PLUG, PLAY, OR RE-EVALUATE: HARNESS COMPONENTS ACROSS MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

An agent harness is the system around a language model that provides the instructions, tools, context, and controls needed for the model to execute a task. Reusing components of this system for new models could reduce repeated testing and tuning, but comparisons of complete harnesses do not reveal which component matters or whether its effect carries over between models. We study each component by comparing the full harness with versions in which one component is removed, then examine whether a component's effect observed on existing models recurs on a new model. We discover three patterns. First, component effects are sparse across model–task–component combinations: removing a component leaves the recorded result unchanged in 84.3% of 1,120 combinations. Second, components exhibit model-specific compatibility: in a separate four-model analysis, the models whose results change differ across components in 68.3% of 360 model–task cases. Third, cross-model recurrence is limited: among 176 cases in which removing a component changes the target model's result, 94.3% have at most one other model showing a change. These findings show that an effect observed on one model should not be assumed to recur on another, motivating component- and model-specific validation before reuse. Source code is available at `https://anonymous.4open.science/r/harness-transfer-artifacts-2041/`.

1 INTRODUCTION

LLM agents combine a base model with an agent harness comprising prompts, tools, memory, planning, and verification (Lin et al., 2026; Yao et al., 2026). As new models are released in rapid succession, developers must identify harness designs that make effective use of each model's capabilities in its target application setting (Zhang et al., 2026b; Lee et al., 2026b). However, comparing complete harnesses is both costly and difficult to interpret: multiple components often change together, obscuring which component drives a performance difference and whether it is well suited to a new model (Fan et al., 2026). We therefore take a component-level view, evaluating one component at a time (DeepSeek-AI, 2026). Yet repeating this process for every new model remains expensive, motivating our central question: **To what extent can component-level evidence from existing models guide the assessment of a component on a new model?**

Recent work treats harness design as an optimization problem, with Meta-Harness (Lee et al., 2026b), HarnessDev (Wu et al., 2026), and AutoSaddler (Park et al., 2026) searching or evolving complete harnesses from execution traces and feedback, and HarnessCompass (Zhang et al., 2026b) optimizing components separately before integrating them. These studies establish harness design as a meaningful optimization target, but primarily evaluate the assembled system, where component interactions can obscure individual contributions. Controlled studies instead vary planning, action-space, and context-management choices across models, showing that component effects depend on model capabilities and operating conditions (Fan et al., 2026). Yet such variation does not indicate when evidence from one model reliably informs another. The key question is therefore when cross-model observations provide relevant and consistent evidence about a component's behavior on a target model.

In this work, we investigate cross-model component transfer through controlled interventions against a shared reference harness. Our analysis connects where component effects arise, which source

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.

models respond, and how often responses recur across models. We evaluate this evidence through held-out prediction and cost-sensitive screening, linking cross-model response patterns to the practical value of component reuse. Our contributions are threefold:

- **Localized component effects.** Component effects are localized rather than uniform across models and tasks. Under controlled single-component interventions, 84.3% of clean three-target configurations show no observed effect, demonstrating that aggregate harness gains cannot be assumed to generalize across configurations.
- **Component-specific source evidence.** The source models that provide relevant evidence depend on the component being assessed. For a fixed target and task, active source sets differ across components in 246/360 (68.3%) clean four-target pairs, motivating component-wise rather than harness-wide source selection.
- **Limited cross-model recurrence and reuse.** Component effects rarely recur across multiple source models, limiting reliable evidence reuse. Among nonzero target effects, 94.3% have at most one active source, and the rate remains 93.2% with three eligible sources; accordingly, weighting offers only an unresolved gain, while conservative screening lowers equal-cost utility.

2 Related Work

Harness Optimization and Self-Evolution. Harness optimization and self-evolution methods adapt prompts, tools, memory, planning, and verification through execution traces, feedback, search, or iterative refinement (Lin et al., 2026; Zhang et al., 2026a;b; Wu et al., 2026; Park et al., 2026; Lee et al., 2026b;a). Their objective is to improve a harness configuration for a given model and environment, rather than estimate the transferability of each component. Plug-in architectures such as DeepSeek Harness (DeepSeek-AI, 2026) expose these mechanisms as independently selectable and replaceable components . These advances make component-level analysis practical, but optimizing a complete harness does not establish whether an individual component's effect recurs after the base model changes.

Cross-Model Harness Evaluation. Benchmark studies show that harness outcomes depend on the base model, task, environment, and evaluation protocol (Yao et al., 2026; Zhang et al., 2026c; Wang et al., 2026; Han & Sun, 2026; Zheng et al., 2026). Related work examines harness-based cross-model adaptation and full-harness transfer across backbones (Qian et al., 2026; Xu et al., 2026). These evaluations characterize complete configurations. Aggregate performance alone does not reveal which components account for observed gains or whether their effects recur on the same tasks. We isolate component interventions to examine their selective effects, variation across task scenarios, and recurrence across models.

Transferability Estimation and Failure-Guided Decisions. Transferability estimation assesses model or representation suitability for new tasks (Nguyen et al., 2020; Shao et al., 2022; You et al., 2021; Chaves et al., 2024; Ding et al., 2024; Li et al., 2024), while model routing selects models by predicted quality, preference, or cost (Chen et al., 2023; Ong et al., 2025; Agrawal & Gupta, 2025). Failure analysis identifies cross-model failure patterns, trajectory-level causes, and failure taxonomies (Saban, 2026; Tank & Nama, 2026; Rafi et al., 2026; Vinay, 2025; Cemri et al., 2025). Together with work on negative transfer (Zhang et al., 2023), these directions motivate behavioral evidence for transfer decisions. We study individual component effects relative to a shared baseline, testing whether similarity in failure profiles predicts responses to the same intervention and can guide component revalidation on a target model.

3 Method

We study how harness component effects vary across models and whether source responses predict effects on a held-out target. In Section 3.1, we define the problem and available information. We construct paired component effects (Section 3.2), characterize component-specific active source sets (Section 3.3), and analyze recurrence and sensitivity to source weighting (Section 3.4). Section 3.5 defines transfer scores and evaluation procedures for prediction and retrospective screening.

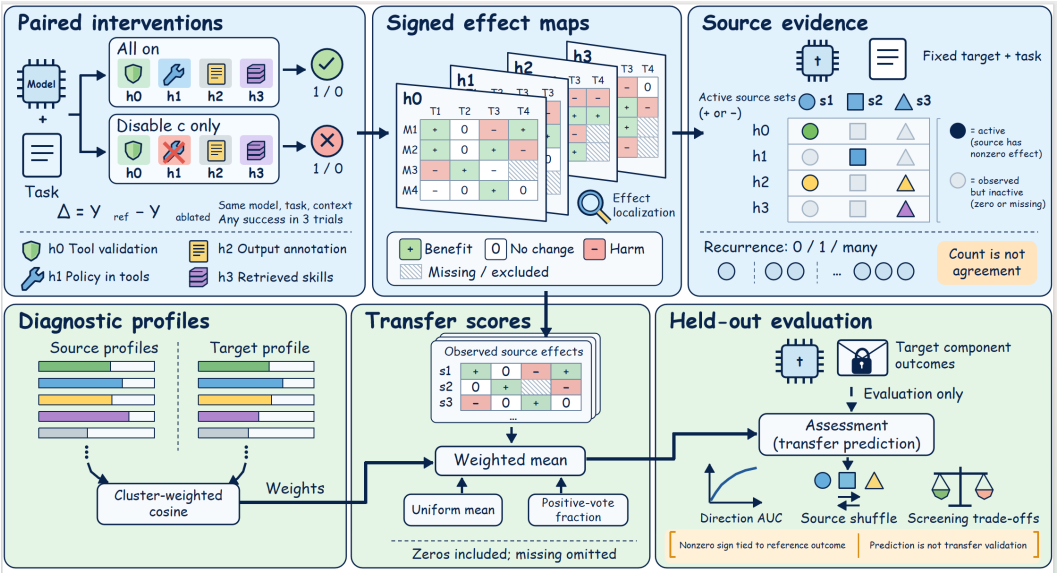

Figure 1: Method overview of our framework for studying cross-model harness component transfer.

3.1 Problem Definition

Here, we define the transfer problem and the information available for scoring. For a held-out target model t, let $s \neq t$ denote an eligible source model; let x be a task instance and c one of the four components in Table 1. For either $m = s$ or $m = t$, a harness setting H yields the binary outcome $Y(m, x, H)$. The paired component effect is $\Delta_{m,x,c} = Y(m, x, H^{\text{ref}}) - Y(m, x, H^{-c})$, where H^{ref} is the all-on reference and H^{-c} disables only c.

The observed source pool $\mathcal{S}^{\text{obs}}_{t,x,c}$ contains eligible non-target models with recorded effects. Observed zeros remain in the pool; missing effects are omitted.

Table 1: Definitions of the four harness components.

