From System Performance to Decision Attribution in Agent Evaluation

Anonymous authors
Paper under double-blind review

Abstract

In agent evaluation, how much of an apparent statistical advantage comes from the decision policy, and how much from the human-designed system around it? An expert shortlist can remove difficult choices before the statistical policy makes a decision, while the LLM may have to propose actions on its own. End-to-end scores combine these contributions. We introduce *Matched Policy Intervention* (MPI), a framework for component-level decision attribution in agent evaluation. MPI holds the *decision affordance* (the external information, action space, budget, and execution rules) fixed and preassigns which policy's recommendation is executed. Paired recommendations reveal whether similar outcomes arise from similar decisions. Across CausalGame, CARE, and a discrete-action adaptation of CausaLab, we find that some apparent end-to-end advantages disappear under matched affordances. On CARE's ChemLex task, a +5.72-point system gain accompanies a matched gate-minus-LLM effect of −0.09 points. In CausalGame, changing the affordance alone raises benchmark success from 25% to 62.5% under a fixed statistical policy. MPI distinguishes the contribution of a decision policy within an agent from the support supplied by its decision affordance.

1 Introduction

When an agent improves after adopting a statistical decision policy, what should the evaluation credit? We study this attribution question by comparing LLM and statistical policies at decision stages within agent workflows, such as recommending the next experiment, choosing what evidence to acquire, or deciding which result to submit. Systems such as *CauTion* already face this choice when deciding whether a causal discovery ensemble should trust an LLM (Peng et al., 2026). A common evaluation compares separately constructed end-to-end systems; for example, Harris & Slivkins (2025) compare LLM and algorithmic policies on exploration and exploitation tasks. Such evaluations measure which system performs better. Attributing that difference to the decision policy requires accounting for the support each system supplies.

Consider an agent selecting a catalyst for the next experiment. An LLM may receive the experimental history and propose any catalyst type, while a statistical method chooses from palladium, nickel, and copper. If a developer selected these candidates using chemical expertise, part of the decision has already been made before the statistical method runs. An apparent statistical advantage can therefore include the contribution of a human expert who narrowed the problem. We call this *action-space privilege*: one policy receives a curated set of choices that another must construct or search for itself. Figure 1(a) illustrates how a comparison can credit human candidate construction to the final selection policy.

This distinction matters because constructing actions and choosing among given actions are different capabilities. Human-designed menus can improve or impair performance by restricting choices; an open action space leaves more of the problem to the LLM. A comparison between these interfaces does not establish which policy selects better under the same conditions. Additional runs of the same two systems improve precision without separating those contributions. The question is therefore how to distinguish the decision policy from its *decision affordance*: the external information, available actions, budget, and execution rules surrounding the decision.

We introduce *Matched Policy Intervention* (MPI) for component-level decision attribution in agent evaluation. Both policies receive the same external record and action menu. Before each episode,

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.

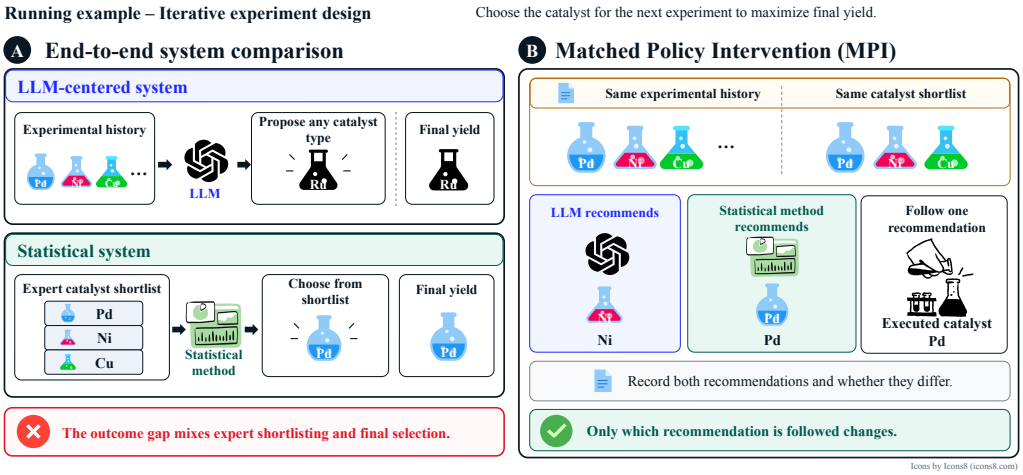

Figure 1: Separating policy decisions from human-designed support. (a) End-to-end evaluation changes both candidate construction and the policy making the final choice. The outcome gap combines their contributions. (b) MPI gives both policies the same history and catalyst shortlist, then blindly preassigns which recommendation is executed. Both recommendations are recorded.

the evaluation protocol specifies which policy's recommendation will be executed at each declared stage, without revealing that assignment to either policy. We call the policy whose recommendation is executed the *active policy*. Figure 1(b) shows the resulting comparison. Holding the affordance fixed lets us ask whether following the statistical policy changes the outcome relative to following the LLM policy.

Sequential agents require this assignment to be defined over stages. For example, acquisition decisions change the evidence available for a later submission. MPI counts those downstream consequences as part of the acquisition policy's effect. It also logs both recommendations at every reached decision, allowing us to distinguish similar outcomes under different choices from simple agreement between policies.

Our contributions are:

1. **We expose an attribution problem in agent evaluation.** End-to-end gaps can combine a policy's decisions with human-designed action spaces and other affordance choices, misrepresenting or even reversing the policy comparison.
2. **We introduce a matched-affordance evaluation protocol.** MPI compares preassigned execution policies under matched conditions to estimate stage-specific effects and uses paired recommendations to measure policy disagreement.
3. **We show why system gains need component-level evidence.** In CausalGame (Chen et al., 2026), CARE (Liu et al., 2026), and a discrete-action adaptation of CausaLab (Yang et al., 2026), some apparent end-to-end advantages disappear under matched affordances. A separate CausalGame experiment improves benchmark success by changing the affordance while keeping the statistical policy fixed.

2 Problem Formulation

2.1 Policies, affordances, and stage-specific effects

Attributing a system gain to a policy requires specifying which decision changes, which affordance is held fixed, and how later decisions enter the comparison. We refer to the LLM and statistical action-selection policies as L and S, respectively. Each maps the information available at a decision point to a recommended action. The following formulation separates the shared decision problem from each policy's internal computation.

For a sequential agent, we index episodes by $i \in \{1, \ldots, n\}$ and decision opportunities within episode i by $t \in \{1, \ldots, T_i\}$. Each opportunity belongs to one of J declared stages: $j(t) \in \{1, \ldots, J\}$. For example, a two-stage agent may acquire information at stage 1 and submit a final answer at stage 2. Let H_{it} denote the complete system history immediately before decision t, including previous observations and actions and the remaining budget.

Write $j = j(t)$ for the current stage. Fixed system rules map the current history to a finite action menu and a common decision record,

$$C_{it} = M_j(H_{it}) \subseteq \mathcal{D}_j, \qquad Z_{it} = R_j\big(V_j(H_{it}), C_{it}\big),$$

where $\mathcal{D}_j$ is the stage-j action schema, V_j determines which external evidence is visible, and R_j renders that evidence together with the realized menu. The record can include observations, prior actions, and the remaining budget.

