A POOL-LEVEL AUDIT OF SELECTED REWARD-MODEL CHECKPOINTS FOR TOOL-TRAJECTORY SELECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Reward models often turn a candidate pool into one selected trajectory. The maximum score by itself does not show whether that trajectory is useful. We audit selected checkpoints on executable tool trajectories. The main evidence comes from 37 mixed-outcome τ^2-bench airline pools, with a secondary closure on AgentSuite airline and held-out retail cells. Five independently started fixed-process runs gave identical results on the original 370 candidates. By contrast, three semantically equivalent serializations agreed on the winner in only $.568$ of pools (95% CI $[.405, .730]$). At the primary joint-evaluator seed, mixed-only excess utility was $-.284$, $-.129$, and $-.100$ in the three cells and varied across seeds. On the same 69 completed mixed pools, the follow-up sensitivity checks found $20/69$ full-cycle winner agreements after reversal of the primary orientation and $62/69$ agreements between padded and dynamic full cycles. A secondary matched-slot arm covered 69/102 gates and provides a feasibility-conditional, binary-outcome, policy-plus-graph diagnostic. Our conclusions are limited to the named checkpoints, policies, pools, serializations, graph constructions, and context protocols; they do not rank selector classes or establish general equivalence.

1 INTRODUCTION

At inference time, reward models may select a candidate trajectory or supply an alignment reward (Zhu & Lu, 2026; Gao et al., 2023). Selection depends on the ordering within a pool. When several candidates share the maximum score, the score defines a top set without choosing one of its members; the tie policy determines which trajectory is returned.

Agentic reward modeling includes process supervision, preference-verification mixtures, and tool-specialized outcome models (Peng et al., 2025; Choudhury, 2025; Zhang et al., 2025; Zhao et al., 2026a; Agarwal et al., 2025). Related benchmarks measure planning and step quality in tool-using agents (Wang et al., 2026b; Fan et al., 2026b; Li et al., 2026), while RMEval connects reward-model scores to downstream policy-checkpoint rankings (Qiu et al., 2026). A concurrent tau2-bench study evaluates whether judges detect false success when verified state changes disagree with completion claims (Advani, 2026). Most of this work evaluates score accuracy, ranking consistency, step quality, or judge detection. We study the executable outcome of the single trajectory returned by a named pool policy. The constructed grid tests general conversational preference checkpoints zero-shot on serialized tool trajectories. Tool-trained and locally fine-tuned evaluators enter only in later diagnostic and closure arms, so the grid does not characterize agent reward models as a class.

We examine named pointwise checkpoints and a task-disjoint joint pointwise/pairwise diagnostic as selectors of executable tool trajectories. The primary natural analysis contains 37 ten-candidate, mixed-outcome airline pools; the secondary closure uses AgentSuite airline and held-out retail pools. The analysis keeps score-defined identification separate from executable outcomes, process repeatability separate from serialization sensitivity, and mixed-only estimates separate from feasible-only and all-task estimates. The task or pool remains the independent unit throughout.

We report:

- a policy-explicit audit of named checkpoints with pool-level uncertainty and separate mixed-only, feasible-only, and all-task denominators;
- an executable-outcome and serialization analysis showing that singleton scores, rank summaries, and deterministic repeats can miss failures exposed by semantically equivalent multi-turn serializations;

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.

- a resource-conditioned closure using three evaluator seeds, five process repeats, and three serializations, together with a released CPU audit of feasibility, provenance, and pairwise support.

Tie-aware best-of-N analysis predates this study. The four estimands below organize reporting and introduce no new algebra. Landesberg (Landesberg, 2026) establishes the decision-validity gap for single-turn best-of-N judging, and RMEval (Qiu et al., 2026) studies ranking consistency across policy checkpoints. Our evidence concerns multi-turn tool-trajectory serialization, executable reference outcomes, and resource-conditioned feasibility. We limit all claims to the measured checkpoints, policies, pool sizes, source–domain cells, and observation protocols.

2 Related Work

Reward-model evaluation. RM-Bench and reWordBench directly test reliability under controlled style, format, or meaning-preserving changes (Liu et al., 2025; Wu et al., 2025). RewardBench, RMB, and PRMBench emphasize benchmark accuracy or process-level labels under their respective task protocols (Lambert et al., 2025; Zhou et al., 2025; Song et al., 2025). RMEval links reward-model scores to downstream policy-checkpoint ranking consistency (Qiu et al., 2026). Score accuracy and rank association across these benchmarks do not by themselves determine which member of a candidate pool is selected. We therefore study the downstream within-pool choice.

Agents, tools, and trajectories. Plan-RewardBench, AgentProcessBench, and ToolPRMBench evaluate planning or step quality in tool-using agents (Wang et al., 2026b; Fan et al., 2026b; Li et al., 2026). Other work trains reward models specifically for agents (Xia et al., 2025; Men et al., 2025; Fan et al., 2026a). A concurrent tau2-bench study shows that LLM judges can miss false success when verified state changes disagree with completion language (Advani, 2026); it does not evaluate candidate-pool selection. ToolSandbox and Hack-Verifiable Environments provide public tool-use settings with executable checks (Lu et al., 2024; Roth et al., 2026).

Decision validity in best-of-N selection. Landesberg (2026) provide the closest prior study. Across 5,000 four-response Chatbot Arena pools, coarse pointwise scores can tie and recover the best response poorly despite adequate global correlation. Their matched best-of-2 audit evaluates an explicit pairwise alternative. We apply the same decision-validity question to multi-turn tool trajectories, for which serialization, executable reference outcomes, and resource gates form part of the observation protocol. The result is an empirical transfer of the audit; it neither proposes a selector theory nor characterizes the agentic-RM class.

Selection and overoptimization. Optimizing a proxy can separate it from its target (Gao et al., 2023; Rafailov et al., 2024; Khalaf et al., 2025). Reward-evaluation and reward-hacking benchmarks examine related selection dynamics (Kim et al., 2025; Wang et al., 2026c; Thaman, 2026). Ensembling and uncertainty-aware methods offer possible mitigations (Coste et al., 2024; Srivastava et al., 2025; Yang et al., 2026). We keep every selector fixed and perform no policy update.

3 Protocol and Methods

3.1 Study design

The primary natural analysis uses the public τ^2-bench Airline With Code Agents release (Snorkel AI, 2026), which contains 50 ten-trajectory tasks from five generators. Thirty-seven tasks contain both outcomes and form the mixed-only discrimination sample (370 candidates; 185 successes). The 13 saturated tasks enter only the descriptive all-task analysis (Appendix H). The closure uses a retained local snapshot labelled `AgentSuite/tau-bench-trajectories`, dated 2026-09-18, with 4,950 rows, 165 tasks (50 airline; 115 retail), and 30 generator files. We froze ten generators before outcome analysis; the released task–generator crosswalk records all 1,650 selected cells. The primary label is `eval_result.score`. Two database-match subfields agree with it, but they are not separate outcomes. A retained metadata record has SHA-256 `10a49503...3d55d`; this authenticates that record, not the raw snapshot. The raw-snapshot hash, upstream accession, and licence record are unavailable. AgentSuite is therefore an internal closure source with insufficient provenance for a fully reproducible public benchmark (Appendix D). The three cells cover airline and retail only.

We use four pointwise checkpoints (Table 1). The full-context natural analysis uses D-Scalar, G-Rubric, and T-Verifier. D-Vector is excluded because its native 4,096-token cap cannot support the paired full-context protocol. Each family name refers to one checkpoint, not a model class.