	Component	Enabled behavior
	`h0` Pre-execution tool validation	Reject calls violating rules or state constraints.
	`h1` Tool-description policy embedding	Add policies to tool descriptions.
	`h2` Post-execution tool-response annotation	Annotate outputs with state-aware guidance.
	`h3` Retrieved-skill injection	Prepend retrieved skills to the system policy.

Scoring uses source intervention effects and diagnostic profiles of both sources and targets. Target intervention outcomes are reserved for evaluation.

3.2 Controlled Component Intervention Effects

In this subsection, we measure component effects through controlled paired interventions to identify where a harness component changes outcomes. This establishes a common measurement basis for characterizing localized effects across models and tasks. Each intervention disables one component in Table 1. For every model–task–component configuration, we compare the all-on reference H^{ref} with H^{-c} while holding the model, task, and other component switches fixed, measuring the contribution of c within the shared reference harness.

For each harness setting, we aggregate three recorded trials into the binary pass-at-three outcome $Y(m, x, H)$: one if any trial succeeds and zero otherwise. Using the paired effect defined in Section 3.1, each model–task–component configuration yields $+1$, 0, or -1, indicating benefit, no change, or harm from retaining the component. Because the two harness settings use the same model, task, and trial aggregation rule, the comparison isolates the recorded change associated with disabling that component. Tabulating these effects across models and tasks produces component-specific maps that locate responses and supply source observations for the subsequent analyses. For binary outcomes, the reference result fixes the direction of every nonzero effect:

$$\Delta_{m,x,c} \neq 0 \quad \Longrightarrow \quad \Delta_{m,x,c} = 2Y(m, x, H^{\text{ref}}) - 1. \tag{1}$$

A nonzero contrast is positive when the reference succeeds and negative when it fails. This relation connects direction prediction to reference-harness performance and motivates the target-identity diagnostic in Section 3.5. A separate graded-outcome sensitivity analysis examines how the observed patterns depend on binary aggregation.

3.3 Component-Specific Active Source Sets

This subsection characterizes component-specific source evidence by identifying the models with nonzero effects for each component. Comparing these active source sets for a fixed target and task reveals changes in both the number and identity of responding models. We use domain-eligible sources with complete component observations, holding the observed source pool fixed across all four components. The active set for component c is $S_{t,x,c} = \{s \in \mathcal{S}^{\text{obs}}_{t,x,c} : \Delta_{s,x,c} \neq 0\}$. We compare these sets across components and report the fraction of evaluated target–task pairs for which at least two sets differ, separating source-count changes from identity changes at fixed count. Active sets include both positive and negative effects; their variation measures which models respond, while directional agreement is characterized in Section 3.4. Because pairs can share source observations, this proportion is a descriptive frequency over the evaluated population.

3.4 Recurrence and Sensitivity to Source Weighting

In this subsection, we quantify how often a component's effects recur across sources and examine when weighting can change their aggregate direction. This connects the availability and consistency of source evidence to the conditions for cross-model reuse. For a fixed component, the recurrence count is $k_{t,x,c} = |S_{t,x,c}|$. We report k for configurations with a nonzero target effect, together with the eligible source-pool size and the signs of active sources. A one-source pool has $k \leq 1$; reporting counts and signs together distinguishes isolated responses, repeated same-direction responses, and disagreement. For a nonempty observed pool and strictly positive weights $w(s,t)$, the weighted score defined in Section 3.5 has direction

$$\operatorname{sgn}\left(\widehat{\Delta}^{\text{weighted}}_{t,x,c}\right) = \operatorname{sgn}\left(\sum_{\substack{s \in S_{t,x,c} \\ \Delta_{s,x,c}=+1}} w(s,t) - \sum_{\substack{s \in S_{t,x,c} \\ \Delta_{s,x,c}=-1}} w(s,t)\right). \tag{2}$$

Equation 2 follows from the binary effects and positive denominator: zero-effect sources contribute only to normalization. Thus $k = 0$ gives a zero score, while $k = 1$ fixes its sign to that of the sole active source. Positive weighting also preserves the sign when all nonzero source effects agree. Reversing the aggregate direction requires mixed source signs and a change in their weighted balance. A zero score can therefore reflect either no active source or exact cancellation between opposing sources; these cases are distinguished by k. Weighting can still change score magnitudes and rankings without changing signs.

3.5 Transfer Scores and Held-Out Evaluation

In this subsection, we convert component-level source evidence into transfer scores and evaluate its predictive and decision-level value on held-out targets. This connects the structure of cross-model evidence to component reuse through direction prediction and benefit–harm trade-offs. Using the clustered diagnostic representation of Saban (2026), we represent model m by its failure profile $\mathbf{f}_m$ on a common probe suite. Let q_j be the fraction of diagnostic tasks in cluster j among K clusters, and $D = \operatorname{diag}(q_1, \ldots, q_K)$. The cluster-weighted cosine assigns source–target similarity weights:

$$w(s,t) = \frac{\mathbf{f}_s^\top D \mathbf{f}_t}{\sqrt{\mathbf{f}_s^\top D \mathbf{f}_s}\sqrt{\mathbf{f}_t^\top D \mathbf{f}_t}}. \tag{3}$$

Appendix B.2 documents the profiles and validates positive weights. For a nonempty observed source pool, we aggregate component effects as

$$\widehat{\Delta}^{\text{weighted}}_{t,x,c} = \frac{\sum_{s \in \mathcal{S}^{\text{obs}}_{t,x,c}} w(s,t)\Delta_{s,x,c}}{\sum_{s \in \mathcal{S}^{\text{obs}}_{t,x,c}} w(s,t)}. \tag{4}$$

To isolate the role of similarity weighting, we compare this score with the uniform mean over the same source pool:

$$\widehat{\Delta}^{\mathrm{uniform}}_{t,x,c} = \frac{1}{|\mathcal{S}^{\mathrm{obs}}_{t,x,c}|} \sum_{s \in \mathcal{S}^{\mathrm{obs}}_{t,x,c}} \Delta_{s,x,c}. \tag{5}$$

Both means include observed zero effects and omit unavailable effects. To separate the balance of source signs from the prevalence of nonzero responses, we also use the positive-source vote

$$p^{\mathrm{vote}}_{t,x,c} = \begin{cases} \dfrac{|\{s \in S_{t,x,c} : \Delta_{s,x,c} = +1\}|}{k_{t,x,c}}, & k_{t,x,c} > 0, \\ 1/2, & k_{t,x,c} = 0. \end{cases} \tag{6}$$

Voting uses only active sources; $1/2$ denotes no directional evidence. For binary effects, the uniform score equals the active fraction $k_{t,x,c}/|\mathcal{S}^{\mathrm{obs}}_{t,x,c}|$ times $(2p^{\mathrm{vote}}_{t,x,c} - 1)$. The two scores therefore agree on direction relative to their neutral scores, but their rankings can differ with source activity.

We evaluate $\widehat{\Delta}^{\mathrm{weighted}}$, $\widehat{\Delta}^{\mathrm{uniform}}$, and p^{vote} using direction AUC on nonzero held-out target effects. The target-identity score provides an in-sample diagnostic of the relationship between reference performance and effect direction in Equation 1; it uses different information from the source-based scores. Source shuffling tests sensitivity to source assignment by reassigning observed effects to weights while preserving source-effect multisets and missing masks. This probes the sign balance in Equation 2 without changing recurrence. Scores are rounded to 12 decimal places before tie-sensitive evaluation. For retrospective screening, we retain configurations with $\widehat{\Delta}^{\mathrm{weighted}} \geq 0$ (active-only) or $\widehat{\Delta}^{\mathrm{weighted}} > 0$ (conservative), with zero denoting abstention. Configuration-level utility is $U(\lambda) = N_{\mathrm{kept},+} - \lambda N_{\mathrm{kept},-}$ for harm penalty $\lambda \geq 0$. Relative to unfiltered utility U_{all} on the same population, screening changes utility by

$$U(\lambda) - U_{\mathrm{all}}(\lambda) = \lambda N_{\mathrm{blocked},-} - N_{\mathrm{blocked},+}. \tag{7}$$

Here $N_{\mathrm{kept},\pm}$ and $N_{\mathrm{blocked},\pm}$ count retained and blocked configurations of each observed sign. Equation 7 separates avoided harm from discarded benefit. When $N_{\mathrm{blocked},-} > 0$, screening improves utility exactly when $\lambda > N_{\mathrm{blocked},+}/N_{\mathrm{blocked},-}$. This break-even ratio complements AUC by identifying when a screening policy improves configuration-level utility.

4 Experiments

We conduct paired interventions on four components (Table 1) across airline, retail, and telecom tasks using GPT-5.5 (OpenAI, 2026b), GPT-5.4-mini (OpenAI, 2026a), Qwen3-4B-AWQ (Qwen Team, 2025), and Llama-3.1-8B-Instruct (Grattafiori et al., 2024). We analyze component effects and source responses, followed by held-out prediction and screening. We address the following research questions:

- **RQ1: How Localized Are Recorded Harness Intervention Effects?** We characterize effect sparsity and concentration across models, tasks, and components.
- **RQ2: Do Active Source-Model Sets Vary Across Components?** We compare active source sets within the same target–task pair.
- **RQ3: How Much Recurrence Supports Prediction and Screening?** We test whether cross-model recurrence enables held-out prediction and retrospective screening.