The statistical policy may compute a private state from this record,

$$Q_{it}^S = g_j^S(Z_{it}, C_{it}; \eta_j^S),$$

where g_j^S may implement posterior inference, uncertainty estimation, or statistical decision scoring, and η_j^S denotes fixed model parameters and priors. Thus Z_{it} is the shared external record, whereas Q_{it}^S is a policy-specific state used only by the statistical policy.

The LLM's fixed policy instructions and episode-independent in-context examples likewise specify its policy, analogous to η_j^S. Prompt content conveying episode-specific evidence or the action menu, including its rendering, remains part of the shared affordance.

We collect the stage-j external conditions into the *decision affordance*

$$F_j = (M_j, V_j, R_j, \mathcal{B}_j, G_j, E_j),$$

where $\mathcal{B}_j$ specifies budget rules, G_j contains guardrails and failure semantics, and E_j executes the selected action, updates the environment, and contributes to scoring. Thus F_j determines which actions are available, what information is exposed, and what happens after selection. Candidate construction can itself embody consequential design choices; here we hold it fixed to isolate subsequent selection. The affordance excludes the two decision policies and which policy controls enactment. We write $F = (F_1, \ldots, F_J)$ for the full affordance allocation. "Fixed" means that these construction and execution rules are identical across policy-assignment arms, not that their realized menus or records are constant across histories.

At each decision, the two stage-specific policies return recommendations from the same realized menu:

$$\widetilde{D}_{it}^L = \pi_j^L(Z_{it}) \in C_{it}, \qquad \widetilde{D}_{it}^S = h_j^S(Z_{it}, Q_{it}^S) \in C_{it}.$$

Our goal is to estimate how the final outcome changes when a specified decision stage follows the statistical policy rather than the LLM policy, under a fixed decision affordance.

Let $a = (a_1, \ldots, a_J) \in \{L, S\}^J$ denote a complete policy assignment across the J stages, and let a_{-j} denote that vector with its stage-j entry removed. Let f be a fixed value of the full affordance allocation F, and let $Y_i(a, f)$ be the potential endpoint of episode i under policy assignment a and affordance f. We use $Y_i(S, a_{-j}, f)$ as shorthand for the endpoint when stage j is assigned to the statistical policy and the remaining stages follow a_{-j}; the notation $Y_i(L, a_{-j}, f)$ is defined analogously.

We orient policy effects so that positive values favor statistical control. The stage-j policy effect at affordance f is

$$\tau_j^P(f) = \frac{1}{2^{J-1}} \sum_{a_{-j} \in \{L,S\}^{J-1}} \mathbb{E}\big[Y_i(S, a_{-j}, f) - Y_i(L, a_{-j}, f)\big]. \tag{1}$$

The superscript P denotes the policy effect, and the expectation is over the target distribution of episodes and environment randomness. The factor 2^{J-1} averages the contrast over every allocation of the other $J - 1$ stages. Thus $\tau_j^P(f)$ isolates stage j without conditioning the conclusion on one particular policy assignment elsewhere in the agent.

Earlier policy assignments may produce different later histories and menus. Those downstream paths are part of assigning the active policy in a sequential agent; conditioning them to be equal would remove part of the treatment effect.

2.2 Separating policy and affordance contributions

An end-to-end gap combines the effects of changing the policy and changing its affordance. We summarize this approximately as

$$\Delta_{\text{total}} = \Delta_{\text{policy}} + \Delta_{\text{affordance}}. \tag{2}$$

The policy term compares policies under a fixed affordance; the affordance term changes that affordance with a policy held fixed. To make the reference path explicit, consider an LLM-controlled implementation and a statistical-control implementation. Let $\mathbf{L}$ and $\mathbf{S}$ denote the all-LLM and all-statistical allocations. If the implementations use affordances F_L and F_S, respectively, their comparison is

$$\begin{aligned}\Delta_{\text{system}} &= \mathbb{E}[Y_i(\mathbf{S}, F_S) - Y_i(\mathbf{L}, F_L)] \\ &= \underbrace{\mathbb{E}[Y_i(\mathbf{S}, F_L) - Y_i(\mathbf{L}, F_L)]}_{\text{decision maker at fixed affordance}} + \underbrace{\mathbb{E}[Y_i(\mathbf{S}, F_S) - Y_i(\mathbf{S}, F_L)]}_{\text{affordance at fixed decision maker}}.\end{aligned} \tag{3}$$

This decomposition assumes that $(\mathbf{S}, F_L)$ is executable. When a statistical rule cannot operate on the LLM's native action interface, this cell is undefined; one can instead decompose through $(\mathbf{L}, F_S)$ if that cell is executable. The path is baseline-specific in the presence of interaction. MPI targets a common affordance on which both methods can run.

Action-space privilege is one source of the affordance contribution: an expert shortlist can improve the statistical system before its policy makes a choice. For example, suppose the LLM outperforms the statistical policy by five points under each of two fixed affordances, while an enhanced affordance f_{enh} improves either policy by 20 points relative to a baseline affordance f_{base}. A comparison of $(\mathbf{L}, f_{\text{base}})$ with $(\mathbf{S}, f_{\text{enh}})$ reports a 15-point statistical advantage, reversing the policy ordering. Additional replications of those two cells cannot supply the missing counterfactual cells. MPI instead holds the decision affordance fixed and changes which recommendation is enacted.

3 Matched Policy Intervention

MPI uses a standard complete factorial evaluation of preassigned execution policies to implement the stage-specific contrast in Eq. (1). Its methodological focus is specifying and checking the boundary between the decision policy and its affordance. This requires defining the shared decision problem, the assignments at other stages, and the downstream outcomes included in the effect. Section 3.1 states the required conditions, Section 3.2 establishes identification and estimation, and Section 3.3 uses paired recommendations to interpret the resulting comparison.

The stage-level policy-assignment indicator $A_{ij} \in \{L, S\}$ is assigned before episode i begins, is hidden from the policies, and applies to all decision opportunities belonging to stage j in that episode. It determines the enacted action D_{it}:

$$D_{it} = \begin{cases} \widetilde{D}_{it}^{L}, & A_{ij} = L, \\ \widetilde{D}_{it}^{S}, & A_{ij} = S. \end{cases} \tag{4}$$

For a two-stage agent, MPI implements the average in Eq. (1) by running all four preassigned configurations (L, L), (L, S), (S, L), (S, S) within each task–seed block.

3.1 Conditions for valid policy attribution

To interpret an outcome contrast as the effect of following one policy rather than the other, the two policies must occupy the same decision point. We require three conditions:

1. **Identical external inputs and action spaces.** Both policies receive the same external record, choose from the same realized menu, and return actions in the same format.
2. **Frozen execution rules.** Menu and record construction, budget, executor, endpoint, guardrails, and failure policy do not change across experimental arms.

3. **Preassigned execution policies.** The policy controlling each stage is specified before the episode and remains fixed throughout that stage. Both recommendations are generated before either is followed, and neither policy is told which recommendation will control the action.