Table 1: Selected pointwise checkpoints. "131k YaRN diagnostic" denotes the recorded extrapolated full-context protocol; D-Vector retained its native cap. Appendix T gives prompts, parsers, training records, and checkpoint version details.

Family	Checkpoint and read-out	Context in study	Training target / tool exposure
D-Scalar	Skywork-Reward-V2-Qwen3-0.6B; sigmoid scalar (Liu et al., 2026)	131k YaRN diagnostic	zero-shot stress test; preference target
D-Vector	ArmoRM-Llama3-8B-v0.1; scalar head (Wang et al., 2024)	4,096 native	zero-shot stress test; multi-objective preference
G-Rubric	R3-Qwen3-4B-14k; integer 1 to 5 (Anugraha et al., 2025)	131k YaRN diagnostic	zero-shot stress test; response rubric
T-Verifier	Qwen3-8B LoRA; success probability	131k YaRN diagnostic; trained at 4,096	local binary-outcome diagnostic

The full-context extension is an extrapolation beyond the native checkpoint. The runner loaded a 40,960-position Qwen3 configuration, set `max_position_embeddings` $= 131{,}072$, applied YaRN RoPE scaling with factor 3.2 and `original_max_position_embeddings` $= 40{,}960$, disabled the KV cache, and chunked the Qwen3 MLP pass. A separate controlled-context arm used the same native 40,960-position configuration without RoPE scaling and nested visible budgets of 4,096, 8,192, and 16,384 tokens. We call the 131k results context-extension diagnostics; they are not general properties of the named checkpoints.

T-Verifier was trained with seed 20260914 for three fixed epochs on 140 examples: 90 from nine disjoint airline tasks and 50 positive APIGen conversations. Its validation set contains 40 examples from four further tasks. Validation accuracies at epochs 1 to 3 were .550, .825, and .800; losses were .8946, .5767, and .5945. The training loop has no early-stopping or checkpoint-selection branch and saves only the terminal epoch. The 37 evaluation task IDs are disjoint from the training and validation tasks. We found no record of evaluation outcomes entering the save rule. The available records are insufficient to establish investigator blinding. We therefore use T-Verifier only as a local diagnostic.

The closure trained a Qwen3-4B-Instruct-2507 two-class LoRA diagnostic with shared pointwise-success and pairwise binary-outcome heads. Every training, validation, and calibration pair was a within-task success–failure pair; same-outcome pairs were absent. Across seeds 20260920/21/22, pair counts were 232/234/234 for training, 78/68/62 for validation, and 66/66/66 for calibration. The fixed retail split contained 70/15/15/15 train/validation/calibration/test tasks; AgentSuite airline and original airline were unseen. Inputs had an 8,192-token limit without truncation. The checkpoint minimized joint validation negative log likelihood (three epochs maximum; patience one), then fit pointwise and pairwise temperatures on calibration tasks. Test and airline outcomes were not used for selection. Seed 20260920 was specified as primary; the other seeds are replications. Test cycles can contain same-outcome edges, so those outputs lie outside supervised pair support and are interpreted only as binary-outcome contrasts (Appendix D.1).

The full-context scorers processed candidates one at a time, without pool position or neighboring candidates. The largest complete joint-evaluator prompt contained 125,301 tokens, below the 262,144-token context limit; the earlier pointwise protocol used the recorded 131,072-token target. All 1,110 original family–candidate rows and 1,020 joint-evaluator rows per seed parsed without error or truncation. We joined oracle labels after scoring.

For the 370 original candidates, we invoked the primary joint evaluator in five separately started processes with fixed CUDA settings, disabled dropout, and deterministic decoding. Three semantic serializations kept the same event objects and changed only JSON key order, whitespace, and indentation. We report process repeatability and serialization sensitivity separately. A separately invoked validator compared the formal analysis with a fresh-directory reproduction before we promoted any closure result.

Fresh constructed pools provide a known-answer control. Each pool contains trajectories with executable success labels and preserves the tested public form. The larger constructed grid, context

Table 2: Study map and evidence hierarchy. The independent unit is always a task/pool; repeated scores, candidates, and serializations are nested.

Block	Status	Data and reuse	Primary question
Constructed N64	confirmatory	64 fresh pools; no candidate-level pooling	Can identification track an executable outcome?
Natural transfer	primary, mixed-only	37 fresh mixed airline pools; three checkpoints	What utility does a named policy attain?
GPU closure	secondary closure	3 cells, 3 seeds; same pools across seeds	How do seed, source, and domain change the diagnostic?
Serialization / repeats	secondary	370 candidates; 5 repeats and 3 formats	Is fixed-process repeatability format-invariant?
Matched slots / context	secondary diagnostic, feasible-only	69/102 gates; one sparse graph	What is observed under the recorded implementation?
Graph / length sensitivity	audit-triggered diagnostic	same 69 feasible mixed pools; no new task samples	How sensitive is the sparse pairwise implementation to graph, orientation, padding, and length?

Table 3: Selector estimands. "Identification" is a score-defined unique maximum, not correctness under the executable reference outcome. The other outcome-based quantities require reference labels and, for utility and regret, a named policy.

Quantity	Pool-level estimand	Question
Score-defined singleton	$I_i = \mathbf{1}\{\lvert A_i\rvert = 1\}$	Did the score surface identify one candidate?
Oracle-free resolution receipt	$\lvert A_i^{\epsilon}\rvert,\ \mathrm{singleton}_d,\ \Delta_{1,2},\ h_{\mathrm{ser}}$	Does the deployed score/serialization expose a stable policy?
Top-set quality	$Q_i = \mathrm{mean}_{c\in A_i} Y_i(c)$	How good is the whole top set?
Policy utility	$U_i(\pi) = \mathbb{E}_{c\sim\pi_i} Y_i(c)$	What does the deployed tie policy attain?
Regret	$R_i(\pi) = \max_{c\in C_i} Y_i(c) - U_i(\pi)$	What attainable reference-outcome utility is lost?

projections, earlier two-pool retail split, pairwise ToolRM endpoint, and factorial analyses are secondary or exploratory; they appear in the appendix. Tables 9–10 record when each analysis was specified and which data were already visible.

3.2 Estimands

For pool i, let C_i contain its candidates and $S_i(c)$ denote a checkpoint score. The top equivalence class is $A_i = \arg\max_{c\in C_i} S_i(c)$. When an executable reference outcome is available, $Y_i(c)$ denotes its label and π_i a named tie policy supported on A_i. Table 3 separates the four selector quantities used throughout.

For binary Y_i, random-within-tie utility equals top-set quality. Every mixed pool contains a success, so regret is $1 - U_i(\pi)$; excess utility subtracts the within-pool success share. These are bookkeeping identities, not additional evidence. We keep three populations separate: primary natural-transfer estimates $\mathbb{E}[U \mid \text{mixed}]$ use 37 original-airline tasks; matched-slot levels estimate $\mathbb{E}[U \mid \text{mixed}, F{=}1]$ on the 69 feasible gates; and the appendix reports descriptive $\mathbb{E}[U]$ over all 50 original-airline tasks. Feasibility coverage $\Pr(F{=}1)$ is a systems outcome. We did not rescore the 21 memory/OOM exclusions with a fallback, so policy-plus-fallback utility is unobserved and is not imputed or folded into coverage.