4.1 Experimental Setup

Each row is a paired model–task–component intervention with the binary pass-at-three outcome (Section 3.2). The matrix has 1,581 observed effects, with zeros distinct from unavailable entries. Removing 141 simulator-confounded Qwen/Llama airline observations gives 1,440 rows for four-target RQ2; dropping Qwen as target gives 1,120 rows for three-target RQ1/RQ3 (Table 2). Qwen is dropped because 38/120 telecom reference trials (31.7%) hit the 50-step cap, but remains a retail/telecom source (protocol: Appendices A.1 and B). RQ3 direction AUC uses nonzero held-out effects; pooled 95% CIs use 2,000 fixed-score, task-clustered bootstrap resamples.

4.2 How Localized Are Recorded Harness Intervention Effects?

Only 176/1,120 clean three-target configurations (15.7%) change the recorded outcome; 84.3% are zero (Figure 2). Responsiveness varies across targets and components, with the largest target-level activity on GPT-5.4-mini. These descriptive fractions locate intervention responses within the observed configuration matrix.

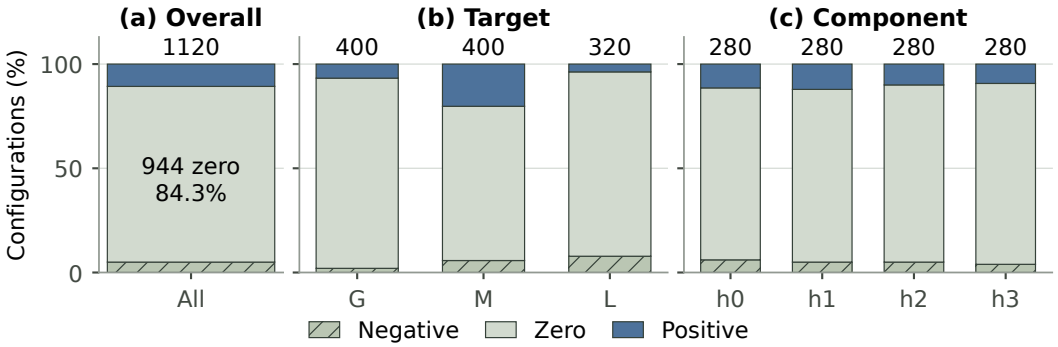

Figure 2: Recorded effect composition in the clean three-target frame. Numbers above bars are configuration counts. Hatching and color distinguish negative, zero, and positive effects; the target and component panels are marginals of the same 1,120 rows. G: GPT-5.5; M: GPT-5.4-mini; L: Llama.

Table 3 separates responsiveness from effect direction. GPT-5.4-mini has 104/400 nonzero effects (26.0%), compared with 35/400 (8.8%) for GPT-5.5. Llama has more negative than positive effects (25 versus 12), whereas the two GPT targets show the opposite balance. Component-level nonzero rates range from 13.2% to 17.5%; each component has both beneficial and harmful cases. The activity rates therefore conceal substantial differences in effect direction. Table 3 reports exact marginal counts, and Figure 3 resolves signs within each model–domain–component cell. These are complementary views of the same clean three-target population, not additional independent observations. Measurement sensitivity is discussed in Appendix D.1.

4.3 Do Active Source-Model Sets Vary Across Components?

Active source sets differ across components in 246/360 clean four-target pairs (68.3%; Figure 4). Most variation changes the active-source count: only five pairs switch source identity at fixed set

Table 2: Distinct analysis populations from the same frozen matrix. Nonzero means signal-bearing; the retention frame is used only for labeled robustness checks.

Population	Total	Zero	Positive	Negative	Nonzero
Clean three-target	1120	944	120	56	176
Clean four-target	1440	1169	178	93	271
Retention	1581	1296	182	103	285

Table 3: Recorded intervention effects in the clean three-target population.

Group	N	**Recorded effects**			**Nonzero**	
		Zero	Positive	Negative	Count	Rate (%)
By target						
GPT-5.5	400	365	27	8	35	8.8
GPT-5.4-mini	400	296	81	23	**104**	**26.0**
Llama	320	283	**12**	**25**	37	11.6
By component						
h_0	280	231	32	17	49	17.5
h_1	280	232	34	14	48	17.1
h_2	280	238	28	14	42	15.0
h_3	280	243	26	11	37	13.2
Overall	**1,120**	**944**	120	56	**176**	**15.7**

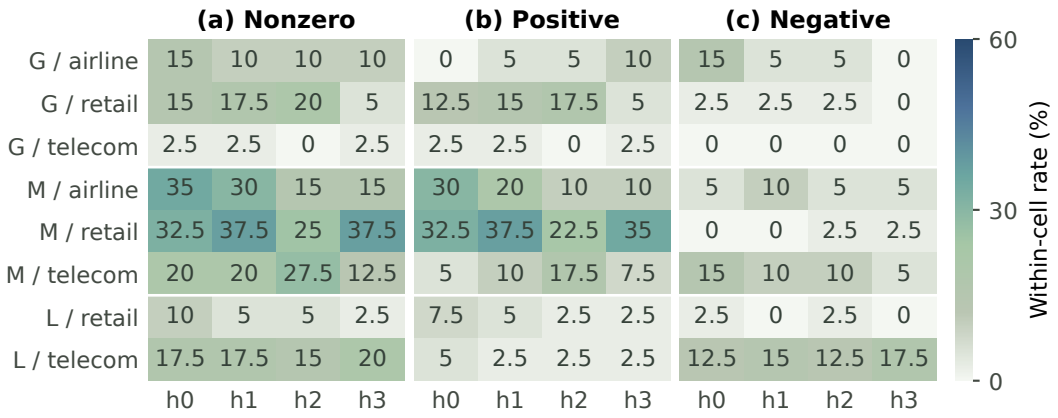

Figure 3: Effect location at finer resolution. Entries are within-cell percentages (G: GPT-5.5; M: GPT-5.4-mini; L: Llama), with 20 airline or 40 retail/telecom tasks per retained model and component. A common color scale permits sign comparisons. Excluded Llama/airline observations are absent, not zero.

size. The clean three-target check is similar (193/280, 68.9%). Thus, components activate different source sets, whose effects may have either sign.

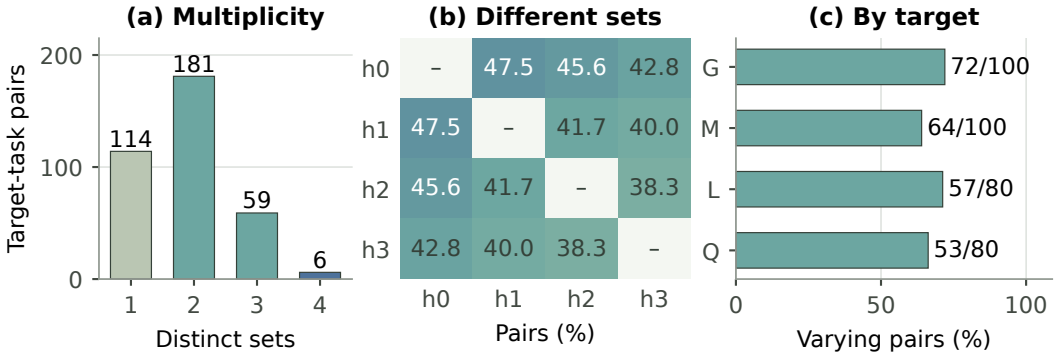

Figure 4: Component-specific active source sets in the clean four-target frame. (a) Number of distinct sets across four components. (b) Fraction of 360 pairs with different source sets for each component pair. (c) Variation within each held-out target's eligible task population (G: GPT-5.5; M: GPT-5.4-mini; L: Llama; Q: Qwen).

Figure 5 distinguishes active sources, observed zeros, and exclusions. Table 4 shows that 241 of 246 varying pairs change source count; only five change identity at fixed count. Variation is high in retail (73.8%) and telecom (70.6%); airline is not directly comparable because only one source is eligible. Because target pools overlap, these rates are descriptive; Appendix D.2 provides pairwise comparisons.

Table 4: Active source-set variation in the clean four-target population; domain rates reflect different eligible source-pool sizes.