The final condition prevents either recommendation from responding to the assignment and produces the paired records used for behavioral interpretation. The common external record does not require common internal computation: policy-specific states such as Q_{it}^S remain private to their policies and are not part of the affordance F_j.

For a deterministic statistical policy, reconstruction from the common record means that there exists a function q_j such that $\widetilde{D}_{it}^S = q_j(Z_{it})$ on the tested decision support. This expresses that its recommendation can be reproduced from the record; it does not require the LLM to implement q_j or receive the private state Q_{it}^S. The empirical reconstruction diagnostic and its approximation limits are reported in Appendix C.

3.2 Identifying and estimating the stage-specific effect

Under the conditions below, comparing preassigned configurations within matched task–seed blocks identifies the stage-specific effect in Eq. (1), including downstream consequences of the enacted action.

If the evaluation covers several prespecified affordance levels, it may target the prespecified average $\tau_j^{P,g} = \sum_{f\in\mathcal{F}} g_f \tau_j^P(f)$, where $g_f \geq 0$ and $\sum_f g_f = 1$. Testing only one affordance is the special case that places all weight on that level.

Identification proposition. Let $b \in \{1, \ldots, B\}$ index the prespecified task–seed blocks, let $w_b \geq 0$ be their analysis weights with $\sum_b w_b = 1$, and let $\overline{Y}_{b,x,a_{-j},f}$ denote the observed mean endpoint among episodes in block b assigned policy $x \in \{L, S\}$ at stage j, allocation a_{-j} at the remaining stages, and affordance f. Define the target distribution in Eq. (1) as the mixture that draws block b with probability w_b and then draws an episode and environment randomness conditional on that block. Assume consistency, no cross-episode interference, fixed assignment-blinded decision policies, complete coverage of every planned policy-assignment cell within each task–seed block, common action support, and frozen surroundings. Within a block, all configurations must use the same target distribution of episode instances and external execution conditions, without configuration-dependent selection or retention of runs. Then the blocked factorial contrast

$$\widehat{\tau}_j^P(f) = \sum_{b=1}^{B} w_b \frac{1}{2^{J-1}} \sum_{a_{-j}\in\{L,S\}^{J-1}} \left(\overline{Y}_{b,S,a_{-j},f} - \overline{Y}_{b,L,a_{-j},f}\right) \tag{5}$$

identifies the block-weighted assignment effect of statistical rather than LLM control. Its interpretation under a common decision problem relies on the conditions in Section 3.1; empirical interface diagnostics are reported in Appendix C.

Equation (5) is evaluated separately within each affordance level. A standardized target uses the same prespecified weighted average, $\widehat{\tau}_j^{P,g} = \sum_f g_f \widehat{\tau}_j^P(f)$.

If the evaluation also crosses policy configurations with two prespecified affordance levels f_0 and f_1, then, for a fixed complete policy assignment a, the same factorial design identifies the affordance effect and the policy–affordance interaction:

$$\tau^F(a) = \mathbb{E}[Y_i(a, f_1) - Y_i(a, f_0)] \quad \text{and} \quad \tau_j^{P\times F} = \tau_j^P(f_1) - \tau_j^P(f_0).$$

Here $\tau^F(a)$ measures the effect of changing the affordance while holding all policy assignments fixed, and $\tau_j^{P\times F}$ measures how that affordance change alters the stage-j policy effect. An affordance main effect can likewise be standardized over policy assignments as $\tau^{F,h} = \sum_a h_a \tau^F(a)$ using prespecified weights. The CausaLab study uses equal weights over its coverage levels and two-stage policy assignments. These additional cells distinguish a policy effect from a change caused by the menu, visible information, or another aspect of the decision affordance.

We also check the estimator in a synthetic setting with known policy and affordance effects (Appendix B). This numerical check uses simulated outcomes rather than runs of LLM or statistical decision policies.

3.3 Policy disagreement and practical interpretation

Paired recommendations complement the identified outcome effect by showing whether the policies made meaningfully different choices. We record both recommendations at each reached decision, including the one that is not executed. We define the policy disagreement rate ρ_j as the fraction of reached stage-j decisions on which $\widetilde{D}^L_{it} \neq \widetilde{D}^S_{it}$. We also report the share of episodes with any disagreement and, when different action indices can encode the same value, both index- and value-level disagreement. Recording the unexecuted choice does not reveal its unobserved downstream outcome and cannot replace running the alternative policy configurations. When the estimated outcome effect is small, high disagreement shows that different recommendations can attain similar outcomes; low disagreement instead indicates convergence on the tested choices.

For deployment-oriented analyses, we compare simultaneous confidence bounds with a prespecified loss tolerance δ: an upper bound below δ supports DELEGATE to the LLM, a lower bound above δ supports RETAIN statistical control, and all other cases are INCONCLUSIVE. The rule uses standard one-sided non-inferiority tests to avoid treating nonsignificance as equivalence. Its verdicts concern the specified stage average and loss tolerance, not equality or every individual allocation of the other stages.

4 Empirical Evaluation

Our experiments examine whether system-level gains survive a matched policy comparison and whether human-designed support can change outcomes without changing the policy. CausalGame compares complete systems with stage-specific policy replacements and separately tests an affordance change. CARE asks whether a statistical gate explains a released system's improvement. A discrete-action CausaLab adaptation tests the protocol with a different task and candidate-construction procedure.

Sequential studies estimate non-LLM minus LLM policy effects within task–seed blocks using Eq. (5); five percentage points is the common practical-loss reference. CARE uses paired seed contrasts. Full inference details and results appear in Appendices A and D.

4.1 CausalGame: system gaps shrink under matched affordances

CausalGame is an interactive causal-reasoning benchmark in which an agent designs and deploys drones governed by a hidden structural causal model. In each session, the agent allocates a 200-drone exploration budget over at most ten acquisitions and then commits one design for a 1,000-drone evaluation; the endpoint is fleet survival. The main stage-wise grid focuses on three mechanism-rich scenes; the released-mode comparison below uses three overlapping benchmark scenes.

This study asks whether a difference between complete CausalGame systems can be attributed to who makes the stage-wise decisions. MPI assigns acquisition and submission control separately while holding the affordance fixed. A frozen candidate-construction procedure supplies the common menu, and the policy assignments are fixed before each session. The main comparison spans three GPT models; the complete grid appears in Appendix D.

The released comparison changes the benchmark's agentic and prompting modes together, whereas MPI changes only which method controls acquisition and submission. Table 1 places these two quantities side by side on three shared scenes.