We evaluate four natural-transfer policies. Manifest order selects the first exact maximum in the retained score-file order. The pool builder preserves source-parquet row order within each task. Candidate position is associated with generator (Cramér's $V = .633$), dataset version ($V = .549$), and their joint label ($V = .587$), so manifest-order results may reflect provenance. Random-within-tie samples uniformly from the exact maximum set. Stable-ID selects the lexicographically smallest complete candidate ID in the top set; the rule is outcome-blind, although candidate IDs encode provenance. Abstain-on-tie returns no candidate when the exact maximum is shared and reports utility only for covered pools. The constructed-grid selection curves use a separate three-policy audit:

fixed permutation order (seed 20260912), ascending ID, and descending ID. In the joint-evaluator closure, the primary and matched-slot rule is the lexicographically smallest complete candidate ID among exact maxima. Random-within-tie is the provenance-neutral sensitivity and is numerically identical in all headline cells because exact ties do not occur there.

The post-result reporting target is D-Scalar under manifest order for mean excess utility. Its 37 exact maxima are singleton, so the four policies agree under exact-equality arithmetic for this checkpoint. That agreement need not survive epsilon or rounding coarsening. We fixed this hierarchy after seeing the original full-context results; it is not a preregistration. All other evaluator-policy pairs and score-resolution analyses are sensitivity results.

To test score resolution, we define

$$g_{ij} = \frac{\max_k S_i(k) - S_i(j)}{\max\{P_{90}(S_i) - P_{10}(S_i), 10^{-8}\}}, \tag{1}$$

and include candidate j in the epsilon top set when $g_{ij} \leq \epsilon$. The fixed grid is $\epsilon \in \{0, .01, .02, .05, .10\}$. We also round D-Scalar and T-Verifier probabilities to 6, 4, 3, 2, 1, and 0 decimals. G-Rubric retains its native integer scale without jitter. Top-1 versus top-2 margins are reported on raw, native, and normalized scales.

3.3 Inference and audit

Policy and resolution intervals use 10,000 shared pool-bootstrap draws with NumPy PCG64 seed 20260920. A candidate-order control applies 1,000 shared within-pool permutations using the same seed while holding scores and labels fixed. We require agreement between the formal analysis and a fresh-directory reproduction after attempt IDs are normalized. An independent implementation reconstructs every policy, permutation, epsilon, rounding, margin, and power quantity from the raw rows.

Power calculations use a two-sided one-sample t planning model with $\alpha = .05$, 80% power, and a smallest effect size of interest (SESOI) of $|\text{mean excess utility}| = .10$. We report the minimum detectable effect (MDE) at the observed denominator and the sample size required for the SESOI, both unadjusted and with a three-family Bonferroni correction. Practical equivalence requires the complete 95% t interval to lie within $[-.10, +.10]$.

Matched input-token-slot policy comparison. This secondary arm rescored pointwise argmax, pairwise Bradley–Terry aggregation (Bradley & Terry, 1952), and pointwise top-5 followed by pairwise reranking with the primary diagnostic evaluator. We use Bradley–Terry as a graph aggregation rule; the sparse, directed cycles and shared classifier output are not treated as evidence that its latent-score or independence assumptions hold. For each pool, M was the larger of the longest single input and half the longest of the 45 pair inputs. Single and pair calls were right-padded to $M/2M$ with the adapter tokenizer's pad ID (EOS only when no pad token was available), and padded positions had zero attention-mask entries. We passed no explicit position IDs. Pointwise scoring made two deterministic calls per candidate and averaged them; the 20 exposures match the planned count but are not independent observations. The pairwise arm used one seeded 10-edge cycle; top-5 used ten pointwise screens and a five-edge cycle. Cycles sort IDs and then shuffle with seed $20260920 + 1009 \times \text{pool_index}$ (top-5 adds 100000); orientations alternate. Each arm scheduled $20M$ token slots, while wall time, FLOPs, and utilization were left unmatched. We fixed eligibility before joining labels: 69 mixed pools completed, 21 were excluded atomically for memory/OOM, and 12 saturated pools were excluded from discrimination analysis. No input was truncated. The 8,192-token training support does not validate test prompts up to 125,301 tokens. With one graph and one orientation per edge, the arm is a binary-outcome and policy-plus-graph implementation diagnostic. It does not compare selector classes or establish graph invariance. Appendix F reports timing, memory, coverage, and implementation details.

Audit-triggered graph and length sensitivity. After the initial closure, we froze the same 69 mixed feasible pools before loading the model. The graph arm used seeds $20260920, 20260921, 20260922$, both orientations, and both the full ten-edge and inherited top-five five-edge cycles (6,210 manifest rows and 3,856 unique padded directed calls). The length arm used the primary forward graph edges (1,035 manifest rows), reused 996 matching padded scores with provenance, and ran 996 new dynamic/unpadded calls. Every exclusion remains in the denominator, with no fallback or post-result pool selection. Both arms are sensitivity diagnostics on the same pools and add no independent samples.

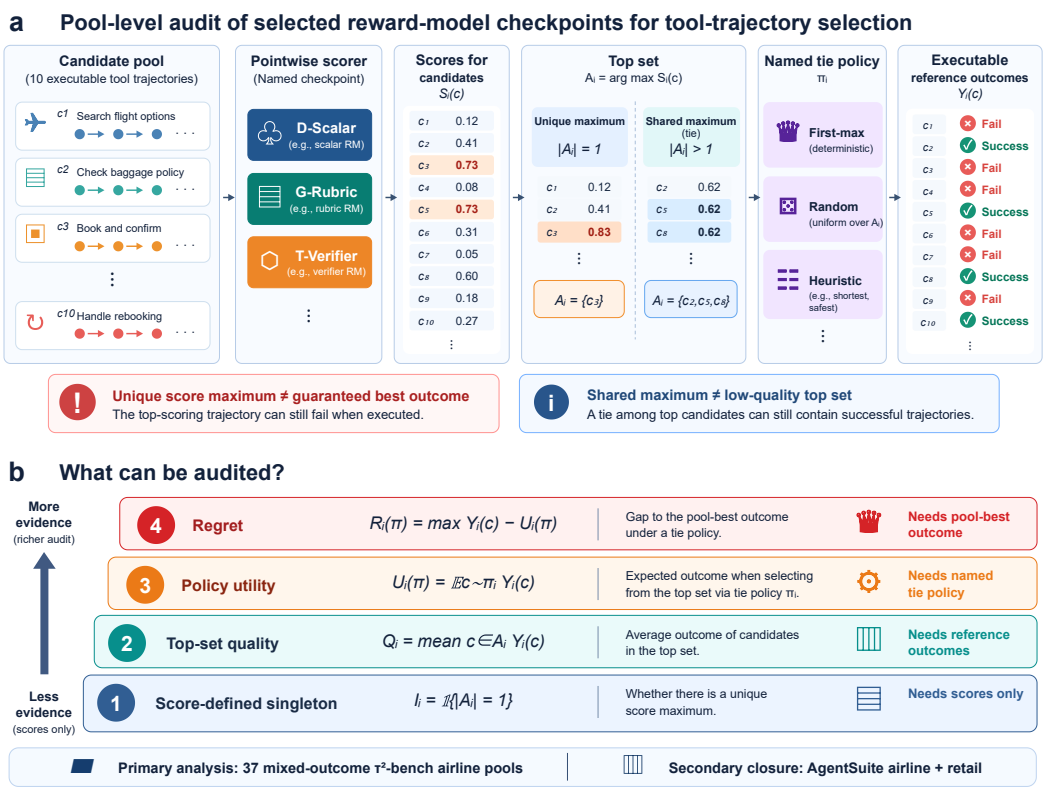

Figure 1: Score-defined selection and outcome-based audit. **a**, pointwise scoring preserves candidate identity, defines an unordered top set, and leaves a named policy to return one member. A unique score maximum does not establish oracle validity, and a shared maximum does not imply low top-set quality. Study-specific reference outcomes are revealed only after scoring, allowing separate audits of the top set and the returned candidate. **b**, the four selector estimands use different evidence: scores identify a unique winner, reference outcomes assess top-set quality, a named policy determines utility, and the pool-best outcome defines regret. Schematic.