Group	Target / domain	Vary n/N (%)	Count Δ	ID-only
Target	GPT-5.5	72/100 (72.0)	69	3
	GPT-5.4-mini	64/100 (64.0)	63	1
	Llama	57/80 (71.2)	56	1
	Qwen	53/80 (66.2)	53	0
Domain	Airline	15/40 (37.5)	15	0
	Retail	118/160 (73.8)	117	1
	Telecom	113/160 (70.6)	109	4
Overall		**246/360 (68.3)**	**241**	**5**

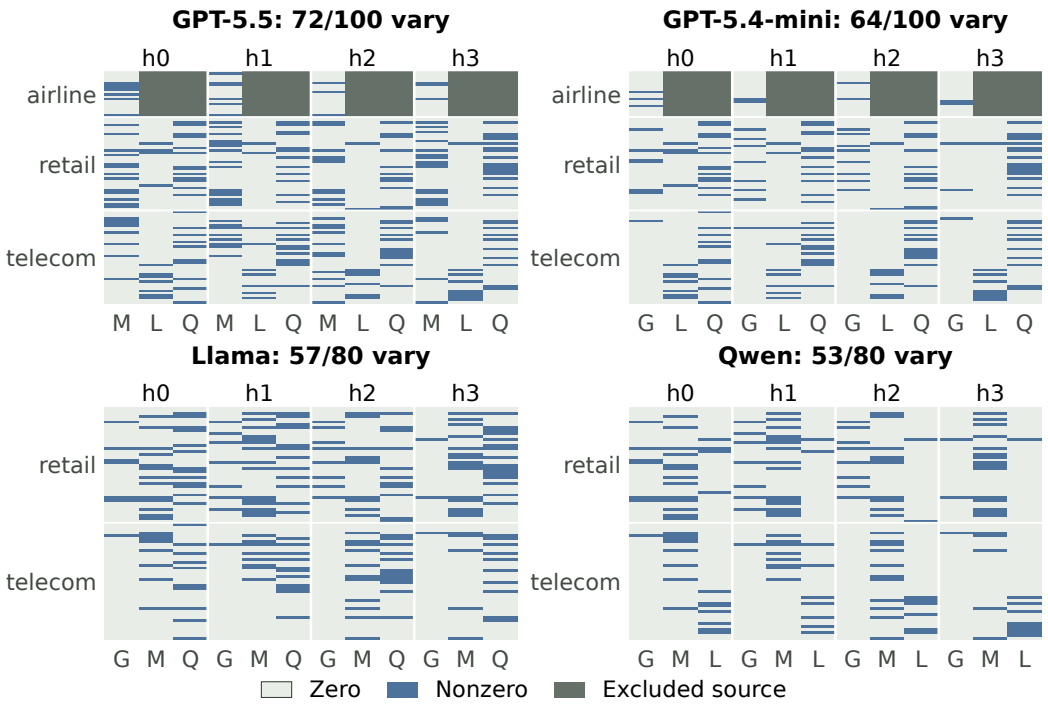

Figure 5: Active-source membership by target. Rows group tasks by domain and columns group sources by component. Blue/light gray/dark gray mark nonzero/zero/excluded entries; G/M/L/Q denote the four models.

4.4 How Much Recurrence Supports Prediction and Screening?

At most one source is active in 166/176 signal-bearing clean three-target configurations (94.3%; Figure 6). Airline necessarily satisfies this bound because its source pool has size one. The scarcity persists in the three-source retail and telecom stratum: 138/148 (93.2%) still have at most one nonzero source. Seven of the 10 multisource rows have mixed signs. Hence recurrence is limited even where multiple sources are available, and repeated activity need not imply consensus. Table 5 separates domain and retention populations.

Table 5: Nonzero source count k for signal-bearing target configurations. The clean-domain rows partition the headline population; retail + telecom is a restricted stratum. Retention is a separate robustness population.

Population / domain	N	k=0	k=1	k=2	k=3	$k \leq 1$ (%)
Clean three-target	176	99	67	7	3	94.3
airline	28	22	6	0	0	100.0
retail	85	35	43	7	0	91.8
telecom	63	42	18	0	3	95.2
Retail + telecom	148	77	61	7	3	93.2
Retention (separate)	285	160	112	9	4	95.4

Profile weighting has only a small, unresolved pooled advantage over uniform averaging (Table 6): the AUC difference is 0.0063, with a paired task-clustered 95% interval of $[-0.0008, 0.0185]$. All three source-based point estimates are below 0.5 in telecom; we treat this domain-level pattern as descriptive because no domain-specific intervals are reported. The target-identity diagnostic is competitive overall (0.6712) and ranks highest in telecom, consistent with the shared-reference label constraint in Section 3.2; its different inputs preclude an equal-budget comparison. Taken together, these diagnostics do not establish reliable prospective selection from source-based scores alone. The separate retention-frame permutation audit below probes sensitivity to source assignment.

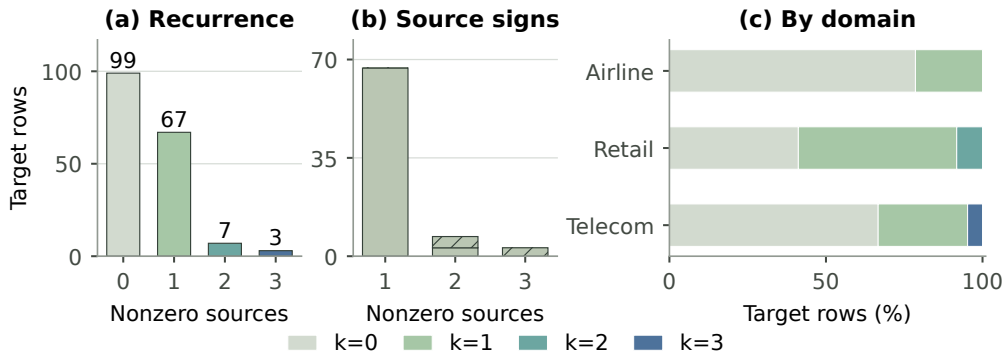

Figure 6: Recurrence and source signs for nonzero target effects. (a) Active-source count. (b) One-sign (solid) versus mixed-sign (hatched) sets. (c) Domain composition (28 airline, 85 retail, 63 telecom rows); airline has one eligible source. Colors encode k in (a) and (c).

Table 6: ROC–AUC and pooled task-clustered 95% intervals on nonzero effects. Airline, retail, and telecom contain 28, 85, and 63 evaluated rows respectively.

Predictor	Airline	Retail	Telecom	Pooled	95% interval
Profile weighted	0.733	0.708	0.439	0.6737	[0.573, 0.762]
Unweighted	0.733	0.703	0.436	0.6673	[0.562, 0.757]
Active-source fraction	0.733	0.738	0.443	0.6709	[0.571, 0.758]
Target identity (diag.)	0.361	0.495	0.716	0.6712	[0.540, 0.808]

Internal source-permutation check. In the separate retention frame, an exhaustive $6^4 = 1{,}296$ effect-to-weight reassignment check leaves score signs unchanged in 281/285 nonzero rows. It preserves source-effect multisets and missing-observation masks, without rerunning models. At the decision-sign level, this audit shows limited sensitivity to source assignment in the retention frame.

Decision-level implications. Conservative screening, which retains only positive scores, blocks 51/56 negative-effect configurations but also discards 72/120 positive-effect configurations. Equal-cost net utility consequently falls from 64 to 43 (Table 7); the policy exceeds unfiltered utility only when the harm-to-benefit cost ratio is above 1.41. Most blocked negatives arise from zero-score abstention, while strictly negative flags have only 5.9% precision. This utility is a retrospective account of individual configurations: marginal effects need not compose. The full cost curves are in Appendix E.4; additional robustness and feasibility checks are summarized in Appendix F.

Table 7: Screening counts in the clean three-target frame. Signs denote observed effects; $U(1)$ is retained positives minus retained negatives.

Policy	Kept +	Kept -	Blocked +	Blocked -	U(1)
None	120	56	0	0	+64
Active-only	107	45	13	11	+62
Conservative	48	5	72	51	+43

5 Conclusion

Our component-level analysis identifies three properties of the observed harness interventions: effects are localized; active source-model sets vary across components in 246/360 (68.3%) target–task pairs in the separate clean four-target analysis; and responses to the same component have limited cross-model recurrence in the reported signal-bearing populations. The existing predictor comparisons show only a small pooled advantage from profile weighting over simple aggregation on the clean three-target population, while conservative screening sacrifices equal-cost net utility. Shared-baseline direction coding further limits the interpretation of prediction AUC as evidence of component-specific transfer. Together, these findings support analyzing component responses separately and accounting for sparse recurrence when interpreting transfer predictions.

6 AI USE STATEMENT

This work was produced with substantial assistance from large language model (LLM) based agents.

LLM-based agents were used for the following subtasks. **Designing methodology and experiments:** the ablation structure, the probe-suite protocol, and the definition of the matched population used in §F.2 were drafted by agents and approved by the authors. **Implementing methods:** the evaluation harness, the profile-extraction and consensus code, and the bootstrap analysis were written by agents. **Analysing and interpreting results:** agents computed the reported statistics and produced the first interpretation of them, including the matched-population recomputation and the interval diagnostics reported in §F.2. **Drafting:** all sections of this manuscript, including this statement, were drafted and revised by agents. No synthetic data was generated; every number reported here derives from executed evaluations.