Component-level attribution yields much smaller point estimates: the complete-system differences range from +0.3 to −33.3 points, whereas the matched joint-policy effects range from −0.8 to +1.8. Thus the released mode gap does not establish an advantage of statistical over LLM decision making. Because the two experiments use different arms, we do not subtract these estimates to quantify an affordance contribution.

scene	complete-system difference	MPI effect
`antenna_trap`	+0.3 ± 9.3	−0.8 ± 5.1
`weather_noise`	−6.6 ± 9.7	+1.8 ± 5.3
Simpson's paradox	−33.3 ± 24.4	−0.3 ± 4.8

Table 1: Complete-system differences and MPI effects in CausalGame, in survival percentage points (estimate ± 95% CI half-width). The complete-system column uses GPT-5.6-sol's released agentic minus prompting modes (six seeds per scene); MPI uses its historical 32-block joint $(S, S) - (L, L)$ contrast, not a single-stage effect. The complete-system intervals are retained as originally reported; MPI intervals use paired t critical values with 31 degrees of freedom. These are independent experiments with different comparison arms, not an additive decomposition.

The stage-specific grid gives the same overall picture. For the two larger GPT models, the acquisition and submission effects remain close to zero and exclude an LLM loss greater than five points. GPT-5.4-mini submission does so as well; its acquisition effect instead favors statistical control by +6.00 points, with simultaneous bounds $[+2.82, +9.18]$. The full model–stage grid appears in Appendix D.

Across the full CausalGame grid, including the supplementary no-win scenes, the methods recommend different actions on 53.6–74.8% of acquisition decisions. The small outcome effects for the larger models therefore do not arise from identical choices. Full interface, submission, and model-family diagnostics appear in Appendices C and E.

4.2 CausalGame: affordance design improves benchmark success

The CausalGame initialization experiment asks whether the benchmark outcome can change even when the decision maker is fixed. Holding statistical control and seeds constant, balanced screening raises the number of sessions crossing the benchmark-defined 75% victory threshold from 6/24 (25%) to 15/24 (62.5%; exact McNemar $p = .012$). Thus, changing the decision affordance alone can improve the benchmark success rate without changing who decides. This improvement concerns the benchmark's success threshold; the changes in mean survival are small and imprecise (Appendix F.1).

4.3 CARE: a system gain need not be a gate advantage

CARE performs sequential chemical optimization over a finite candidate pool. An LLM proposes a challenger policy, after which a deterministic gate decides whether to authorize the challenger before the next unobserved reaction outcome is revealed. We replace only this gate with a GPT-5.4-mini gate policy on the Minerva and ChemLex datasets, using 30 matched seeds per policy and dataset. Both gate policies receive the same nine evidence components, choose from the same authorize/reject actions, and use the same executor. Protocol and confirmatory-sample details appear in Appendix G.

This study asks whether CARE's complete-system improvement can be attributed to the identity of its gate policy. On ChemLex, full CARE reaches a mean final yield of 89.62 versus 83.91 for the incumbent, a +5.72-point system gain. Yet replacing the deterministic gate with the LLM gate policy yields 89.71: the matched gate−LLM policy effect is −0.09 points (95% CI $[-2.38, +2.20]$). The observed CARE advantage on ChemLex therefore does not identify the deterministic gate as its source. Across the two datasets in Table 2, MPI distinguishes system gains that persist at the component level from those that do not.

dataset	CARE − incumbent	gate − LLM (95% CI)
ChemLex	+5.72	−0.09 $[-2.38, +2.20]$
Minerva	+7.83	+11.01 $[+6.38, +15.65]$

Table 2: CARE complete-system improvements and effects of changing only the gate policy over 30 seeds per policy. The second comparison holds the evidence, available actions, and executor fixed. Minerva's pooled estimate includes ten exploratory seeds; the subsequent 20-seed confirmatory effect is +14.35 $[+8.34, +20.36]$. Intervals use paired t inference on unrounded seed-level differences.

4.4 CausaLab: evaluation across shared menus

We use a discrete-action adaptation of CausaLab's task specification to test MPI under a different task and candidate construction procedure (Appendix H). Each episode samples a hidden ten-node linear or quadratic SCM, provides two observational records and 36 hard-intervention opportunities, and asks the agent to predict a held-out intervention frequency.

For context, an end-to-end comparison of all-LLM control with neutral menus against all-statistical control with screened menus favors the LLM by 9.38 points for GPT-5.4-mini and 12.50 for GPT-5.6-sol. We use the matched design below to evaluate stage-specific policies within shared menu conditions.

The experiment crosses acquisition control, submission control, and menu coverage for GPT-5.4-mini and GPT-5.6-sol, with 32 blocks per model and 512 episodes in total. Both policies receive the same realized menus. At acquisition, the non-LLM policy is a deterministic coverage rule; at submission, it is a BIC-weighted aggregation rule over a shared candidate menu.

model	stage	non-LLM − LLM	simultaneous bounds
GPT-5.4-mini	acquisition	+0.00	[−4.66, +4.66]
GPT-5.4-mini	submission	−3.12	[−11.02, +4.77]
GPT-5.6-sol	acquisition	−1.56	[−4.75, +1.63]
GPT-5.6-sol	submission	−3.12	[−8.59, +2.34]

Table 3: Effects of changing acquisition and submission control in CausaLab, averaged equally over the neutral and screened menus. Each comparison holds the decision affordance fixed within its menu condition.

Relative to the end-to-end gaps, the four stage-specific point estimates are smaller in magnitude, ranging from −3.12 to 0.00 points. All exclude an LLM loss greater than five points, although the intervals do not uniformly rule out sizable LLM advantages. Acquisition recommendations differ on 61.3% and 74.2% of decisions, confirming that this result compares distinct choices. Full affordance and submission diagnostics appear in Appendices E and F.2.

5 Related Work

Agent scaffolds and evaluation. Scaffold choices can move agent performance substantially even when the underlying model is fixed (Starace, 2026). Recent work standardizes, maps, and discloses evaluation harnesses (Zhu et al., 2026a;b; Rombaut, 2026). Zhang et al. (2026) advocate both locked-harness comparisons and factorial evaluation of model and harness choices. These practices make the support surrounding an agent visible and improve system-level comparisons. MPI focuses on a component-level question: how does the final outcome change when a particular stage follows a different policy under the same affordance? Blinded policy assignment, common action support, and stage-wise comparisons make that question experimentally testable.

Policy choice and delegation. Prior work compares LLM and statistical decisions in isolated tasks or workflow roles (Harris & Slivkins, 2025; Buitrago López et al., 2026; Xie et al., 2026; Wang & Cui, 2026), and recent systems learn or prescribe which decision maker to follow (Dixon, 2026; Peng et al., 2026). Function allocation and adjustable autonomy have also studied when control should move between humans and automation (Sheridan & Verplank, 1978; Parasuraman et al., 2000; Horvitz, 1999). The clinical-versus-actuarial literature compares rules with expert judgment on common inputs (Meehl, 1954; Dawes et al., 1989). MPI examines how human-designed support enters comparisons of LLM and statistical policies and provides matched evidence for replacing a decision maker. Non-inferiority tests bound the practical loss relevant to that decision (Lakens, 2017).

Attribution and experimental validity. Attribution work studies estimable alternatives to infeasible causal quantities (Han et al., 2025), counterfactual effect decomposition in sequential multi-agent systems (Triantafyllou et al., 2025), and the effects of corruption, distribution shift, information leakage, and metric choice on attribution conclusions (Zhang & Nanda, 2024; Zheng et al., 2025). MPI

uses these concerns to specify a valid policy replacement: the external decision problem and execution rules must remain fixed while the active policy changes. Its matched comparisons estimate the resulting outcome effect, including consequences at later stages.