4 Results

4.1 Known-answer controls (confirmatory)

The fresh controls compared score-defined identification with the executable outcome. D-Scalar produced a singleton maximum in every pool but selected a solved trajectory in only $15/32$ pools. Its excess utility was $-.031$ (95% CI $[-.188, .156]$). G-Rubric produced shared maxima whose top sets were .937 solved on average. Ascending-ID, descending-ID, seeded-permutation, and random-within-tie policies yielded utilities of 1.000, .438, .969, and .937. A single score surface can therefore yield different utilities when its top set contains multiple candidates (Appendix J).

4.2 Named-policy natural transfer (primary, mixed-only)

Table 4 shows the dependence on the tie policy. G-Rubric's manifest-order utility exceeds its random-within-tie value by .102. Stable-ID ordering gives .324; candidate IDs encode provenance, so this is a deterministic sensitivity, not a neutral deployment estimate. Abstention avoids forced choices in tied pools but covers only the six singleton pools, all of which select failures.

Manifest order is the source-parquet row order retained by the pool builder, and the pointwise evaluators never see it. Across 1,000 within-pool reorderings, G-Rubric's mean utility was .414; the 2.5 to 97.5 percentile range was $[.324, .514]$, and the full range was $[.270, .595]$. Only .040 of draws matched or exceeded the observed manifest utility. This policy-order diagnostic measures a structured tie rule. It does not test evaluator input-order sensitivity and is not a randomization p-value. The near-upper-tail value is consistent with alignment between row order and outcomes.

Table 4: Primary mixed-only named-policy utility on 37 airline pools. Intervals are 95% pool-bootstrap intervals. Exact singleton is score-defined; utility uses the executable reference outcome. Redundant binary-pool columns are retained for audit. G-Rubric abstention covers 6/37 pools, so its utility is conditional on coverage.

Family	Policy	Singleton	Coverage	Utility	Excess	Regret
D-Scalar	all four named	1.000	1.000	.486 [.324, .649]	−.014 [−.130, .108]	.514 [.351, .676]
G-Rubric	manifest-order	.162	1.000	.514 [.351, .676]	+.014 [−.181, .203]	.486 [.324, .649]
G-Rubric	random-within-tie	.162	1.000	.412 [.292, .535]	−.088 [−.243, .058]	.588 [.465, .708]
G-Rubric	stable-ID	.162	1.000	.324 [.189, .486]	−.176 [−.346, −.005]	.676 [.514, .811]
G-Rubric	abstain-on-tie	.162	.162 [.054, .297]	.000 [.000, .000]	−.717 [−.900, −.429]	1.000 [1.000, 1.000]
T-Verifier	all four named	1.000	1.000	.514 [.351, .676]	+.014 [−.135, .159]	.486 [.324, .649]

D-Scalar and T-Verifier had positive pool-level Spearman estimates, .122 (95% CI $[.015, .232]$) and .167 ($[.052, .281]$). Their AUC estimates were .593 ($[.512, .674]$) and .628 ($[.538, .717]$). Neither checkpoint showed detectable manifest-order excess utility. G-Rubric's rank intervals contained zero for Spearman and .5 for AUC. Rank association did not resolve top-1 utility.

The natural pools also have a scaffold imbalance. Original-dialogue candidates succeeded in $115/185$ cases (.622), while code-generation candidates succeeded in $70/185$ (.378). D-Scalar, G-Rubric, and T-Verifier selected the lower-base-rate code-generation scaffold in 26, 30, and 22 of the 37 pools, respectively. A CPU audit found a position–outcome association of Cramér's $V = .278$ with within-pool permutation $p = 4.99975 \times 10^{-5}$, alongside position–generator/version $V = .587$. Manifest-order and stable-ID estimates therefore depend on provenance; we report them as deterministic sensitivities, not as causal scaffold effects.

4.3 Score resolution and power (secondary audit)

Exact arithmetic produced singleton maxima in every D-Scalar and T-Verifier pool. Their policy agreement therefore reflects exact equality arithmetic and need not persist under coarsening. At $\epsilon = .10$, mean top-set size increased to 1.70 and 2.78, while near-singleton rates fell to .568 and .297. G-Rubric remained at a mean top-set size of 3.68 across the epsilon grid because its native score is integer-valued. Rounding probabilities to two decimals reduced singleton rates to .892 for D-Scalar and .649 for T-Verifier; one-decimal rates were .514 and .324. Median normalized top-1 versus top-2 margins were .107, 0, and .032 for D-Scalar, G-Rubric, and T-Verifier. A deployment receipt should expose the tie policy or abstain when the epsilon set is non-singleton, the rounded score is coarse, or the margin is near zero. Appendix G reports the fixed grids, and the GPU closure reports five process repeats and three serialization variants.

For the post-result D-Scalar reporting target, mean excess utility was $-.0135$ and the 95% t interval was $[-.138, .111]$. The 80%-power MDE at $n = 37$ was .1768. Under the planning model, detecting the .10 SESOI requires 112 pools, or 149 after a three-family Bonferroni correction. Corresponding manifest-order MDEs were .2869 for G-Rubric and .2178 for T-Verifier; unadjusted required sample sizes were 291 and 169. None of the 12 evaluator-policy 95% t intervals lay inside $[-.10, +.10]$, so the non-detections do not establish practical equivalence.

4.4 GPU closure: seeds, cells and stability (secondary)

Seed 20260920 was evaluated separately in the three source–domain cells before pooling. Table 5 reports its ordinary cell runs; saturated AgentSuite pools are excluded from ranking inference. The primary seed gave negative excess utility in every cell: the original-airline interval $[-.449, -.108]$ is below the $-.10$ SESOI, but its three-seed span $[-.284, +.014]$ crosses zero. Across seeds, the AgentSuite-airline span was $[-.182, -.129]$ and the retail span $[-.100, +.167]$ (replication-only spans $[-.068, +.014]$, $[-.182, -.182]$, and $[+.033, +.167]$). These are seed diagnostics, not additional pools.

Five fixed-process repeats were identical (winner agreement, ICCs, drift, and rank displacement all $1.000/0$). Three semantic serializations had all-template agreement .568 (95% CI $[.405, .730]$) and mean pairwise agreement .703 (95% CI $[.586, .811]$). Under stable-ID, original-airline success@1/excess/regret were $(.216, -.284, .784)$ for sorted-compact and insertion-compact and $(.297, -.203, .703)$ for sorted-pretty; outcome disagreement was .126 (95% CI $[.054, .216]$). The receipt freezes a version label, file name/size, and schema record and requires an alternative-serialization recheck. Agreement 1.0 and median normalized margin .01 remain unvalidated hy-

Table 5: Primary joint-evaluator results by source–domain cell. Estimates use the ordinary (unpadded) cell run from seed 20260920 and 10,000 pool-bootstrap draws; the independent unit is the mixed task/pool. The tie rule is the lexicographically smallest complete candidate ID among exact maxima, with no exact ties in these cells ($0/37$, $0/38$, $0/15$). Table 20 is a distinct fresh padded rescoring estimand, which is why its retail pointwise row is not numerically identical. The three cells span two semantic domains.

