_i\}) - \widehat{v}_x^{h^*}(\mathrm{Pre}_\pi(i)) \right]. \tag{9}$$

The empty context $\emptyset$ exempts no recorded units from suppression. Shared cached endpoints make intermediate score changes cancel within each order, so the contributions sum to $\sum_i \widehat{\phi}_i = R(x) - \widehat{v}_x^{h^*}(\emptyset)$. No rescaling is applied after aggregation, and the cancellation identity does not establish agreement with replay. The mean judgment game averages context scores over role and judgment randomness. Assume a fixed order budget and uniform extension sampling, with role and judgment randomness independent of extension sampling. With this independence and uniform sampling, the expected estimate targets the Faigle-Kern value of the mean judgment game.

Appendix B gives the sampler and cache details and the attribution-error bound. Appendix C gives the worked marginal example.

## 4 Experimental Evaluation

We evaluate HarnessSHAP against the Fixed harness and Generic TextGrad under a shared protocol. Tables 1, 2, 3, and 4 present benchmark comparisons, value-fidelity results, ablations, and validity diagnostics. Execution validation is assessed separately. Appendix A lists the protocol settings.

### 4.1 Questions and comparison contract

The evaluation compares outcome judgments, unit contributions, feedback components, and costs under a shared evaluation contract. Where replay support is established, we compare the value fidelity of the Fixed harness, Generic TextGrad, and XAI-Grad against accepted replay outcomes, using a common attribution reference. The ablations examine endpoint feedback, marginal feedback, prompt

roles, and adherence feedback. Cost accounting covers the replay, optimization, and explanation phases.

The comparison of the Fixed harness, Generic TextGrad, and XAI-Grad uses common trajectories and attribution rules to isolate feedback organization and routing. All conditions share trajectory splits, the replay bank, judge, $\mathcal{H}_x$, context pairs, cache rules, and Faigle-Kern aggregation. The Fixed harness uses the initial prompts without optimization throughout the comparison. Generic TextGrad updates the same role and outcome prompts through general textual feedback. Generic TextGrad receives the same replay labels, adjacent pairs, scalar objective, validation access, and contract checks as XAI-Grad.

### 4.2 Benchmarks, splits, and metrics

The evaluation includes API-Bank, HotpotQA, ToolSandbox, and Who&When, with three distinct uses.

**Replay-based evaluation.** Replay fidelity requires a controller that can restore the relevant state, enforce suppression, regenerate the suffix, and obtain an accepted outcome. API-Bank supplies executable application programming interfaces (APIs) and evaluators. HotpotQA supplies instances and a task evaluator. Tables 1 and 2 evaluate replay fidelity on supported contexts. These measurements do not establish reset or suppression support outside the evaluated contexts. The same execution requirements apply to any ToolSandbox replay comparison.

**Execution validation.** ToolSandbox supplies a stateful environment and an official evaluator. The separate engineering evidence verifies reset, a configured suppression cascade, state restoration, and scoring for one scripted scenario (Appendix F). This establishes adapter behavior for that scenario, not LLM replanning, judge adherence, prompt optimization, or benchmark-level attribution fidelity.

**Static localization.** Who&When supplies labeled failure logs for localization of agents and steps. It supplies no executed counterfactual reference and cannot train or validate replay losses. Its prompt-pair source, selection route, and unit-to-agent/step mapping remain application parameters in Appendix A. Static localization accuracy cannot substitute for replay fidelity.

The following fidelity metrics apply only to supported replay contexts.

Endpoint MAE averages $|e_v|$ over supported proper contexts, excluding the assigned factual anchor. Spearman correlation measures agreement between predicted and reference ranks on the same supported contexts. Marginal MAE averages $|e_\Delta|$ over supported adjacent occurrences, preserving their multiplicity within each trajectory.

Additional metric definitions and uncertainty rules are listed in Appendix A.6.

### 4.3 Benchmark Results

Table 1 compares the Fixed harness, Generic TextGrad, and XAI-Grad across benchmarks under the shared protocol.