6 Discussion and Conclusion

In agent evaluation, an apparent statistical advantage can include decisions made by the human who designed the action space. Some end-to-end advantages disappear under matched affordances, while changing the affordance alone can improve benchmark success under a fixed statistical policy. MPI provides component-level evidence for changing an agent's decision policy or its support. We focus on gains from adding statistical modules to LLM agents: do they come from the statistical rule or its support? If an LLM performs at least as well under matched affordances, the system gain alone does not justify preferring the statistical rule, even before crediting the LLM's action-construction ability.

Matching action spaces also changes which capability is being evaluated. Our comparisons measure selection from shared menus whose contents may vary with the history. They do not measure an LLM's ability to construct useful actions in a larger or open-ended space, which may itself be an advantage. Where deployment requires action generation without a supplied shortlist or zero-shot transfer to new tasks, these comparisons alone cannot determine which complete system to deploy. Nevertheless, constrained action interfaces are also an established design practice for supporting validation and controlled execution, as in schema-constrained tool calling (Anthropic, 2026). Jev's native selection among supplied options (TypeSafe AI, 2026) further illustrates the practical relevance of MPI's shared-menu setting. A broader evaluation should therefore distinguish action construction from selection and account for human expertise supplied through the affordance. The CausaLab end-to-end comparison illustrates that affordance differences can also accompany an LLM advantage; the same attribution question applies in either direction.

The reported effects concern prespecified stages and shared finite action sets in the tested scientific and causal-reasoning agent tasks. They neither establish a universal ranking of LLM and statistical policies nor quantify the prevalence of action-space privilege across existing agent benchmarks. Transfer to other agent domains remains untested. Extending the protocol to open-ended or continuous actions and learning adaptive delegation policies remain future work.

Reproducibility statement

Code is available at `https://anonymous.4open.science/r/causal_llm-9988/README.md`. The repository provides configuration files, experimental environments, and data preparation and analysis scripts to support reproduction of the evaluation protocols and reported results.

AI use statement

Generative AI tools were used primarily to assist with translation, manuscript drafting, language editing, and code writing. The research questions, MPI framework, experimental protocols, and analytical decisions were developed and determined by the authors. All confirmatory protocols and amendments were frozen before their corresponding data collection; all reported numerical results were regenerated from retained raw data by deterministic scripts; citations and factual claims were checked against source materials; and the authors reviewed all AI-assisted text. The authors take responsibility for the final content of this work, including material produced with generative-AI assistance.

References

Anthropic. Strict Tool Use: Claude Platform Documentation, 2026. URL `https://platform.claude.com/docs/en/agents-and-tools/tool-use/strict-tool-use`. Accessed September 25, 2026.

Alejandro Buitrago López, Javier Pastor-Galindo, and José A. Ruipérez-Valiente. Should LLM agents decide in social simulations? comparing finite-state and LLM-based decision policies. *arXiv preprint arXiv:2606.12369*, 2026.

Zhenhao Chen, Yongqiang Chen, Chenxi Liu, Junchi Yu, Xiangchen Song, Zijian Li, Jialin Li, Philip Torr, Bo Han, and Kun Zhang. CausalGame: Benchmarking causal thinking of LLM agents in games. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. Oral presentation.

Robyn M. Dawes, David Faust, and Paul E. Meehl. Clinical versus actuarial judgment. *Science*, 243 (4899):1668–1674, 1989.

Matthew Francis Dixon. Adaptive AI delegation under uncertainty: A Bayesian governance policy for sequential decision authority. *arXiv preprint arXiv:2606.29406*, 2026.

Yujin Han, Lei Xu, Sirui Chen, Difan Zou, and Chaochao Lu. Beyond surface structure: A causal assessment of LLMs' comprehension ability. In *International Conference on Learning Representations*, 2025.

Keegan Harris and Aleksandrs Slivkins. Should you use your large language model to explore or exploit? *arXiv preprint arXiv:2502.00225*, 2025. Accepted to UAI 2026.

Eric Horvitz. Principles of mixed-initiative user interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pp. 159–166, 1999.

Daniël Lakens. Equivalence tests: A practical primer for t tests, correlations, and meta-analyses. *Social Psychological and Personality Science*, 8(4):355–362, 2017.

Guanyu Liu, Weiyi Kong, Chao Tang, Zeyu Wang, Boer Zhang, Baiqing Li, Peiyu Zhang, and Tianyu Shi. CARE: Context-aware ranking evolution with executable scoring programs for budgeted reaction optimization. *arXiv preprint arXiv:2606.14581*, 2026.

Paul E. Meehl. *Clinical versus Statistical Prediction: A Theoretical Analysis and a Review of the Evidence*. University of Minnesota Press, 1954.

Raja Parasuraman, Thomas B. Sheridan, and Christopher D. Wickens. A model for types and levels of human interaction with automation. *IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans*, 30(3):286–297, 2000.

Bo Peng, Kaiwen Wu, Sirui Chen, Zhiheng Wang, Yu Qiao, and Chaochao Lu. CauTion: Knowing when to trust LLMs for ensemble causal discovery. *arXiv preprint arXiv:2606.03602*, 2026.

Benjamin Rombaut. Inside the scaffold: A source-code taxonomy of coding agent architectures. *arXiv preprint arXiv:2604.03515*, 2026.

Thomas B. Sheridan and William L. Verplank. Human and computer control of undersea teleoperators. Technical report, MIT Man-Machine Systems Laboratory, 1978.

Jason Starace. Scaffold effects on GAIA: A controlled comparison. *arXiv preprint arXiv:2606.08529*, 2026.

Stelios Triantafyllou, Aleksa Sukovic, Yasaman Zolfimoselo, and Goran Radanovic. Counterfactual effect decomposition in multi-agent sequential decision making. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, 2025.

TypeSafe AI. Choice: Jev API Documentation, 2026. URL `https://docs.typesafe.ai/primitives/choice`. Accessed September 25, 2026.

Renxiang Wang and Jiaming Cui. Beyond tier labels: Role- and deployment-dependent model substitution in multi-call LLM workflows. *arXiv preprint arXiv:2608.09155*, 2026.

Zikai Xie, Wenmei Li, Man Luo, Jun Jiang, and Linjiang Chen. Language models guide symbolic equation discovery by controlling search. *arXiv preprint arXiv:2607.04156*, 2026.

Junlin Yang, Dylan Zhang, Xiangchen Song, Qirun Dai, Xiao Liu, Yuen Chen, Aniket Vashishtha, Jing Shi, Chenhao Tan, and Hao Peng. CausaLab: A scalable environment for interactive causal discovery toward AI scientists. *arXiv preprint arXiv:2605.26029*, 2026.

Fred Zhang and Neel Nanda. Towards best practices of activation patching in language models: Metrics and methods. In *International Conference on Learning Representations*, 2024.

Yunbei Zhang, Janet Wang, Yingqiang Ge, Weijie Xu, Jihun Hamm, and Chandan K. Reddy. Stop comparing LLM agents without disclosing the harness. *arXiv preprint arXiv:2605.23950*, 2026.