Cell	Mixed pools	Success@1	95% CI	Excess utility	95% CI	AUC
Original airline	37	.216	[.081, .351]	−.284	[−.449, −.108]	.435
AgentSuite airline	38	.342	[.184, .500]	−.129	[−.271, .016]	.506
AgentSuite retail	15	.533	[.267, .800]	−.100	[−.267, .053]	.696

potheses, not promotion gates. Any thresholds must be fixed before the final test set and evaluated on independent tasks for coverage, failure detection, executable-outcome utility, abstention/fallback rate and utility, and latency.

4.5 Matched slots (secondary feasible-only diagnostic)

Fresh padded inputs produced the conditional levels in Table 20 and the 27 contrasts in Table 21; these are separate from the ordinary scores in Table 5. Directions varied by cell. All Holm-adjusted p-values were 1.000, no interval met the $\pm.10$ equivalence criterion, and MDEs were .171–.356. This arm does not rank policies or establish equivalence.

4.6 Graph/orientation, padding/length, and support sensitivity (audit-triggered)

The formal sensitivity package completed all $6 \times 69 = 414$ graph variants for each cycle construction (828 pool-level rows), with 69/69 coverage for every variant and no failed or excluded score rows. The primary forward graph (seed 20260920) reproduced the primary winners for all 69 full cycles and all 69 top-five cycles. Reversing the same edges reduced agreement to $20/69 = .290$ for full cycles and $24/69 = .348$ for top-five cycles. Forward replications with seeds 20260921 and 20260922 agreed with the primary full-cycle winners in $21/69 = .304$ and $23/69 = .333$ pools, respectively (top-five: $36/69 = .522$ and $39/69 = .565$). Across the three seeds, forward full-cycle excess utility ranged from $-.178$ to $-.164$ and reversed-orientation full-cycle excess was $-.207$ in every seed; top-five excess ranged from $-.217$ to $-.145$. Mean within-pool Spearman rank correlation against the primary graph ranged from .602 to .719 for non-primary full cycles and from .426 to .712 for non-primary top-five cycles. No fallback was invoked. Across the 3,856 completed score receipts, median wall time was 3.33 seconds and median/max peak allocated memory was $10.22/24.63$ GiB. The policy comparison is therefore conditional on the graph and its orientation; it is not evidence about a selector class.

The paired padding arm completed all 1,035 edge pairs on all 69 pools. Mean absolute probability drift was .0121 for $\leq$ 8k tokens ($n = 365$), .0182 for 8–16k ($n = 536$), .0199 for 16–32k ($n = 125$), and .0143 for $>$ 32k ($n = 9$). Padded and dynamic tensors returned the same full-cycle excess utility to three decimals, while their winners agreed on $62/69 = .899$ full cycles and $67/69 = .971$ top-five cycles. Aggregate utility was stable in this paired implementation check, but individual winner identities changed; the $>$ 32k stratum is too small for a broad length claim.

Pair-support audit. Training, validation, and calibration used only success–failure pairs, whereas 63.2% of the original-airline test-cycle edges were same-outcome. The frozen support diagnostic therefore used all 1,167 opposite-outcome pairs across the 69 feasible pools. It changed 43/69 winners relative to the primary sparse cycles, but success@1 changed only from .377 to .406 (paired difference $+.029$, 95% CI $[-.058, .101]$); excess utility changed from $-.164$ to $-.135$. Because this graph is label-conditioned, denser, and candidate-ID-oriented, it identifies neither a pure support effect nor a deployable policy (Appendix D.1).

Coverage and fallback. Mixed-pool feasibility was 16/37, 38/38, and 15/15 for original airline, AgentSuite airline, and AgentSuite retail; including 12 saturated airline gates, total completion was 69/102. The original-airline included/excluded success shares were $.619/.410$ (difference .209, 95% CI $[.023, .385]$), so its levels are feasible-subset estimates. Excluded pools were not rescored with any fallback, and no combined fallback utility is reported. Coverage, 21 memory/OOM exclusions, 12 saturated exclusions, and conditional utility remain separate quantities.

The all-task denominator, context projections, retail ceiling case, pairwise ToolRM endpoint, and factorial controls appear in Appendix H. Tables 9–10 mark them as descriptive, secondary, or exploratory.

5 Discussion and conclusion

Across the tested checkpoint–pool pairs, exact maxima and positive rank association failed to determine policy utility. Small changes to epsilon or rounding exposed near ties, and semantically equivalent event-key order, whitespace, or indentation changed winners despite deterministic process repeats. This extends earlier decision-validity evidence to multi-turn serialization and executable reference outcomes within the settings we measured. The four estimands organize the report without proposing a new theory or a general agentic-RM benchmark.

A deployment receipt should name the score-to-choice policy and record epsilon-set size, rounded singleton rate, margins, serialization version/schema, coverage, and fallback behavior. Agreement 1.0 and margin $.01$ remain prospective hypotheses. Any gate would need an independent evaluation with thresholds fixed in advance and measurements of coverage, failure detection, utility, abstention/fallback, and latency. Until that evaluation exists, a failed receipt calls for an explicit tie or fallback decision and provides no validity certificate.

The three headline populations answer different questions: mixed-only natural transfer, feasible-only matched-slot levels, and descriptive all-task utility. At the primary seed, success@1/excess utility was $(.216, -.284)$, $(.342, -.129)$, and $(.533, -.100)$ across the three cells, with wider spans across seeds. The sensitivity package shows that the pairwise arm depends on graph seed, orientation, and support definition: support-only scoring changed 43/69 winners without establishing an aggregate utility difference, while dynamic and padded scoring preserved aggregate utility but changed some winner identities. We did not evaluate an all-pairs reference or a mature agent-specific ORM/PRM baseline, so the study does not rank selector classes or establish general length invariance.

6 Limitations

The evidence covers selected checkpoints, two related transactional domains, and small conditional samples. AgentSuite has no retained raw-snapshot hash, accession, or licence record; the joint evaluator is a local diagnostic, not a mature agent-specific baseline. The primary $n = 37$ MDE is .177; matched-slot coverage is 69/102 and is outcome-associated; no fallback utility was observed. Post-result analyses remain secondary or exploratory. Seeds, candidates, repeats, and serializations are nested measurements, not new task samples. The completed support arm is outcome-conditioned and changes orientation as well as edge support; it does not replace an all-pairs reference or a mature agent-specific ORM/PRM baseline. The $>$ 32k length stratum has only nine edges. Appendix E gives the full boundaries.

Reproducibility and Data Availability

The anonymous supplement permits internal reconstruction and numerical re-execution but not independent end-to-end model reproduction. The earlier audit reproduced 7,136 values with maximum difference 0; the support audit verified 1,167/1,167 score rows, and 11/11 fallacy checks passed. We omit weights, adapters, logits, raw AgentSuite data, and credentials. The CPU-only release adds edge-support records, the task–generator crosswalk, hashes of retained inputs, and a minimal rerun script. The separate sensitivity package `ars-gpu-sensitivity-20260924` releases the frozen 69-pool contract, graph/length/support manifests, completed score receipts, paired summaries, and independent audits. It contains no external ORM/PRM evidence. Author contributions and funding are withheld during double-blind review.