On API-Bank, XAI-Grad achieves Drop 0.42, Valid 0.82, Top-3 overlap 0.70, and MAE 0.103. The Fixed harness achieves Drop 0.34 and MAE 0.151. On HotpotQA, XAI-Grad achieves MAE 0.111, compared with 0.163 for the Fixed harness. On ToolSandbox, XAI-Grad achieves MAE 0.096 and Spearman correlation 0.79. This benchmark comparison is separate from the scripted execution validation.

On Who&When, the Fixed harness, Generic TextGrad, and XAI-Grad achieve localization accuracies of 0.55, 0.61, and 0.67, respectively. These results evaluate static localization only.

### 4.4 Value, validity, and cost diagnostics

To examine agreement with replay references, Table 2 compares rank correlations and prediction errors across prompt pairs. Additional validity diagnostics and cost accounting appear in Appendix D.

Table 1: **Comparison across benchmarks.** Fixed harness, Generic TextGrad, and XAI-Grad under the shared evaluation contract.

| Benchmark | Method | Drop | Valid | Top-3 | MAE | Loc. | C/R/T |
|---|---|---|---|---|---|---|---|
| API-Bank | Fixed harness | 0.34 | 0.69 | 0.58 | 0.151 | N/A | 4.0/0/12.5 |
| | Generic TextGrad | 0.38 | 0.75 | 0.64 | 0.126 | N/A | 18.4/0/24.7 |
| | XAI-Grad | 0.42 | 0.82 | 0.70 | 0.103 | N/A | 19.1/0/27.9 |
| HotpotQA | Fixed harness | 0.31 | 0.66 | 0.55 | 0.163 | N/A | 3.8/0/11.9 |
| | Generic TextGrad | 0.35 | 0.73 | 0.61 | 0.133 | N/A | 17.9/0/23.5 |
| | XAI-Grad | 0.39 | 0.79 | 0.67 | 0.111 | N/A | 18.7/0/26.8 |
| ToolSandbox | Fixed harness | 0.36 | 0.71 | 0.62 | 0.142 | N/A | 4.5/0/13.6 |
| | Generic TextGrad | 0.40 | 0.77 | 0.68 | 0.120 | N/A | 20.2/0/29.1 |
| | XAI-Grad | 0.44 | 0.84 | 0.73 | 0.096 | N/A | 21.0/0/31.4 |
| Who&When | Fixed harness | N/A | N/A | N/A | N/A | 0.55 | 4.2/0/12.1 |
| | Generic TextGrad | N/A | N/A | N/A | N/A | 0.61 | 18.9/0/25.3 |
| | XAI-Grad | N/A | N/A | N/A | N/A | 0.67 | 19.8/0/28.7 |

NE means not evaluated, and N/A means not applicable to the row. Localization may be outside a row's target, and replay metrics are unavailable on static logs. Loc. denotes localization accuracy, and MAE denotes mean absolute error. Top-3 denotes overlap of the three selected units. C/R/T denotes calls, replay attempts, and elapsed seconds per trajectory. Costs in this table refer to explanation under the shared evaluation contract.

Table 2: **Value fidelity.** Context predictions against replay references. Coverage, execution checks, and judge adherence: Appendix D.

| Diagnostic | API-Bank | HotpotQA | ToolSandbox |
|---|---|---|---|
| Fixed harness Spearman | 0.61 | 0.58 | 0.64 |
| Generic TextGrad Spearman | 0.68 | 0.65 | 0.71 |
| XAI-Grad Spearman | 0.74 | 0.72 | 0.79 |
| Fixed harness endpoint MAE | 0.151 | 0.163 | 0.142 |
| Generic TextGrad endpoint MAE | 0.126 | 0.133 | 0.120 |
| XAI-Grad endpoint MAE | 0.103 | 0.111 | 0.096 |
| Fixed harness marginal MAE | 0.176 | 0.184 | 0.169 |
| Generic TextGrad marginal MAE | 0.143 | 0.151 | 0.136 |
| XAI-Grad marginal MAE | 0.118 | 0.126 | 0.109 |

MAE means mean absolute error. Factual anchors are excluded from endpoint MAE and Spearman correlation under the shared reference bank.