The authors set the research direction, ruled on framing and on which claims the evidence supports, and reviewed quantitative claims against the recorded artifacts. Each quantitative claim in this paper is linked to a stored experimental artifact in a provenance record maintained alongside the manuscript.

Verification identified a population mismatch and sensitivity to the resampling procedure in the source-selection comparison (§F.2). The comparison was withdrawn under a decision rule fixed before the matched estimate was computed. Each bibliography entry was checked by resolving its identifier against the source record independently of the entry's own metadata. All but two of the entries carry an identifier and were resolved that way; the remaining two (Condorcet 1785; Zhang et al. 2023) carry none and were checked by title against the source record instead. Entries found to be inaccurate were corrected against the resolved record.

7 REPRODUCIBILITY STATEMENT

Code, benchmark configurations, evaluation logs, failure profiles, and the toggle-effect matrix are included in the supplementary material and will be released upon acceptance. Section 4.1 specifies the models, benchmarks, scoring rule, and evaluation protocol; Appendix B.2 reports the LOMO, UMAP, and HDBSCAN configuration, with additional data-quality controls documented in the appendix. Local model inference used two RTX 4090 GPUs, while the downstream analysis is CPU-only. The primary numerical freeze, SHA-256 identifiers, exclusions, source-pool rules, score definitions, and 2,000-replicate task-cluster bootstrap with `seed=42` are specified in Appendix A.1. Separate archived robustness experiments retain their own explicitly stated populations and resampling conventions.

8 ETHICS STATEMENT

All experiments use simulated service environments provided by τ-bench, with model-driven users. The study involves no human participants, personally identifiable information, or sensitive data.

The main deployment risk is interpreting direction prediction as component-specific harm detection. Under the binary shared-baseline construction, the direction of a nonzero effect is determined by the target's reference outcome. On the clean three-target population, conservative screening lowers equal-cost net utility ($+43$ against $+64$ for unfiltered transfer), and exceeds unfiltered transfer only when a harm costs more than $1.41\times$ a missed improvement. These evaluations measure retrospective screening utility; deployment safety remains untested. Transfer evaluations should verify that the outcome label represents the intended decision and assess screening under an explicit loss model.

Model evaluations run through standard API access and local inference under the providers' terms of service. The nominal five-model design schedules 7,500 simulations of service tasks, including the shared probe baseline; this count excludes retries. The downstream analysis is CPU-only. We release code and data on acceptance, including the negative results and the withdrawn comparison.

REFERENCES

Shubham Agrawal and Prasang Gupta. LLMRank: Understanding LLM strengths for model routing. *arXiv preprint arXiv:2510.01234*, 2025. URL https://arxiv.org/abs/2510.01234.

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

Levy Chaves, Eduardo Valle, Alceu Bissoto, and Sandra Avila. Back to the basics on predicting transfer performance. *arXiv preprint arXiv:2405.20420*, 2024. URL https://arxiv.org/abs/2405.20420.

Lingjiao Chen, Matei Zaharia, and James Zou. FrugalGPT: How to use large language models while reducing cost and improving performance. *arXiv preprint arXiv:2305.05176*, 2023. URL https://arxiv.org/abs/2305.05176.

DeepSeek-AI. DeepSeek Harness: Everything is a Plugin. https://github.com/deepseek-ai/deepseek-harness, 2026.

Yuhe Ding, Bo Jiang, Aijing Yu, Aihua Zheng, and Jian Liang. Which model to transfer? a survey on transferability estimation. *arXiv preprint arXiv:2402.15231*, 2024. URL https://arxiv.org/abs/2402.15231.

Run-Ze Fan, Zihao Zhang, Simin Ma, Yebowen Hu, Shouju Wang, Kaiqiang Song, Fei Liu, Hamed Zamani, and Xiaoyang Wang. An empirical study of harness design for coding agents. *arXiv preprint arXiv:2609.20804*, 2026. URL https://arxiv.org/abs/2609.20804.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. URL https://arxiv.org/abs/2407.21783.

Jiwoo Han and Yuekai Sun. How good is your harness? In *Proceedings of the 43rd International Conference on Machine Learning (ICML 2026)*, 2026. URL https://icml.cc/virtual/2026/68323.

Hyunin Lee, Jinglue Xu, Jeffrey Seely, Donghyun Lee, Matei Zaharia, and Yujin Tang. Recursive harness self-improvement. *arXiv preprint arXiv:2607.15524*, 2026a. URL https://arxiv.org/abs/2607.15524.

Yoonho Lee, Roshen Nair, Qizheng Zhang, Kangwook Lee, Omar Khattab, and Chelsea Finn. Meta-Harness: End-to-end optimization of model harnesses. *arXiv preprint arXiv:2603.28052*, 2026b. URL https://arxiv.org/abs/2603.28052.

Ziyu Li, Hilco van der Wilk, Danning Zhan, Megha Khosla, Alessandro Bozzon, and Rihan Hai. Model selection with model zoo via graph learning. In *Proceedings of the IEEE International Conference on Data Engineering (ICDE)*, 2024. URL https://arxiv.org/abs/2404.03988.

Jiahang Lin, Shichun Liu, Chengjun Pan, Lizhi Lin, Shihan Dou, Zhiheng Xi, Xuanjing Huang, Hang Yan, Zhenhua Han, Tao Gui, and Yu-Gang Jiang. Agentic harness engineering: Observability-driven automatic evolution of coding-agent harnesses. *arXiv preprint arXiv:2604.25850*, 2026. URL https://arxiv.org/abs/2604.25850.

Cuong V. Nguyen, Tal Hassner, Matthias Seeger, and Cedric Archambeau. LEEP: A new measure to evaluate transferability of learned representations. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2020. URL https://arxiv.org/abs/2002.12462.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E. Gonzalez, M Waleed Kadous, and Ion Stoica. RouteLLM: Learning to route LLMs with preference data. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025. URL https://arxiv.org/abs/2406.18665.

OpenAI. Introducing GPT-5.4 mini and nano, March 2026a. URL https://openai.com/index/introducing-gpt-5-4-mini-and-nano/.

OpenAI. Introducing GPT-5.5, April 2026b. URL https://openai.com/index/introducing-gpt-5-5/.

Sungho Park, Wonjoong Kim, Rongyuan Tan, Jue Zhang, Wook-Shin Han, Pengfei Gao, Chanyoung Park, Yongqiang Yao, Rao Fu, Elsie Nallipogu, Qingwei Lin, Saravan Rajmohan, and Dongmei Zhang. AutoSaddler: Automatic harness optimization with durable updates from agent execution traces. *arXiv preprint arXiv:2608.23041*, 2026. URL https://arxiv.org/abs/2608.23041.

Cheng Qian, Wenting Zhao, Liangwei Yang, Heng Wang, Jielin Qiu, Heng Ji, Silvio Savarese, Huan Wang, and Shelby Heinecke. AI4AI at test-time: Strong-to-weak capability transfer via harnesses. *arXiv preprint arXiv:2608.12307*, 2026. URL https://arxiv.org/abs/2608.12307.

Qwen Team. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

Md Nakhla Rafi, Md Ahasanuzzaman, Dong Jae Kim, Zhijie Wang, and Tse-Hsun Chen. FALAT: Tracing failures in LLM agent trajectories via dependency-guided search. *arXiv preprint arXiv:2606.00765*, 2026. URL https://arxiv.org/abs/2606.00765.

Nicholas Saban. FailureScope: Cross-Regime behavioral diagnosis of language model weaknesses. *arXiv preprint arXiv:2606.09878*, 2026. URL https://arxiv.org/abs/2606.09878.

Wenqi Shao, Xun Zhao, Yixiao Ge, Zhaoyang Zhang, Lei Yang, Xiaogang Wang, Ying Shan, and Ping Luo. Not all models are equal: Predicting model transferability in a self-challenging fisher space. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 286–302, 2022. URL https://arxiv.org/abs/2207.03036.

Darshan Tank and Baran Nama. The regression tax: Decomposing why skills help and hurt LLM agents. *arXiv preprint arXiv:2607.22520*, 2026. URL https://arxiv.org/abs/2607.22520.

Vaishali Vinay. Failure modes in LLM systems: A system-level taxonomy for reliable AI applications. *arXiv preprint arXiv:2511.19933*, 2025. URL https://arxiv.org/abs/2511.19933.

Yike Wang, Huaisheng Zhu, Zhengyu Hu, Yige Yuan, Zhengyu Chen, Shakti Senthil, Hannaneh Hajishirzi, Yulia Tsvetkov, Pradeep Dasigi, and Teng Xiao. Rethinking the evaluation of harness evolution for agents. *arXiv preprint arXiv:2607.12227*, 2026. URL https://arxiv.org/abs/2607.12227.

Yuhao Wu, Jingyuan Zhang, Jiajun Shi, Xinping Lei, Qingshui Gu, Yuxuan Zhang, Zexuan Wang, Chen He, Chen Huang, Maojia Song, Zhiyuan Zeng, Shaowen Wang, Jinkai Liu, Yunfeng Shi, Jiaheng Liu, Shen Yan, Wenhao Huang, Ge Zhang, and Wenxuan Zhang. HarnessDev: Can LLMs create and evolve their own agent harness? *arXiv preprint arXiv:2609.01437*, 2026. URL https://arxiv.org/abs/2609.01437.