AI Use Statement

Large language models were used in two ways: to help write the experiment and analysis scripts, and to polish the language of this manuscript. The research question, study design, estimands, choice of baselines, preregistration, and the interpretation of results were all made by the human authors. Every quantitative claim derives from executed experiment artifacts and can be recomputed from the released validation scripts; the authors are responsible for the accuracy and originality of the final content.

REFERENCES

Laksh Advani. From confident closing to silent failure: Characterizing false success in llm agents. *arXiv preprint arXiv:2606.09863*, 2026. URL https://arxiv.org/abs/2606.09863.

Mayank Agarwal, Ibrahim Abdelaziz, Kinjal Basu, Merve Unuvar, Luis A. Lastras, Yara Rizk, and Pavan Kapanipathi. Toolrm: Outcome reward models for tool-calling large language models. *arXiv preprint arXiv:2509.11963*, 2025. URL https://arxiv.org/abs/2509.11963.

David Anugraha, Zilu Tang, Lester James V. Miranda, Hanyang Zhao, Mohammad Rifqi Farhansyah, Garry Kuwanto, Derry Wijaya, and Genta Indra Winata. R3: Robust Rubric-Agnostic Reward Models. *arXiv preprint arXiv:2505.13388*, 2025. URL https://arxiv.org/abs/2505.13388.

Ralph Allan Bradley and Milton E. Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3–4):324–345, 1952. doi: 10.2307/2334029. URL https://doi.org/10.2307/2334029.

Yuyan Bu, Liangyu Huo, Yi Jing, and Qing Yang. Beyond Excess and Deficiency: Adaptive Length Bias Mitigation in Reward Models for RLHF. In *Findings of the Conference of the North American Chapter of the ACL (NAACL) 2025*, 2025. doi: 10.18653/v1/2025.findings-naacl.169. URL https://doi.org/10.18653/v1/2025.findings-naacl.169.

Sanjiban Choudhury. Process reward models for llm agents: Practical framework and directions. *arXiv preprint arXiv:2502.10325*, 2025. URL https://arxiv.org/abs/2502.10325.

Thomas Coste, Usman Anwar, Robert Kirk, and David Krueger. Reward Model Ensembles Help Mitigate Overoptimization. In *ICLR 2024 poster*, 2024. doi: 10.48550/arXiv.2310.02743. URL https://doi.org/10.48550/arXiv.2310.02743.

Kaixuan Fan, Kaituo Feng, Manyuan Zhang, Tianshuo Peng, Zhixun Li, Yilei Jiang, Shuang Chen, Peng Pei, Xunliang Cai, and Xiangyu Yue. Exploring Reasoning Reward Model for Agents. In *Findings of the Association for Computational Linguistics (ACL Findings) 2026*, 2026a. doi: 10.18653/v1/2026.findings-acl.95. URL https://doi.org/10.18653/v1/2026.findings-acl.95.

Shengda Fan, Xuyan Ye, Yupeng Huo, Zhi-Yuan Chen, Yiju Guo, Shenzhi Yang, Wenkai Yang, Shuqi Ye, Jingwen Chen, Haotian Chen, Xin Cong, and Yankai Lin. Agentprocessbench: Diagnosing step-level process quality in tool-using agents. *arXiv preprint arXiv:2603.14465*, 2026b. URL https://arxiv.org/abs/2603.14465.

Daniel Fein, Max Lamparth, Violet Xiang, Mykel J. Kochenderfer, and Nick Haber. One bias after another: Mechanistic reward shaping and persistent biases in language reward models. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=20c6jemABm.

Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 10835–10866, 2023. URL https://proceedings.mlr.press/v202/gao23h.html.

Zeyu Huang, Zihan Qiu, Zili Wang, Edoardo M. Ponti, and Ivan Titov. Post-hoc reward calibration: A case study on length bias. *arXiv preprint arXiv:2409.17407*, 2024. URL https://arxiv.org/abs/2409.17407.

Hadi Khalaf, Claudio Mayrink Verdun, Alex Oesterling, Himabindu Lakkaraju, and Flavio du Pin Calmon. Inference-time reward hacking in large language models. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/590a0cc0306c1c63e2d66a51a407718f-Abstract-Conference.html.

Hyeonji Kim, Sujeong Oh, and Sanghack Lee. Mitigating Length Bias in RLHF through a Causal Lens. In *AAAI Conference on Artificial Intelligence (AAAI) 2026*, 2026. doi: 10.1609/aaai.v40i21.38806. URL https://doi.org/10.1609/aaai.v40i21.38806.

Sunghwan Kim, Dongjin Kang, Taeyoon Kwon, Hyungjoo Chae, Dongha Lee, and Jinyoung Yeo. Rethinking reward model evaluation through the lens of reward overoptimization. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13252–13280, 2025. doi: 10.18653/v1/2025.acl-long.649. URL https://doi.org/10.18653/v1/2025.acl-long.649.

Nathan Lambert, Valentina Pyatkin, Jacob Daniel Morrison, Lester James V. Miranda, Bill Yuchen Lin, K. Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, Noah A. Smith, and Hanna Hajishirzi. RewardBench: Evaluating Reward Models for Language Modeling. In *Findings of the Conference of the North American Chapter of the ACL (NAACL) 2025*, 2025. doi: 10.18653/v1/2025.findings-naacl.96. URL https://doi.org/10.18653/v1/2025.findings-naacl.96.

Eddie Landesberg. When llm judge scores look good but best-of-n decisions fail, 2026. URL https://arxiv.org/abs/2603.12520.

Dawei Li, Yuguang Yao, Zhen Tan, Huan Liu, and Ruocheng Guo. Toolprmbench: Evaluating and advancing process reward models for tool-using agents. *arXiv preprint arXiv:2601.12294*, 2026. URL https://arxiv.org/abs/2601.12294.

Renhao Li, Jianhong Tu, Yang Su, Yantao Liu, Fei Huang, Hamid Alinejad-Rokny, Derek F. Wong, Junyang Lin, and Min Yang. Toolrm: Towards agentic tool-use reward modeling. *arXiv preprint arXiv:2510.26167*, 2025. URL https://arxiv.org/abs/2510.26167.

Chris Yuhao Liu, Liang Zeng, Yuzhen Xiao, Jujie He, Jiacai Liu, Chaojie Wang, Rui Yan, Wei Shen, Fuxiang Zhang, Jiacheng Xu, Yang Liu, and Yahui Zhou. Skywork-Reward-V2: Scaling Preference Data Curation via Human-AI Synergy. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2507.01352.

Yantao Liu, Zijun Yao, Rui Min, Yixin Cao, Lei Hou, and Juanzi Li. Rm-bench: Benchmarking reward models of language models with subtlety and style. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2410.16184.

Jiarui Lu, Thomas Holleis, Yizhe Zhang, Bernhard Aumayer, Feng Nan, Felix Bai, Shuang Ma, Shen Ma, Mengyu Li, Guoli Yin, Zirui Wang, and Ruoming Pang. Toolsandbox: A stateful, conversational, interactive evaluation benchmark for llm tool use capabilities. *arXiv preprint arXiv:2408.04682*, 2024. URL https://arxiv.org/abs/2408.04682.