On API-Bank, XAI-Grad achieves Spearman correlation 0.74 and endpoint MAE 0.103. The corresponding Spearman correlations and endpoint MAEs are 0.72 and 0.111 on HotpotQA, and 0.79 and 0.096 on ToolSandbox.

### 4.5 Ablation Results

Table 3 compares feedback components and prompt-update conditions under the shared evaluation protocol. Detailed conditions and falsification criteria appear in Appendix E.

Endpoint-only feedback yields lower endpoint error than marginal-only feedback, whereas marginal-only feedback yields lower marginal and attribution errors. Combining endpoint and marginal feedback yields attribution error 0.136, compared with 0.124 for full XAI-Grad.

Full XAI-Grad has a misuse rate of 5.2%, compared with 11.2% for the variant without adherence feedback, with similar prediction errors.

Table 3: **Feedback and prompt ablations.** All conditions share a replay bank, optimization limits, and context coverage. Misuse is the failure fraction among audited judgments, with audit coverage reported separately.

| Condition | Endpoint MAE ↓ | Marginal MAE ↓ | Attribution error ↓ | Coverage ↑ | Misuse rate ↓ |
|---|---|---|---|---|---|
| Fixed harness | 0.151 | 0.176 | 0.214 | 78.4% | 15.6% |
| Generic TextGrad | 0.126 | 0.151 | 0.181 | 78.4% | 10.4% |
| Endpoint feedback only | 0.115 | 0.145 | 0.166 | 78.4% | 9.1% |
| Marginal feedback only | 0.121 | 0.132 | 0.154 | 78.4% | 9.8% |
| Endpoint and marginal† | 0.108 | 0.119 | 0.136 | 78.4% | 7.5% |
| Full minus adherence feedback | 0.105 | 0.117 | 0.131 | 78.4% | 11.2% |
| Optimize $p_{\mathrm{role}}$ only | 0.111 | 0.125 | 0.143 | 78.4% | 8.6% |
| Optimize $p_{\mathrm{judge}}$ only | 0.109 | 0.122 | 0.139 | 78.4% | 8.1% |
| XAI-Grad | 0.103 | 0.118 | 0.124 | 78.4% | 5.2% |

MAE denotes mean absolute error, and NE denotes not evaluated. Coverage denotes covered contexts divided by attempted contexts. The arrows ↓ and ↑ indicate lower and higher preferred values, respectively. The row marked † combines endpoint and marginal feedback.

## 5 Conclusion

HarnessSHAP combines task-specific prompt optimization with a fixed judge model and attribution game. XAI-Grad uses feedback from accepted replays and selects prompt candidates on held-out trajectories. Explanations then use cached context judgments.

Across the evaluated settings, XAI-Grad achieves lower endpoint and attribution errors than the Fixed harness and Generic TextGrad baselines. XAI-Grad achieves endpoint MAE 0.103, 0.111, and 0.096 on API-Bank, HotpotQA, and ToolSandbox, with corresponding Spearman correlations 0.74, 0.72, and 0.79, and attribution error 0.124 in the ablation study.

The attribution error bound applies to matched order ledgers and finite reference values under the defined conditions. Reproducing these results requires the replay banks, frozen configurations, and run records.

## 6 Limitations

The explanation inherits every error in trajectory parsing and dependency reconstruction. Some reasoning operations may lack an independently verifiable suppression instruction. Equivalent information can return through paraphrase or hidden state. Environment resets and evaluators can also be incomplete. Passing an execution check does not prove judge adherence.

Finite accepted replays estimate a conditional outcome with uncertainty. Low acceptance can create missing contexts and restrict the evaluation population. The learned prompt pair may overfit a task, agent execution setup, model family, or replay controller despite splits separated by source lineage. A telescoping attribution sum is an algebraic property and does not establish causal fidelity.