Tianshi Xu, Huifeng Wen, and Meng Li. Adapting the interface, not the model: Runtime harness adaptation for deterministic LLM agents. *arXiv preprint arXiv:2605.22166*, 2026. URL https://arxiv.org/abs/2605.22166.

Yilun Yao, Xinyu Tan, Chao-Hsuan Liu, Yaoming Li, Zhengyang Wang, Wenhan Yu, Zhewen Tan, Yuxuan Tian, Guangxiang Zhao, Lin Sun, Xiangzheng Zhang, and Tong Yang. HarnessBench: Measuring harness effects across models in realistic agent workflows. *arXiv preprint arXiv:2605.27922*, 2026. URL https://arxiv.org/abs/2605.27922.

Kaichao You, Yong Liu, Jianmin Wang, and Mingsheng Long. LogME: Practical assessment of pre-trained models for transfer learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2021. URL https://arxiv.org/abs/2102.11005.

Hangfan Zhang, Shao Zhang, Kangcong Li, Chen Zhang, Yang Chen, Yiqun Zhang, Lei Bai, and Shuyue Hu. Self-harness: Harnesses that improve themselves. *arXiv preprint arXiv:2606.09498*, 2026a. URL https://arxiv.org/abs/2606.09498.

Luan Zhang, Ruochen Zhou, Dandan Song, Zhengyu Chen, Yuhang Tian, Jun Yang, Huipeng Ma, Chenhao Li, Guangyuan Feng, Xudong Li, Yizhou Jin, and Yan Xu. HarnessCompass: Guiding automatic harness evolution toward generalizable and effective agent harnesses. *arXiv preprint arXiv:2608.01918*, 2026b. URL https://arxiv.org/abs/2608.01918.

Wen Zhang, Lingfei Deng, Lei Zhang, and Dongrui Wu. A survey on negative transfer. *IEEE/CAA Journal of Automatica Sinica*, 10(2):305–329, 2023.

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*, 2026c. URL https://arxiv.org/abs/2605.23950.

Congjie Zheng, Chuanyi Xue, Bin Liang, Jun Yang, and Changshui Zhang. SEAGym: An evaluation environment for self-evolving LLM agents. *arXiv preprint arXiv:2606.17546*, 2026. URL https://arxiv.org/abs/2606.17546.

A Data Provenance and Evaluation Populations

A.1 Version authority and evidence boundary

All main-text numerical results refer to the released-summary matrix at manuscript commit `a5ee089`, not to a fully raw-reconciled dataset. The 26 September 2026 audit separately reconstructs all available pairs linked by the existing canonical-table pointers. It does not select alternative runs because their outcomes better match a summary, impute missing trials, or launch new evaluations. Thus there are two named evidence layers: the *released-summary frame* and the *raw-reconstructable subset*. Statistics from different layers must not share a denominator implicitly.

The second server audit recovered the historical canonical-table builder. It prioritized full agreement with summary means, then exact overlap and reward coverage, when choosing among candidate run files. Its mapping lists 92 selected cell–arm entries pointing to 56 unique runs; shared baselines are repeated across cells. Eight entries used a weaker match. The present audit does not repeat that selection, but inherits its pointers. Agreement under this linkage is not independent verification of intended run identity, and does not by itself establish that results were selected for favorable performance.

The canonical table, binary matrix, and failure profiles are frozen with SHA-256 hashes in the accompanying `current_version.json` and source manifest. For the binary matrix, the exact identifier is

`88617f304be4f1f60e8479830a057d10dfc8`
`2d8bf1890e729e6b33e825a38cca`

The first server's rejoined original-results archive has SHA-256

`958b030752d033446c5e1b8a741a4b618`
`6412b2480d42991b0ab887b8f6fbd57`

The archive contains 141 result files. A historical matrix on the second server matches the older graded release; it is not an independent validation of the repaired summary matrix. Checksums establish identity, not correctness.

A.2 Trial-to-summary reconciliation

Existing pointers support paired reconstruction for 1,490 of 1,581 rows. The binary summaries agree in 1,471/1,490 rows and the trial means agree in 1,446/1,490. All available embedded trial arrays match the linked archived trials exactly; therefore the remaining discrepancies cannot be dismissed as merely stale embedded arrays. For 21 of the 44 mean discrepancies, the same-directory run summary also agrees with the raw trials rather than the canonical value; these include eight of the 19 binary discrepancies. These are corroborating representations of the same run, not independent repeats. Table 8 localizes all discrepancies. Eighty rows have summary-only pointers without the

Table 8: Summary disagreements within available linked raw pairs. Components use paper IDs. Binary disagreements are a subset of the graded disagreements.

Model / domain	Component	Binary	Trial mean
GPT-5.5 / airline	h0	2	10
GPT-5.5 / airline	h1	1	3
Qwen / telecom	h3	10	20
Llama / retail	h1	0	2
Llama / telecom	h0	6	9
Total		19	44

corresponding original ablation records in the inspected archive: GPT-5.4-mini/airline h0–h1 and GPT-5.5/airline h2–h3, 20 rows per arm. Eleven more rows lack task trials inside the referenced result files: five GPT-5.5/airline h1 and six Qwen/telecom h3 rows. These 91 rows are unresolved, not zero effects. Among 1,890 distinct observed arm–task records, 17 contain fewer than three gradeable rewards. Available-reward aggregation is reported with that limitation; it is not a complete three-trial success-probability estimate.

A.3 Recorded configuration differences

Configurations are inspectable for 44 of 48 ablation arms. Thirty-six differ from their reference on at least one checked field; eight match the selected fields, which is not full causal certification. GPT-target references record `openai/gpt-5.5` as user simulator while available ablations record `openai/gpt-5.4-mini`. Qwen/Llama airline references record `openai/user-simulator`, unlike their ablations. Qwen/Llama telecom reference step limits are 50, versus 200 for the ablations. Historical routing equivalence for model aliases has not been established. A shared endpoint, a current adapter configuration, or an identical task ID cannot prove equivalence of historical simulators or budgets. The archived `git_commit` field is `unknown`. Consequently, the existing eligibility masks do not create a configuration-matched causal population.

A.4 Population ledger and separate raw reconstruction

The released retention frame contains 1,581 rows, excluding 19 Llama/airline configurations without gradeable outcomes. The linked h3 ablation file contains records for all 20 airline tasks, but only one gradeable reward in total; the other 19 tasks cannot supply outcome labels. Raw-file presence is not outcome availability. Removing 141 flagged Qwen/Llama airline rows from target and source pools gives 1,440 rows. Removing Qwen as target, but not as an otherwise eligible source, gives 1,120 headline rows. These are nested reporting frames, not additive samples. The inherited Qwen-target exclusion is a reporting convention, not evidence that remaining observations are free of simulator or budget differences. The raw retention subset has 1,227/167/96 zero/positive/negative

Table 9: Raw-reconstructable subsets, separate from the released populations in main Table 2. Counts use linked available rewards, including incomplete arms; excluded unresolved rows are not imputed as zero.

Evidence	Population	Rows	Binary nonzero	Graded nonzero
Raw subset	Retention	1,490	263	601
Raw subset	Filtered four-target	1,349	249	570
Raw subset	Filtered three-target	1,035	159	432

binary contrasts; the filtered three-target raw subset has 876/107/52. These are coverage-qualified sensitivity counts, not replacements for the full-matrix AUCs. Excluding unresolved records may itself change the evaluated population.

B Experimental and Scoring Protocol

B.1 Component mapping, observations, and missingness

Implementation IDs h2, h3, h4, h5 map to paper IDs h0, h1, h2, h3, respectively. A record is a model–domain–task–component contrast, not an independent trial. The intended protocol pairs an all-on reference with a single switch disabled over three trials. Binary success is any success; graded success is the mean of gradeable rewards. Null rewards remain missing and are never recoded as failures. A missing arm is distinct from an observed zero contrast. The audit groups trials by the recorded run and exact task, retaining trial IDs, seeds, termination information, and file hashes.

B.2 Frozen profiles and aggregation

The downstream audit uses the released profiles without refitting clusters. Their five cluster sizes are 14, 13, 27, 13, and 33, summing to 100 probes; the archived assignment contains no noise probes. With $q = (14, 13, 27, 13, 33)/100$, failure fractions f_{mk} are aggregated by the cluster-size-weighted cosine in Equation 3. All used pairwise weights are positive. Uniform and weighted averages include observed zero effects in their denominators and omit unavailable sources. The active-source vote is the positive fraction among nonzero sources, returning $1/2$ if none are active. Scores are rounded to 12 decimal places for rank ties and zero-score decisions.