Xu Zheng, Farhad Shirani, Zhuomin Chen, Chaohao Lin, Wei Cheng, Wenbo Guo, and Dongsheng Luo. F-Fidelity: A robust framework for faithfulness evaluation of explainable AI. In *International Conference on Learning Representations*, 2025.

Pengyu Zhu, Lijun Li, Yaxing Lyu, Qianxin Luo, Jingyi Yang, Yi Liu, Tingfeng Hui, Xinyu Yuan, Li Sun, Sen Su, and Jing Shao. UniACE: A unified framework for evaluating LLM agentic capabilities. *arXiv preprint arXiv:2605.27898*, 2026a.

Pengyu Zhu, Li Sun, Philip S. Yu, and Sen Su. "LLM agent performance" is not a single evaluation target. *arXiv preprint arXiv:2602.03238*, 2026b.

A MPI IDENTIFICATION DETAILS

A.1 BLOCKED FACTORIAL IDENTIFICATION

Let $A_i = (A_{i1}, \ldots, A_{iJ})$ be the policy-assignment vector assigned before episode i, let $b \in \{1, \ldots, B\}$ index the prespecified blocks, and let $w_b \geq 0$ satisfy $\sum_b w_b = 1$. The target distribution first draws block b with probability w_b and then draws an episode conditional on that block. Let $Y_i(a, f)$ denote its potential endpoint under policy-assignment vector a and fixed affordance f. Each planned cell is run within each block; the CausalGame rerun and CausaLab runners shuffle cell order. Under the consistency, comparability, and policy conditions in Section 3.2,

$$\mathbb{E}\left[\overline{Y}_{b,a,f} \mid b\right] = \mathbb{E}[Y_i(a, f) \mid b] .$$

Substitution into Eq. (5), followed by linearity of expectation, yields

$$\begin{aligned}\mathbb{E}[\widehat{\tau}_j^P(f)] &= \sum_b w_b \frac{1}{2^{J-1}} \sum_{a_{-j}} \mathbb{E}\big[Y_i(S, a_{-j}, f) - Y_i(L, a_{-j}, f) \mid b\big] \\ &= \tau_j^P(f).\end{aligned}$$

For tested affordance levels with prespecified weights g_f, linearity gives $\mathbb{E}[\widehat{\tau}_j^{P,g}] = \sum_{f \in \mathcal{F}} g_f \mathbb{E}[\widehat{\tau}_j^P(f)] = \tau_j^{P,g}$, so this prespecified average requires no additional identification assumption beyond support for its component cells. The potential outcome includes all downstream histories induced by the earlier enacted decisions. Thus the argument does not require later menus or states to match across counterfactual episodes. It requires only that the surrounding policy for constructing and executing those menus is frozen.

Under these conditions, the assignment effect is identified regardless of the interface check. A common-decision interpretation requires common external evidence and action support, but allows private policy states. Passing the check shows that native precision or hidden fields rarely change the statistical action relative to the shared record. Failure leaves the estimate valid under these conditions, but requires reporting the evidence mismatch. The diagnostic constructs a fixed function $q_j(Z_{it})$ and measures its action mismatch with the statistical recommendation. Low mismatch supports approximate action reconstruction on the tested distribution; it does not prove equality of information or require the LLM to implement q_j.

Uncertainty and scope. For the equal-weight blocks used here, write D_b for the within-block factorial contrast in Eq. (5). We use

$$\widehat{\tau} = B^{-1} \sum_b D_b, \qquad \widehat{\mathrm{SE}}(\widehat{\tau}) = \sqrt{\frac{\sum_b (D_b - \widehat{\tau})^2}{B(B-1)}}.$$

Recomputed mean-contrast intervals use t_{B-1} critical values; a family of K two-sided comparisons at error allocation α uses $t_{B-1,1-\alpha/(2K)}$. The benchmark-specific allocations are stated in Appendix D. These block-level intervals differ from the decision-level reconstruction diagnostic in Appendix C. The historical complete-system intervals in Table 1 are retained as originally reported, as noted in its caption. Stage averages do not certify every conditional contrast: when a particular deployment allocation is the target, the other stages should be fixed to that allocation. Complete factorial coverage costs 2^J runs per block; larger designs require a prespecified sampling and estimation scheme rather than applying the complete-cell estimator to unobserved cells.

The main CausalGame controller constructs each LLM request afresh from the current public record and menu. It does not append earlier assistant replies, reasoning traces, or unexecuted recommendations. The latter are stored only in the experiment log. Executed actions and their outcomes remain in the shared record, as required for sequential decision making.

A.2 AFFORDANCE CONTRASTS

If the evaluation crosses $F_i \in \{f_0, f_1\}$ with policy assignment under the same matched-block conditions, the same argument applied to cell means identifies the affordance main effect at fixed pol-

icy assignment and the policy–affordance interaction. A prespecified weighted average over policy-assignment cells likewise identifies $\tau^{F,h}$ by linearity. A diagonal complete-system comparison contains only (L, f_0) and (S, f_1); without additional assumptions it cannot identify either main effect. This is why the sign-reversal simulation changes the factorial support rather than only the number of repetitions.

B Synthetic validation

We simulate known cell means with Gaussian noise, including the 2×2 sign-reversal setting in Section 2.2. No LLM or statistical decision policy is run.

Each of 2,000 repetitions contains 32 simulated blocks. Block effects have standard deviation 8 and cell noise has standard deviation 5. The practical-loss tolerance is $\delta = 5$ points. The partial estimator receives half of the planted affordance change, corresponding to an incomplete policy replacement. Table 4 reports all three regimes. The planted (τ^P, τ^F) pairs are $(8, 0)$, $(8, 10)$, and $(-5, 20)$ for no confounding, inflation, and sign reversal, respectively. "Wrong conclusion" counts a definitive verdict opposite to the verdict implied by the true effect relative to δ; inconclusive verdicts are not counted as errors. This differs from an incorrect point-estimate sign or a confidence interval on the wrong side of zero.

regime	comparison	bias	RMSE	coverage	wrong conclusion
no confounding	complete systems	+0.05	1.23	95.3%	0.0%
	partial	−0.01	1.26	94.9%	0.0%
	MPI	+0.04	0.89	95.1%	0.0%
inflation	complete systems	+10.01	10.08	0.0%	0.0%
	partial	+5.03	5.17	2.0%	0.0%
	MPI	−0.00	0.86	95.5%	0.0%
sign reversal	complete systems	+20.01	20.05	0.0%	100.0%
	partial	+10.00	10.08	0.0%	1.6%
	MPI	+0.02	0.88	95.1%	0.0%

Table 4: Synthetic recovery across 2,000 repetitions per regime. In the inflation regime, all definitive conclusions point in the correct direction despite biased effect magnitude; coverage exposes the attribution error. In the sign-reversal regime, the complete-system comparison also changes the decision.

In the sign-reversal setting, MPI recovers the planted effect with no sign errors, whereas the complete-system estimate has the wrong sign in all 2,000 repetitions. Bias and coverage are reported in Table 4.