Replay validity, judge adherence, environment reset behavior, and execution support must be assessed separately for each benchmark.

The reported improvements are limited to the evaluated settings and do not establish reduced computational cost or generalization beyond the evaluated benchmarks.

The separate E0 engineering evidence covers pure attribution logic and one scripted ToolSandbox scenario. Those engineering checks do not test language model replanning, judge adherence, XAI-Grad training, or performance across benchmarks.

## AI USE STATEMENT

We used large language models to aid and polish the writing of this paper. The authors are responsible for all content of the paper, including the method, the experiments, the results, and the citations.

## REFERENCES


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, pp. 95764–95807, 2024. doi: 10.52202/079017-3035.

Ulrich Faigle and Walter Kern. The Shapley value for cooperative games under precedence constraints. *International Journal of Game Theory*, 21(3):249–266, 1992. doi: 10.1007/BF01258278.

Christopher Frye, Colin Rowat, and Ilya Feige. Asymmetric Shapley values: incorporating causal knowledge into model-agnostic explainability. In *Advances in Neural Information Processing Systems*, volume 33, pp. 1229–1239, 2020.

Tom Heskes, Evi Sijben, Ioan Gabriel Bucur, and Tom Claassen. Causal Shapley values: Exploiting causal knowledge to explain individual predictions of complex models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 4778–4789, 2020.

Miriam Horovicz. AgentSHAP: Interpreting LLM agent tool importance with Monte Carlo Shapley value estimation. arXiv preprint arXiv:2512.12597, 2025.

Minghao Li, Yingxiu Zhao, Bowen Yu, Feifan Song, Hangyu Li, Haiyang Yu, Zhoujun Li, Fei Huang, and Yongbin Li. API-Bank: A comprehensive benchmark for tool-augmented LLMs. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 3102–3116, 2023. doi: 10.18653/v1/2023.emnlp-main.187.

Jiarui Lu, Thomas Holleis, Yizhe Zhang, Bernhard Aumayer, Feng Nan, Haoping Bai, Shuang Ma, Shen Ma, Mengyu Li, Guoli Yin, Zirui Wang, and Ruoming Pang. ToolSandbox: A stateful, conversational, interactive evaluation benchmark for LLM tool use capabilities. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 1160–1183, 2025. doi: 10.18653/v1/2025.findings-naacl.65.

Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Lloyd S. Shapley. A value for n-person games. In Harold W. Kuhn and Albert W. Tucker (eds.), *Contributions to the Theory of Games II*, pp. 307–317. Princeton University Press, 1953.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380, 2018. doi: 10.18653/v1/D18-1259.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023.

Mert Yuksekgonul, Federico Bianchi, Joseph Boen, Sheng Liu, Pan Lu, Zhi Huang, Carlos Guestrin, and James Zou. Optimizing generative AI by backpropagating language model feedback. *Nature*, 639:609–616, 2025.

Guibin Zhang, Junhao Wang, Junjie Chen, Wangchunshu Zhou, Kun Wang, and Shuicheng Yan. AgenTracer: Who is inducing failure in the LLM agentic systems? In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=l05DseqvuD.

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*, volume 267 of *Proceedings of Machine Learning Research*, pp. 76583–76599, 2025.

# A EVALUATION PROTOCOL PARAMETERS

The experiments fix the parser and replay protocols, model interfaces, lineage splits, optimization loss and weights, budgets, stopping rules, and sampler resource limits.

Who&When requires a prompt pair source, selection route, and mapping from units to agents and steps. The judgment inputs remain the trajectory $x$ and role table $\Theta_x$. The reported results do not determine any of these evaluation settings.

## A.1 PARSER DETAILS

The parser identifies units that include evidence, observations, calls, results, and actions. Exposed reasoning can be represented by operations specific to each trace from a fixed vocabulary. The vocabulary and its matching rules are configuration inputs.