Tianyi Men, Zhuoran Jin, Pengfei Cao, Yubo Chen, Kang Liu, and Jun Zhao. Agent-RewardBench: Towards a Unified Benchmark for Reward Modeling across Perception, Planning, and Safety in Real-World Multimodal Agents. In *Annual Meeting of the Association for Computational Linguistics (ACL) 2025*, 2025. doi: 10.48550/arXiv.2506.21252. URL https://doi.org/10.48550/arXiv.2506.21252.

Hao Peng, Yunjia Qi, Xiaozhi Wang, Zijun Yao, Bin Xu, Lei Hou, and Juanzi Li. Agentic reward modeling: Integrating human preferences with verifiable correctness signals for reliable reward systems. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15934–15949, 2025. doi: 10.18653/v1/2025.acl-long.775. URL https://doi.org/10.18653/v1/2025.acl-long.775.

Wenjie Qiu, Xuqin Zhang, Yihang Zhang, Aoran Wang, Zongzhang Zhang, and Yang Yu. Rmeval: Bridging reward model evaluation and policy evaluation through ranking consistency. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, 2026. doi: 10.1145/3770855.3817593. URL https://doi.org/10.1145/3770855.3817593.

Rafael Rafailov, Yaswanth Chittepu, Ryan Park, Harshit S. Sikchi, Joey Hejna, W. Bradley Knox, Chelsea Finn, and S. Niekum. Scaling Laws for Reward Model Overoptimization in Direct Alignment Algorithms. In *NeurIPS 2024 poster*, 2024. doi: 10.48550/arXiv.2406.02900. URL https://doi.org/10.48550/arXiv.2406.02900.

Amit Roth, Ankur Samanta, Matan Halevy, Yoav Levine, and Yonathan Efroni. Hack-verifiable environments: Towards evaluating reward hacking at scale. *arXiv preprint arXiv:2605.20744*, 2026. URL https://arxiv.org/abs/2605.20744.

Wei Shen, Rui Zheng, Wenyu Zhan, Jun Zhao, Shihan Dou, Tao Gui, Qi Zhang, and Xuanjing Huang. Loose lips sink ships: Mitigating length bias in reinforcement learning from human feedback. *arXiv preprint arXiv:2310.05199*, 2023. URL https://arxiv.org/abs/2310.05199.

Lin Shi, Chiyu Ma, Wenhua Liang, Xingjian Diao, Weicheng Ma, and Soroush Vosoughi. Judging the judges: A systematic study of position bias in llm-as-a-judge. In *Proceedings of IJCNLP-AACL*, pp. 292–314, 2025. URL https://aclanthology.org/2025.ijcnlp-long.18/.

Ibne Farabi Shihab, Fariya Afrin, Sanjeda Akter, and Anuj Sharma. Est-prm: Stress-testing process reward models before they become load-bearing. *arXiv preprint arXiv:2606.00437*, 2026. URL https://arxiv.org/abs/2606.00437.

Snorkel AI. Tau2-Bench Airline With Code Agents. Hugging Face dataset, Apache-2.0, 2026. URL https://huggingface.co/datasets/snorkelai/Tau2-Bench-Airline-With-Code-Agents. 500 released tau2-bench airline trajectories from five frontier models, 50 tasks x 10 candidates, with executable reward and reward_breakdown fields.

Mingyang Song, Zhaochen Su, Xiaoye Qu, Jiawei Zhou, and Yu Cheng. PRMBench: A Fine-grained and Challenging Benchmark for Process-Level Reward Models. In *Annual Meeting of the Association for Computational Linguistics (ACL) 2025*, 2025. doi: 10.48550/arXiv.2501.03124. URL https://doi.org/10.48550/arXiv.2501.03124.

Pragya Srivastava, Harman Singh, Rahul Madhavan, Gandharv Patil, Sravanti Addepalli, Arun Suggala, Rengarajan Aravamudhan, Soumya Sharma, Anirban Laha, Aravindan Raghuveer, Karthikeyan Shanmugam, and Doina Precup. Robust reward modeling via causal rubrics. *arXiv preprint arXiv:2506.16507*, 2025. URL https://arxiv.org/abs/2506.16507.

Kunvar Thaman. Reward hacking benchmark: Measuring exploits in llm agents with tool use. *arXiv preprint arXiv:2605.02964*, 2026. URL https://arxiv.org/abs/2605.02964.

Atticus Wang, Iván Arcuschin, and Arthur Conmy. Automatically finding reward model biases. *arXiv preprint arXiv:2602.15222*, 2026a. URL https://arxiv.org/abs/2602.15222.

Haoxiang Wang, Wei Xiong, Tengyang Xie, Han Zhao, and Tong Zhang. Interpretable Preferences via Multi-Objective Reward Modeling and Mixture-of-Experts. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, 2024. URL https://arxiv.org/abs/2406.12845.

Jiaxuan Wang, Yulan Hu, Wenjin Yang, Zheng Pan, Xin Li, and Lan-Zhe Guo. Aligning agents via planning: A benchmark for trajectory-level reward modeling. *arXiv preprint arXiv:2604.08178*, 2026b. URL https://arxiv.org/abs/2604.08178.

Xuekang Wang, Zhuoyuan Hao, Shuo Hou, Hao Peng, Juanzi Li, and Xiaozhi Wang. Reproducing, analyzing, and detecting reward hacking in rubric-based reinforcement learning. *arXiv preprint arXiv:2606.04923*, 2026c. URL https://arxiv.org/abs/2606.04923.

Zhaofeng Wu, Michihiro Yasunaga, Andrew Cohen, Yoon Kim, Asli Celikyilmaz, and Marjan Ghazvininejad. rewordbench: Benchmarking and improving the robustness of reward models with transformed inputs. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, 2025. URL https://arxiv.org/abs/2503.11751.

Yu Xia, Jingru Fan, Weize Chen, Siyu Yan, Xin Cong, Zhong Zhang, Yaxi Lu, Yankai Lin, Zhiyuan Liu, and Maosong Sun. AgentRM: Enhancing Agent Generalization with Reward Modeling. In *Annual Meeting of the Association for Computational Linguistics (ACL) 2025*, 2025. doi: 10.48550/arXiv.2502.18407. URL https://doi.org/10.48550/arXiv.2502.18407.

Tong Xie, Andrew Bai, Yuanhao Ban, Yunqi Hong, Haoyu Li, and Cho-Jui Hsieh. When Distance Distracts: Representation Distance Bias in BT-Loss for Reward Models. In *ICML 2026 regular*, 2026. URL https://openreview.net/forum?id=MBk6Pur6RX.

Daniel Yang, Samuel Stante, Florian Redhardt, Lena Libon, Parnian Kassraie, Ido Hakimi, Barna Pásztor, and Andreas Krause. Rewarduq: A unified framework for uncertainty-aware reward models. *arXiv preprint arXiv:2602.24040*, 2026. URL https://arxiv.org/abs/2602.24040.

Xuanchang Zhang, Wei Xiong, Lichang Chen, Tianyi Zhou, Heng Huang, and Tong Zhang. From lists to emojis: How format bias affects model alignment. *arXiv preprint arXiv:2409.11704*, 2024. URL https://arxiv.org/abs/2409.11704.