The original profile-construction description refers to leave-one-model-out (LOMO), UMAP, and HDBSCAN. This audit verifies the frozen-profile arithmetic, not every historical clustering setting or

Table 10: Exact within-target ROC–AUC values corresponding to Figure 7. Positive and Negative are the observed nonzero contrast counts. Identity is constant within each target and therefore has AUC 0.5 whenever both signs are represented.

Target	Positive	Negative	Weighted	Uniform	Vote	Identity
GPT-5.5	27	8	0.8472	0.8194	0.8148	0.5000
GPT-5.4-mini	81	23	0.6651	0.6699	0.6656	0.5000
Llama	12	25	0.5433	0.5700	0.5767	0.5000

absence of target-information leakage during representation construction. The old generic parameter recipe is not presented as a fully verified reconstruction specification. Target intervention outcomes are excluded from source aggregation; target baseline probe profiles are available. Outcome holdout must not be confused with an independent target-free representation audit.

B.3 Fixed-score uncertainty and scope

AUC is computed only on nonzero target contrasts. The three-target intervals use 2,000 bootstrap resamples, seed 42, of 100 domain–task clusters, keeping all associated target/component rows together. Scores and profiles remain fixed; these intervals do not propagate new trials, profile estimation, model-pool sampling, or uncertain run identity. Rank-based and pairwise AUC implementations agree. Resampling an incorrect provenance assignment cannot repair it. All new audit computations operate on existing artifacts only.

C Label Coupling and Target-Ability Diagnostics

C.1 Scope of the nonzero-label identity

Equation 1 in the main text gives the shared-reference label identity. Its restriction to nonzero contrasts matters: it does not predict whether a component will change the outcome in the first place. Evaluating occurrence and evaluating direction conditional on occurrence are different tasks. Thus the identity limits the interpretation of nonzero-effect AUC without proving that every predictor is uninformative. Inference is conditional on an outcome-selected set, rather than on all candidate configurations.

C.2 Ability information already available in the profile

Because the five clusters partition the 100 probes, the overall baseline probe pass rate is exactly $a_m = 1 - \sum_k q_k f_{mk}$. For GPT-5.5, GPT-5.4-mini, Llama, and Qwen these values are 0.96, 0.80, 0.15, and 0.46. The ability diagnostic therefore does not require acquiring additional target intervention labels. It remains an in-sample diagnostic, not a validated deployment rule. Being constant within a target, it cannot rank that target's task–component pairs, even when it contributes discrimination after targets are pooled. This distinction separates between-target ability information from within-target component ranking.

C.3 Within-target ranking rather than another pooled comparison

Figure 7 visualizes target-level ranking, and Table 10 reports the exact AUC values and sign counts. The target-identity score is constant within each target and has AUC 0.5 when both effect signs are represented. Source scores vary across tasks and components within a target, although the observed point estimates do not establish consistent improvement from weighting. The sign counts are small and imbalanced, so no per-target significance claim is made. This diagnostic uses the frozen three-target released summaries without additional model evaluations.

The existing paired weighted-minus-identity comparison is 0.0025 with a fixed-score 95% interval of $[-0.1839, 0.1757]$. This wide interval does not establish superiority or equivalence. Numerical proximity in the pooled comparison also does not prove that target identity explains all predictive information.

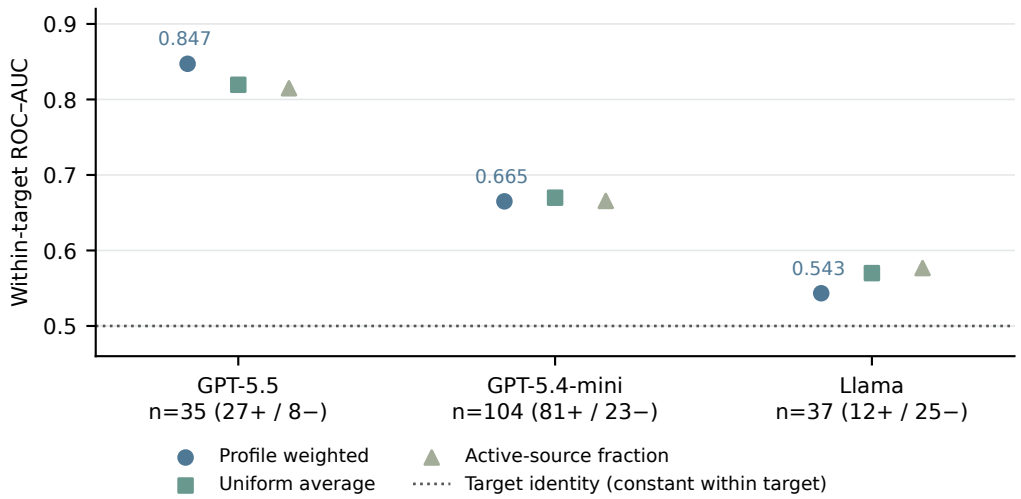

Figure 7: Within-target ranking on nonzero contrasts in the three-target frame. Labels give positive and negative contrast counts. Dots compare source-based score rules within each target; the dashed line is AUC 0.5 for the constant target-identity diagnostic. Conditioning on target removes between-target variation from the pooled comparison. These are descriptive point estimates, not claims of statistically significant differences.

D Outcome Resolution and Source Recurrence

D.1 Outcome-resolution sensitivity across analysis frames

Any-success coding and trial-mean coding resolve different levels of outcome variation. Table 11 places the binary and graded results side by side for the three-target, four-target, and retention frames, so the binary rows supply the denominators for interpreting this sensitivity analysis. Configurations remain fixed within each named frame, while the nonzero subset and its recurrence percentage change with outcome coding. In the retention frame, 356 binary-zero contrasts become graded-nonzero; signs agree on jointly nonzero contrasts, and the current graded nonzero count is 641. Raw-subset counts remain separate in Table 9.

Table 11: Binary and graded outcome resolution in the existing released-summary frames. N counts all configurations; Positive, Negative, and Nonzero refer to target contrasts under the named coding, and the last column gives the share of nonzero contrasts with at most one active source. The frames are nested analysis populations, not independent replications.

Frame	Coding	N	Positive	Negative	Nonzero	$k \leq 1$ (%)
Three-target	Binary	1120	120	56	176	94.3
Three-target	Graded	1120	367	103	470	74.9
Four-target	Binary	1440	178	93	271	95.2
Four-target	Graded	1440	446	164	610	72.6
Retention	Binary	1581	182	103	285	95.4
Retention	Graded	1581	457	184	641	71.0

D.2 Interpreting source-set variation

Main Table 4 reports the aggregate set/count decomposition, while Table 12 resolves source-set variation into all six component pairs. Pairwise differences occur for 138–171 of the same 360 target–task pairs, showing that component choice frequently changes which source models provide nonzero evidence. Because activity-based sets contain both beneficial and harmful source contrasts, membership and cardinality changes describe evidence availability rather than the utility of the newly active source. The rows share sources, targets, and tasks and are therefore descriptive views of one analysis frame.

D.3 How outcome coding changes recurrence

Under graded coding, the retention-frame histogram of active-source counts $k = 0, 1, 2, 3$ is $(131, 324, 162, 24)$ among 641 nonzero target contrasts. Thus 455/641 (71.0%) have at most one

Table 12: Pairwise differences between active source-model sets in the clean four-target frame. Each row compares the two named components across the same 360 target–task pairs. Rates are descriptive because rows share observations.

Components	Different sets	Pairs	Rate (%)
h0 / h1	171	360	47.5
h0 / h2	164	360	45.6
h0 / h3	154	360	42.8
h1 / h2	150	360	41.7
h1 / h3	144	360	40.0
h2 / h3	138	360	38.3

active source. These values complement, rather than reproduce, the binary histograms in main Figure 6 and Table 5. The larger graded source support shows that sparsity depends on outcome resolution as well as source availability; it is not a universal transfer law.

Mixed source signs are necessary, not sufficient, for positive reweighting to reverse a sign. Neither that sign constraint nor the observed recurrence counts bound AUC. No null analysis preserving the marginal zero rates is claimed here, and no empirical transfer ceiling follows from these counts.

E Predictive and Decision Diagnostics

E.1 Alternative populations and outcome coding

Table 13 supplies the existing sensitivity results omitted from the main AUC table. The headline three-target binary row is deliberately excluded: it remains in main Table 6. Each retained row has its own nonzero denominator. Graded coding changes both that selected set and the source scores; its AUC is not a paired estimate of improvement over binary coding. None of these alternatives replaces the main evaluation population or provides independent replication.

Table 13: Population and outcome-coding sensitivity beyond the primary three-target binary evaluation in Table 6. Identity is an ability diagnostic, not a component-selection policy. Each row has its own nonzero denominator; values are point estimates, not significance claims.