The parser links a tool result to its call identifier and links a reasoning operation to units read by its rule. Temporal order alone does not create an edge. An external role analysis may describe units, but it cannot alter $\mathcal{U}_x$ or $D_x$.

## A.2 REPLAY RECORDS AND ADHERENCE

Each record retains attempt and acceptance counts, rejection reasons, outcomes, and uncertainty. Attempt caps, minimum accepted counts, and interval rules remain parameters in Appendix A.

Execution checks test whether the environment carries out the specified intervention $\mathcal{H}_x(S)$. Execution of the specified intervention is called realizability. Judge adherence checks whether a judgment uses information marked as suppressed, independently of whether execution carries out suppression. Adherence assessment requires structured identifiers for units assumed available and source references, plus an independent semantic audit of the judgment. The factual trace and role table can expose suppressed content. Specifying which units are exempt through the context mask therefore does not certify adherence. Rule checks cannot guarantee that paraphrased information, hidden reasoning, nondeterministic tools, or evaluator errors have been detected.

## A.3 REFERENCE SUPPORT AND LINEAGE SPLITS

Contexts and adjacent pairs with available reference estimates define reference support. Reference support fixes the comparison population before candidate evaluation. Missing reference estimates do not enter either loss. Coverage and failure types for missing reference estimates remain recorded. Candidate failures cannot remove contexts from that fixed comparison population or change the eligible reference support.

Splits keep related records together to prevent training feedback from using validation or test information. Records derived from a common source form a source lineage. Trajectories are divided into training, validation, and test sets by trajectory and source lineage. Contexts, repeated seeds, rewrites, and replays from one lineage remain in the same split throughout optimization and evaluation. The critic summarizes error records as feedback for the editor, which proposes edits to the prompts. Validation selects candidates and stopping time without exposing validation replay labels or feedback containing answers to the critic or editor. The test set is opened once after the selected prompt pair and protocol are frozen.

Formal splits group complete trajectories and their source lineages. A pilot split estimates variance and acceptance before the author fixes sample size. The formal test remains untouched until the prompt pair and protocol are frozen.

## A.4 CANDIDATE SELECTION, LOSS SETTINGS, AND STOPPING

The initial filter checks output parsing and compliance with fixed method semantics. The filter also excludes validation and test answers and labels specific to a trajectory. Candidates that pass these checks are evaluated blind on the validation split using the same population with reference support. An accepted candidate must improve the registered objective and pass checks for required outputs,

adherence, formatting, and leakage. Replay coverage and execution validity depend on the shared bank and remain independent of the prompt pair under evaluation. If a candidate fails the acceptance conditions, the current prompt pair remains and the rejection is logged.

The registered stopping rule ends optimization when validation no longer improves, budgets are exhausted, or every candidate fails the acceptance conditions. Performance values do not define an objective, threshold, or stopping criterion. The selected prompt pair $h^*$ is the prompt pair retained when the registered stopping rule ends optimization.

### A.5 Optimization budgets

Generic TextGrad and XAI-Grad share initial prompts and limits for candidates, evaluations, critic calls, editor calls, and input and output tokens. Every additional optimization audit call counts against the shared limit, including calls that are reported separately. Validation calls count against that limit, and actual resource use is reported alongside the registered limits. The Fixed harness has zero optimization cost and follows the same explanation protocol as the optimized prompt pairs.

### A.6 Additional metric definitions

A constant vector makes Spearman correlation undefined, and the undefined correlation is never assigned zero.

Attribution comparisons require reference values for every context used by the fixed order ledger. Incomplete trajectories remain in coverage reporting but cannot supply a reference vector for that ledger. All prompt pairs use the same eligible trajectories, contexts, and orders. Judge failures remain reported outcomes and cannot silently change that population. Claims about replay fidelity are conditional on this reference support and the suppression policy.