The simulation uses the fixed random seed 20260824.

C Held-out interface check

The CausalGame acquisition check prespecifies an offline display-only policy before confirmatory data collection. Applied to the displayed strings, it chooses: (i) the lowest-index screening row if one exists; otherwise (ii) the lowest-index maximizer of the displayed exploration score when that score exceeds the frozen threshold 0.004; otherwise (iii) the lowest-index maximizer of the displayed exploitation score. The loss is strict 0/1 disagreement with the statistical policy's paired recommendation; printed-precision ties are not excused. The pass rule is a one-sided exact 95% upper confidence bound no larger than $\varepsilon = .05$.

The confirmatory data contain 128 sessions on `antenna_trap` and `antenna_trap_simpsons_paradox`, seeds 9000–9015, four policy-assignment cells, and 1,280 acquisition decisions. Outcome fields are stripped before applying the check. There are 12 mismatches (0.94%) and 35 displayed ties. The exact upper bound is 1.51%, so the check passes under the registered binomial calculation. Each scene contributes 6 mismatches among 640 decisions. That calculation treats decisions as independent; Appendix F.2 additionally reports a post hoc seed-clustered sensitivity analysis.

This check evaluates action agreement at the displayed precision. It asks whether the rendered CausalGame record reproduces the logged statistical recommendation. The check is offline and does not alter either policy's inputs.

For provenance, the earlier exploratory check is retained in the dated archive. Its held-out half contains 10 mismatches among 320 decisions (3.12%), with a 5.24% upper bound, and therefore fails the same rule. The formal check uses fresh decisions without changing the reconstruction rule, 0/1 mismatch loss, or tolerance after this failure.

D Sequential results at the five-point tolerance

All effects below are statistical-policy minus LLM-policy outcome in percentage points. CausalGame bounds use Bonferroni simultaneity across ten declared rows ($\alpha = .05$). The CausaLab experiment uses four policy-effect rows with $\alpha = .025$ allocated to its initial evaluation. Claude uses $\alpha = .05$ across its two stages. The decision rule follows Section 3.3 with $\delta = 5$.

system	model	family	stage	n blocks	estimate	sim. bounds	conclusion
CausalGame	GPT-5.4-mini	mech.	acq.	192	+6.00	[+2.82, +9.18]	inconclusive
CausalGame	GPT-5.4-mini	mech.	sub.	192	+0.75	[−2.30, +3.80]	delegate
CausalGame	GPT-5.4-mini	no-win	acq.	96	+0.34	[−1.38, +2.06]	delegate
CausalGame	GPT-5.4-mini	no-win	sub.	96	−0.31	[−2.53, +1.92]	delegate
CausalGame	GPT-5.5	mech.	acq.	96	+0.27	[−4.13, +4.67]	delegate
CausalGame	GPT-5.5	mech.	sub.	96	−0.33	[−4.85, +4.19]	delegate
CausalGame	GPT-5.6-sol	mech.	acq.	192	+0.69	[−2.27, +3.65]	delegate
CausalGame	GPT-5.6-sol	mech.	sub.	192	+0.91	[−1.66, +3.48]	delegate
CausalGame	GPT-5.6-sol	no-win	acq.	96	−0.93	[−2.81, +0.96]	delegate
CausalGame	GPT-5.6-sol	no-win	sub.	96	−0.75	[−3.06, +1.55]	delegate
CausaLab	GPT-5.4-mini	all	acq.	32	+0.00	[−4.66, +4.66]	delegate
CausaLab	GPT-5.4-mini	all	sub.	32	−3.12	[−11.02, +4.77]	delegate
CausaLab	GPT-5.6-sol	all	acq.	32	−1.56	[−4.75, +1.63]	delegate
CausaLab	GPT-5.6-sol	all	sub.	32	−3.12	[−8.59, +2.34]	delegate
Claude grid	Haiku 4.5	mech.	acq.	48	+1.05	[−3.80, +5.90]	inconclusive
Claude grid	Haiku 4.5	mech.	sub.	48	−4.06	[−9.53, +1.40]	delegate

Table 5: Results at the five-point tolerance for the sequential studies: 14 support delegation to the LLM and two are inconclusive relative to five points. No lower bound exceeds five points; this does not imply absence of a statistical-method advantage (mini acquisition has bounds $[2.82, 9.18]$). CausalGame uses a fixed common interface, and CausaLab uses common realized menus at both stages.

For CausalGame, GPT-5.4-mini and GPT-5.6-sol combine seeds 400–431 with the independent rerun at seeds 8000–8031; GPT-5.5 uses the powered extension only. The historical grid used in Table 1 is not pooled into these stage-effect estimates. The superseded CausaLab comparison is also excluded.

E Paired-recommendation differences

The smaller submission denominators reflect unavailable paired recommendations, not missing outcome episodes. Outcome analyses retain ITT scores from complete blocks. Transport errors use the frozen statistical fallback; invalid active LLM choices use the specified default or failure score. The failure-free CARE and CausaLab runs reported below are separate from the CausalGame runs described here.

In the CausaLab environment, acquisition disagreement is .613 for GPT-5.4-mini and .742 for GPT-5.6-sol, with at least one disagreement in 100.0% and 98.4% of episodes. Submission index disagreement is .981/.832, but value-level disagreement is .227/.109 because the weighted-average and top candidate options often coincide numerically. These distinctions were prespecified before analysis.

system	model / family	stage	decisions	ρ	episodes with any
CausalGame	GPT-5.4-mini / mech.	acq.	7,666	.639	1.000
CausalGame	GPT-5.4-mini / mech.	sub.	767	.034	.034
CausalGame	GPT-5.4-mini / no-win	acq.	3,822	.536	1.000
CausalGame	GPT-5.4-mini / no-win	sub.	382	.094	.094
CausalGame	GPT-5.5 / mech.	acq.	3,591	.748	1.000
CausalGame	GPT-5.5 / mech.	sub.	364	.066	.066
CausalGame	GPT-5.6-sol / mech.	acq.	7,311	.684	.997
CausalGame	GPT-5.6-sol / mech.	sub.	750	.064	.064
CausalGame	GPT-5.6-sol / no-win	acq.	3,826	.550	1.000
CausalGame	GPT-5.6-sol / no-win	sub.	384	.195	.195
Claude grid	Haiku 4.5 / mech.	acq.	1,919	.721	1.000
Claude grid	Haiku 4.5 / mech.	sub.	191	.058	.058

Table 6: Index-level disagreement between the paired recommendations. Only records with a parsed LLM recommendation and both recommendations available enter these behavioral denominators. Submission has one decision per eligible episode, so its decision and episode rates coincide.

F Affordance ablation and cell means

F.1 Paired screening-vs-D-optimal ablation (CausalGame, statistical policy)

The paired ablation uses the same frozen statistical policy, downstream machinery, and seeds (6000–6011) in both conditions. The only change is balanced-screening rather than D-optimal initialization. The benchmark-defined victory threshold is 75% in both scenes.