Frame	Coding	Nonzero	Weighted	Uniform	Vote	Identity
Three-target	Graded	470	0.6465	0.6392	0.6279	0.6657
Four-target	Binary	271	0.6446	0.6376	0.6430	0.6663
Four-target	Graded	610	0.6344	0.6259	0.6081	0.6978
Retention	Binary	285	0.6493	0.6416	0.6490	0.6699
Retention	Graded	641	0.6117	0.5993	0.5970	0.7060

E.2 Pooled uncertainty and domain heterogeneity

Figure 8 visualizes the fixed-score pooled intervals and the domain-level point estimates reported numerically in the main paper. The pooled source-based predictors are close, whereas the domain panel reveals a telecom reversal: source-based scores fall below chance while the identity diagnostic rises above it. This pattern motivates reporting both aggregation levels. The identity series uses target ability and remains an in-sample diagnostic with different inputs from the source-only predictors.

E.3 AUC sensitivity under exhaustive source reassignment

The main text reports score-sign invariance under source reassignment. The additional diagnostic here concerns ranking: across the existing $6^4 = 1,296$ retention-frame assignments, AUC ranges from 0.6171 to 0.6658, with mean 0.6406. Of these assignments, 395 achieve at least the observed AUC of 0.6493, a descriptive tail fraction of 0.3048. Therefore, score-sign invariance must not be read as AUC invariance. Reassignment preserves the observed source-effect multisets and missing masks; it neither rules out all predictive signal nor establishes a population-level causal result.

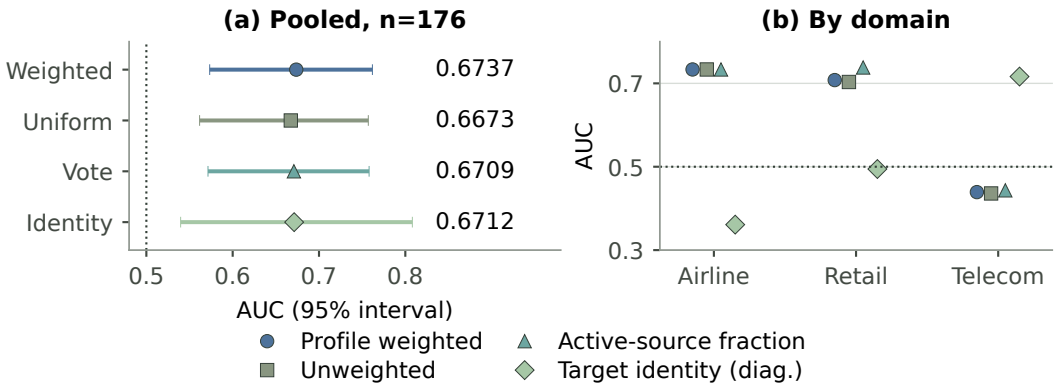

Figure 8: Prediction diagnostics on nonzero contrasts in the clean three-target frame. The left panel shows pooled ROC–AUC with task-clustered fixed-score 95% intervals; the right panel shows domain point estimates. Airline, retail, and telecom contain 28, 85, and 63 evaluated rows. The identity series is an in-sample target-ability diagnostic.

E.4 Screening policy accounting and full cost sensitivity

Figure 9 summarizes the discrete policy-level trade-off between retained utility and blocked positive and negative effects. The same retained counts from main Table 7 determine

$$U_{\text{none}} = 120 - 56\lambda, \qquad U_{\text{active}} = 107 - 45\lambda, \qquad U_{\text{conservative}} = 48 - 5\lambda.$$

Figure 10 then traces these utilities continuously over the harm-to-benefit ratio λ. Active-only exceeds unfiltered utility for $\lambda > 13/11$; conservative exceeds it for $\lambda > 72/51$ and exceeds active-only for $\lambda > 59/40$. These algebraic crossings summarize the observed marginal configuration effects under stated costs. Blocking all has zero utility and remains a distinct policy; prospective joint-harness utility requires a separate deployment evaluation.

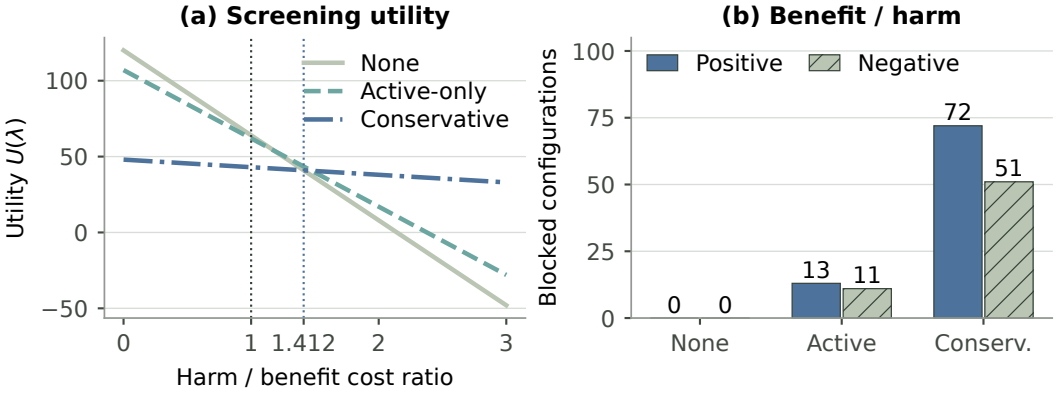

Figure 9: Retrospective screening trade-offs in the clean three-target frame. The left panel compares utility as the harm-to-benefit ratio changes; the right panel shows positive and negative configurations blocked by each policy. Conservative screening blocks 51 negative effects together with 72 positive effects and crosses unfiltered utility at $72/51 \approx 1.412$.

F Limitations and Concise Audit History

F.1 Unresolved limits of the existing evidence

The experiment pool is small and heterogeneous; recorded simulator and step-budget differences limit causal interpretation. Trials are few and occasionally incomplete. Raw coverage is not complete, and unresolved run-to-summary discrepancies may change estimates. Profiles encode baseline ability and are not independently re-estimated in this audit. Eligibility filters, nonzero-effect conditioning, source count, and coding resolution all change the analysis target. None of the reported intervals covers these systematic uncertainties. Retrospective screening is not prospective safety validation, and marginal effects need not compose across components.

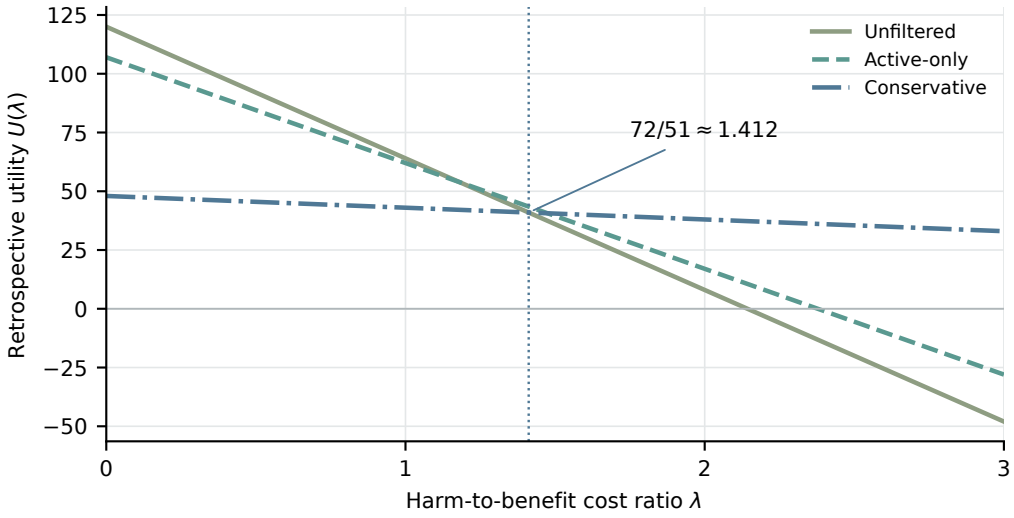

Figure 10: Continuous cost sensitivity computed from the same observed configuration contrasts. The marked conservative/unfiltered crossing is $72/51 \approx 1.412$; the preferred policy changes with the assigned harm-to-benefit ratio.

F.2 What was removed from the PDF, and why

The original appendix and source snapshot remain in the audit history. Figures and count tables already present in the main text are not repeated. The supplement retains within-target ranking, graded and population sensitivity, source-reassignment ranking diagnostics, and full cost curves. Historical figures with incompatible values are not silently restored. Failed model-expansion logs, old probe-size and clustering sweeps, detailed script-defect history, idealized ceiling calculations, and external few-shot examples are archived rather than treated as current primary evidence. Their populations, representations, or assumptions are not aligned with the present release. The previously withdrawn source-selection comparison and its interval variants remain withdrawn; this revision does not resurrect them with a favorable subset or select intervals after inspecting outcomes. Removing them is a scope correction, not evidence that their limitations have been resolved.

F.3 Version discipline for further editing

Every numerical claim must specify its evidence layer, population, outcome coding, and score rule. The version registry links summary results to the separate raw-reconciliation ledger. Historical 281-row binary and 640-row graded counts are not current results. Unresolved values remain explicit; future run selection must be justified by provenance and intended configuration rather than agreement with outcomes.