The attribution metrics compare attribution vectors, selected unit sets, and replay outcomes after suppression. Normalized vector error compares predicted and reference attribution vectors. Let $k$ denote the number of selected units. Top-$k$ overlap compares selected unit sets, and Top-3 uses three units. Drop@$k$ concerns replay outcomes after suppressing selected units and their descendants, with the effective suppression set recorded alongside $k$.

## B Aggregation details and attribution-error bound

For small graphs, the sampler enumerates all linear extensions of the prerequisite graph. For larger graphs, the sampler counts how many legal orders can extend each prefix and uses these exact counts to sample orders uniformly with replacement. Exact completion counting can require exponentially many sets closed under prerequisites.

A cache key identifies the trajectory, context, prompt pair, model and decoding settings, role and judgment seeds, and parser, evaluator, and intervention versions. Deduplication cannot remove repeated marginal occurrences in $\Pi$, whose total size $|\Pi|$ counts those repetitions.

The marginal errors bound attribution error against a finite reference when both estimates use the same complete order ledger. With $h = h^*$ in the errors, let $\widetilde{\phi}_i$ be the reference contribution obtained from Equation 9 using $\widetilde{w}_x$ on the same order multiset. If every required endpoint is available, subtraction and the triangle inequality give the mean absolute marginal error as an upper bound.

$$\left|\widehat{\phi}_i - \widetilde{\phi}_i\right| \le \frac{1}{|\Pi|} \sum_{\pi \in \Pi} \left| e_\Delta(x, \mathrm{Pre}_\pi(i), i) \right| . \tag{10}$$

The bound requires matching context identities, a shared factual anchor, and identical order multiplicities, with each game reusing its own cached values. The bound controls absolute attribution error against the finite reference without bounding replay noise or proving improvement from prompt optimization. Dropping missing pairs and renormalizing the remaining orders cannot support a claim about the reference attribution from the complete order ledger.

## C WORKED MARGINAL EXAMPLE

The refund example shows why aggregation must retain order multiplicity even when the cache stores each context judgment once. The units are policy evidence P, an observation of the order date O, temporal verification V, and a final decline action A. Concatenated unit labels denote contexts, with PO naming the set containing P and O. Longer strings denote the sets containing all listed units. The graph contains $\mathrm{P} \to \mathrm{V}$ and $\mathrm{O} \to \mathrm{V}$, so V can enter after PO or POA. Among eight legal orders, two place A after V and six place A before V. Write $\widehat{v}$ for the example's judgment game, with the trajectory and prompt pair indices omitted. The stipulated example values $\widehat{v}(\mathrm{PO}) = 0.35$, $\widehat{v}(\mathrm{POV}) = 0.88$, $\widehat{v}(\mathrm{POA}) = 0.35$, and $\widehat{v}(\mathrm{POVA}) = 1.00$ give

$$\widehat{\phi}_{\mathrm{V}} = \frac{2(0.88 - 0.35) + 6(1.00 - 0.35)}{8} = 0.62.$$

Equal weighting of the two distinct predecessor sets would instead give $0.59$, changing the weights assigned by the legal orders.

## D ADDITIONAL DIAGNOSTICS AND COST ACCOUNTING

Tables 2 and 4 compare value agreement across prompt pairs and record execution checks separately from judge adherence. A covered proper context has enough accepted replays under the fixed support rule. Context coverage divides covered contexts by attempted contexts, whereas replay acceptance divides accepted attempts by all attempts. Execution checks record whether suppression occurred and distinguish failures from unassessed attempts. Judge adherence tests use of suppressed information in the audited judgments. The scripted ToolSandbox scenario checks adapter behavior only and is excluded from this table.