Scene	Cond.	Wins	Mean surv.	SD	Paired Δ mean (95% CI)
antenna_trap	screening	8/12	66.0%	26.3	+1.9 [−18.3, +22.1]
antenna_trap	D-optimal	3/12	64.1%	13.5	
antenna_trap_high_def	screening	7/12	64.9%	24.3	+3.9 [−4.8, +12.5]
antenna_trap_high_def	D-optimal	3/12	61.0%	19.3	

Table 7: Paired affordance ablation. Mean-difference intervals use paired t critical values with 11 degrees of freedom.

Pooled victory rate: 15/24 (62.5%) screening vs 6/24 (25.0%) D-optimal. Discordant pairs: screening-only wins 10, D-optimal-only wins 1 (both won in 5, neither in 8); exact sign-binomial McNemar test, two-sided $p = .0117$. Descriptively, the effect concentrates at the victory threshold rather than in mean survival: balanced screening produces more outcomes above 75%, while the paired mean differences remain small and imprecise.

F.2 Current factorial cell means and diagnostic sensitivity

The current CausalGame analysis contains 672 complete four-cell task–seed blocks; no outcome blocks are missing. The following tables report the cell means underlying the current analyses.

A post hoc sensitivity analysis of the interface check resamples all decisions jointly within each seed, retaining both scenes and all four policy-assignment cells (16 clusters; 20,000 bootstrap draws; RNG seed 20260918). The one-sided 95% percentile upper bound is 1.48%, below the 5% tolerance. This sensitivity accounts for within-seed dependence but is not an exact finite-sample bound or a replacement for the registered analysis.

model	scene	n blocks	S/S	S/L	L/S	L/L
gpt-5.4-mini	antenna trap	64	79.11	77.51	70.69	70.01
gpt-5.4-mini	high defense	64	77.27	76.62	73.32	72.74
gpt-5.4-mini	Simpson's paradox	64	78.41	79.64	74.01	71.80
gpt-5.5	antenna trap	32	78.08	79.36	76.40	75.26
gpt-5.5	high defense	32	73.97	75.28	76.13	78.50
gpt-5.5	Simpson's paradox	32	78.28	77.42	77.72	76.75
gpt-5.6-sol	antenna trap	64	78.59	76.76	77.68	76.17
gpt-5.6-sol	high defense	64	78.21	74.67	79.06	76.48
gpt-5.6-sol	Simpson's paradox	64	78.61	78.49	73.85	77.94

Table 8: Current CausalGame mechanism-rich cell means, combining the powered extension and fresh rerun where available. Survival percentages; cells list acquisition/submission control.

model	menu	S/S	S/L	L/S	L/L
gpt-5.4-mini	neutral	46.88	53.12	43.75	53.12
gpt-5.4-mini	screen	43.75	40.62	43.75	43.75
gpt-5.6-sol	neutral	46.88	53.12	50.00	56.25
gpt-5.6-sol	screen	43.75	43.75	43.75	43.75

Table 9: CausaLab adaptation: accuracy percentages across all eight cells per model, 32 seed blocks.

model	contrast	estimate	pointwise 95% CI
GPT-5.6-sol	$\tau_{\text{acq}}^{P\times F}$	+3.12	$[-1.31, +7.56]$
GPT-5.6-sol	$\tau^{F,h}$	−7.81	$[-22.55, +6.92]$
GPT-5.4-mini	$\tau_{\text{acq}}^{P\times F}$	−3.12	$[-9.50, +3.25]$
GPT-5.4-mini	$\tau^{F,h}$	−6.25	$[-20.91, +8.41]$

Table 10: Menu effects (screened minus neutral) and acquisition–menu interactions in the CausaLab adaptation. These are pointwise intervals, unlike the simultaneous policy-effect bounds.

G CARE: PROTOCOL AND CONFIRMATORY RESULTS

On Minerva, mean final yield is 80.18 for the incumbent, 88.01 for the CARE gate, and 76.99 for the LLM judge. The reproduced CARE system gain is therefore +7.83 points. On the confirmatory seeds 10–29, the gate−judge contrast is +14.35 points (95% CI [+8.34, +20.36], $p = .00008$); the pooled 30-seed contrast is +11.01 with 95% CI [+6.38, +15.65] and is reported only descriptively because seeds 0–9 were inspected before the confirmatory extension. Contrasts use unrounded paired values.

The two gate policies agree on only 18% of Minerva decisions. The LLM judge rejects 132 of 161 Minerva authorization opportunities (82%). On the pre-registered ChemLex replication, the corresponding means are 83.91 (incumbent), 89.62 (gate), and 89.71 (judge). The gate−judge contrast is -0.09 ± 2.29 points ($p = .94$), and the judge rejects 28 of 52 opportunities (54%). Each authorization is counted once per seed, round, and candidate, rather than once per copy in the logs. Intervals use paired t inference with 19 degrees of freedom for the confirmatory Minerva contrast and 29 for the 30-seed contrasts. No controller failures occur on either dataset.

The same replacement is therefore consequential on Minerva and undetected on ChemLex. Because both gate policies receive the same external evidence and binary action support, this is a matched policy contrast between two complete policies. Runs use the released code at commit `9b1c741d` with GPT-5.4-mini. The interim look is disclosed above.

H CAUSALAB ADAPTATION

Our discrete-action CausaLab adaptation is based on the published task specification rather than its released artifact, which was unavailable to us at implementation time. Table 11 compares the published benchmark with our implementation.

dimension	published CausaLab artifact	frozen environment used here
Graph scale	Main suites span 3–7 nodes	One fixed scale, $k = 10$ nodes
SCM families	Linear and quadratic structural-equation families	Deterministic per-node linear or quadratic equations
Initial evidence and budget	Two observational records and $4(k-1)$ intervention budget	Same budget formula; two records and 36 intervention opportunities at $k = 10$
Intervention semantics	Shift-style intervention changes a base term while retaining incoming-parent contributions	Hard single-variable intervention clamps the chosen node and severs its incoming influence for that update
Interface and trace	Embodied DiscoveryWorld interaction with a DSL for trajectory-level hypotheses	Programmatic matched-menu controller interface; no embodied world and no DSL
Disturbances	Includes controlled hidden-disturbance suites	No hidden disturbance; mechanisms are deterministic
Outcome	Reactor-frequency prediction plus graph, equation, and trajectory-level recovery metrics	Binary reactor-frequency accuracy at tolerance 0.5 (the endpoint analyzed here)
Fixed implementation choices	Defined by the released benchmark artifact	Root values Unif$[-3, 3]$, coefficients in $\{-2, -1, 1, 2\}$, intercepts Unif$[-1, 1]$, intervention values clipped to $[-5, 5]$

Table 11: Comparison of the published CausaLab benchmark and our discrete-action adaptation.

The implementation passed oracle, budget-enforcement, and metric checks. In the matched-menu experiment, both policies receive an identical six-option submission menu at identical precision: five candidate predictions and the weighted average as an explicit sixth action. The statistical policy chooses that sixth action; the LLM may choose any menu entry. The experiment contains 512 episodes, has no controller failures, and records paired recommendations at both acquisition and submission. The result therefore concerns control over that fixed six-option menu. The environment was frozen and hashed before the agent runs.