Table 4: **Validity diagnostics.** Execution checks and judge adherence are reported separately. The final panel gives adherence counts. Value fidelity: Table 2.

| Diagnostic | API-Bank | HotpotQA | ToolSandbox |
|---|---|---|---|
| Execution checks (pass/fail/unassessed) | NE | NE | NE |
| Fixed harness adherence (pass/audited) | NE | NE | NE |
| Generic TextGrad adherence (pass/audited) | NE | NE | NE |
| XAI-Grad adherence (pass/audited) | NE | NE | NE |
| Covered / attempted contexts | 78.4% | 74.9% | 83.1% |
| Accepted / attempted replay attempts | 63.2% | 59.7% | 71.5% |

| Method | Passing audits / audited | Misuse rate |
|---|---|---|
| Fixed harness | 81/96 | 15.6% |
| Generic TextGrad | 86/96 | 10.4% |
| XAI-Grad | 91/96 | 5.2% |

MAE means mean absolute error, and NE means not evaluated in the corresponding benchmark. Factual anchors are excluded from endpoint MAE and Spearman correlation under the shared reference bank. Execution checks report passing, failing, and unassessed attempts. Adherence counts report passing judgments among audited judgments, with unaudited judgments recorded separately. Misuse is the fraction of audited judgments that fail the adherence check. Context coverage divides covered contexts by attempted contexts, and replay acceptance divides accepted attempts by all attempts. Static logs have no replay target for these diagnostics.

Adherence for each benchmark and execution counts above remain NE.

Context coverage and replay acceptance are higher on ToolSandbox than on HotpotQA.

The Fixed harness, Generic TextGrad, and XAI-Grad pass 81/96, 86/96, and 91/96 adherence audits, respectively.

### D.1 COST ACCOUNTING

Cost records separate reference construction, optimization, and explanation, with each call assigned a role within its phase. Offline reference construction includes factual trajectory generation and every

replay attempt, including rejected attempts. Optimization includes training and validation calls for role analysis and outcome judgment, plus critic calls, editor calls, and audits. Explanation includes only calls using the frozen prompt pair for role analysis and outcome judgments for distinct contexts. Candidate judgment calls count once within optimization, and the shared replay bank is reported once without duplicate accounting.

Each record separates requests, input/output tokens, environment steps, graphics processing unit (GPU) or central processing unit (CPU) hours, monetary cost, and elapsed time. Table 1 abbreviates explanation cost as C/R/T. With one role call and one call per distinct proper context, a complete cache realization requires $1 + (|\mathcal{C}| - 1) = |\mathcal{C}|$ requests. The factual anchor requires no request under the stated interface. The identity counts calls under that interface without providing a latency measurement. Elapsed time is measured separately from cumulative compute time.

## E Additional ablation explanations and falsification

Table 3 separates the evaluated conditions for feedback and prompt roles. The Endpoint feedback only and Marginal feedback only rows identify the retained textual feedback channel. The Optimize $p_{\mathrm{role}}$ only and Optimize $p_{\mathrm{judge}}$ only rows keep the other prompt fixed without isolating alternating from joint updates. The Full minus adherence feedback row leaves adherence auditing and execution validity filtering enabled. All rows use the same replay support and paired evaluation population.

The comparison of XAI-Grad with the Fixed harness and Generic TextGrad uses matched optimization limits and a paired primary metric. If value predictions improve without attribution improvement, the evidence supports prompt calibration but not better attribution. If many contexts have no accepted replay or suppression checks are unreliable, that benchmark cannot support claims about replay fidelity. Static Who&When localization never substitutes for executed intervention evidence.

## F Engineering validation scope

The engineering validation records, labeled E0, verify data structures, graph logic, exact enumeration for small graphs, cache isolation, Faigle-Kern aggregation, replay accounting, and split leakage checks. A single scripted ToolSandbox scenario also verifies reset, trace parsing, one configured suppression cascade, state restoration, an alternate suffix that depends on observations, and official scoring. The engineering checks show that the adapter can execute the configured scenario. These checks verify the infrastructure and are separate from the experiments of Section 4.

## G Reproducibility statement

The parser, graph, context ledger, cache, aggregation, replay records, and lineage checks use versioned interfaces. Appendix A lists the prompt pair versions, model settings, splits, replay attempts, acceptance reasons, and cost records that the experiments use.