Zhenru Zhang, Chujie Zheng, Yangzhen Wu, Beichen Zhang, Runji Lin, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. The Lessons of Developing Process Reward Models in Mathematical Reasoning. In *Findings of the Association for Computational Linguistics: ACL 2025*, 2025. doi: 10.18653/v1/2025.findings-acl.547. URL https://doi.org/10.18653/v1/2025.findings-acl.547.

Jian Zhao, Runze Liu, Kaiyan Zhang, Zhimu Zhou, Junqi Gao, Dong Li, Jiafei Lyu, Zhouyi Qian, Biqing Qi, Xiu Li, and Bowen Zhou. GenPRM: Scaling Test-Time Compute of Process Reward Models via Generative Reasoning. In *AAAI Conference on Artificial Intelligence (AAAI) 2026*, 2026a. doi: 10.48550/arXiv.2504.00891. URL https://doi.org/10.48550/arXiv.2504.00891.

Kangwen Zhao, Jianfeng Cai, Jinhua Zhu, Ruopei Sun, Dongyun Xue, Wengang Zhou, Li Li, and Houqiang Li. Bias Fitting to Mitigate Length Bias of Reward Model in RLHF. In *Annual Meeting of the Association for Computational Linguistics (ACL) 2026*, 2026b. doi: 10.18653/v1/2026.acl-long.133. URL https://doi.org/10.18653/v1/2026.acl-long.133.

Enyu Zhou, Guodong Zheng, Binghai Wang, Zhiheng Xi, Shihan Dou, Rong Bao, Wei Shen, Limao Xiong, Jessica Fan, Yurong Mou, Rui Zheng, Tao Gui, Qi Zhang, and Xuanjing Huang. RMB: Comprehensively Benchmarking Reward Models in LLM Alignment. In *ICLR 2025 Poster*, 2025. doi: 10.48550/arXiv.2410.09893. URL https://doi.org/10.48550/arXiv.2410.09893.

Youheng Zhu and Yiping Lu. On the Power of (Approximate) Reward Models for Inference-Time Scaling: Sequential Monte Carlo and Beyond. In *ICML 2026 regular*, 2026. URL https://openreview.net/forum?id=MrIDZjIsNF.

A Complete primary-cell table

Table 6 lists all 18 balanced-grid cells.

Table 6: All 18 balanced-N64 primary cells. Each interval uses 10,000 pool-bootstrap draws; Holm correction is over the six cells within an estimand. The positive Selection rows are index-bound descriptive diagnostics under the fixed cyclic, non-counterbalanced role assignment, not causal role estimates.

Estimand	Family	Domain	Estimate	95% CI	State
Dose	D-Scalar	HVE	-0.0252	$[-0.0283, -0.0222]$	supported negative
Dose	D-Scalar	ToolSandbox	$+0.0176$	$[+0.0059, +0.0285]$	supported positive
Dose	D-Vector	HVE	-0.0263	$[-0.0265, -0.0260]$	supported negative
Dose	D-Vector	ToolSandbox	-0.0113	$[-0.0122, -0.0105]$	supported negative
Dose	G-Rubric	HVE	-0.0425	$[-0.0458, -0.0392]$	supported negative
Dose	G-Rubric	ToolSandbox	-0.2288	$[-0.2697, -0.1846]$	supported negative
Insertion cond.	D-Scalar	HVE	$+0.0863$	$[+0.0824, +0.0905]$	supported positive
Insertion cond.	D-Scalar	ToolSandbox	-0.0979	$[-0.1123, -0.0839]$	supported negative
Insertion cond.	D-Vector	HVE	$+0.0176$	$[+0.0174, +0.0178]$	supported positive
Insertion cond.	D-Vector	ToolSandbox	$+0.0133$	$[+0.0125, +0.0140]$	supported positive
Insertion cond.	G-Rubric	HVE	-0.0764	$[-0.0813, -0.0714]$	supported negative
Insertion cond.	G-Rubric	ToolSandbox	-0.0638	$[-0.0895, -0.0387]$	supported negative
Selection	D-Scalar	HVE	$+0.7685$	$[+0.6736, +0.8617]$	index-bound positive
Selection	D-Scalar	ToolSandbox	-0.0799	$[-0.1804, +0.0125]$	inconclusive
Selection	D-Vector	HVE	$+0.6551$	$[+0.5386, +0.7664]$	index-bound positive
Selection	D-Vector	ToolSandbox	$+0.1032$	$[+0.0098, +0.1883]$	inconclusive after Holm
Selection	G-Rubric	HVE	-0.0305	$[-0.1487, +0.0837]$	inconclusive
Selection	G-Rubric	ToolSandbox	$+0.0054$	$[-0.0093, +0.0218]$	inconclusive

B CONSTRUCTED-GRID RESPONSE WITHIN EACH INSERTION CONDITION

Each slope fits the original q axis within one insertion-condition label and averages over carriers. OLS fits a linear rate to the normalized regressor $x = q/256$; it is not the endpoint difference between $q = 0$ and $q = 256$, since curvature and saturation separate the two quantities. Because $q = 0$ inserts no turn and positive doses use realized carrier strings, the values describe the combined protocol response, not a pure token-length effect. For D-Scalar, the pooled slope hides sign changes across conditions in both domains. The first two labels occupy physical slot 1 but have different turn-index metadata, so the design has two physical positions, not three. The two D-Scalar/ToolSandbox entries shown as $+0.0500$ are $+0.04997$ and $+0.05001$ before rounding. All estimates are exploratory and unadjusted. The released `score_sensitivity.csv` contains the full 54-row condition-by-carrier breakdown.

Table 7: Constructed-grid slope within each insertion condition, averaged over carriers.

Family	Domain	observation-after	action-before	action-after
D-Scalar	HVE	-0.0573	-0.0540	$+0.0356$
D-Scalar	ToolSandbox	$+0.0500$	$+0.0500$	-0.0471
D-Vector	HVE	-0.0313	-0.0312	-0.0163
D-Vector	ToolSandbox	-0.0141	-0.0137	-0.0063
G-Rubric	HVE	-0.0159	-0.0250	-0.0867
G-Rubric	ToolSandbox	-0.2483	-0.2257	-0.2125

C DENOMINATORS AND PROVENANCE

Production matched every planned denominator (Table 8). The provenance crosswalk is in Tables 9–10.

Table 8: Closed production denominators.

Unit	Expected	Completed
Pools	64	64
Base trajectories	4,096	4,096
Intervention variants	221,184	221,184
Score rows	2,064,384	2,064,384
Selection rows	1,119,744	1,119,744
Abstentions / errors / quarantines / silent exclusions	0/0/0/0	0/0/0/0

Table 9: Core analysis provenance. "Fresh/reuse" distinguishes independent pools from reuse of hypothesis-generating or sensitivity data.

Analysis	Data / date	Hypothesis source	Status / multiplicity family	Fresh pools / reuse	Release artifact / validation
balanced N64	constructed, Sep 11–12	earlier run; plan set before new scores	primary; six cells per estimand, Holm	fresh seeds; earlier run only generated design	tables and pool-level records; logical gate
coarsening	N64, Sep 13	generated from N64 result	mechanism illustration; descriptive	no; reuses scores	gap analysis; logical gate
Block P	control, Sep 13	preregistered control	secondary; three checkpoints	fresh control pools	Block-P scores; logical gate
Block S	control, Sep 13	position-confound concern	secondary; four contrasts, Holm	fresh pools; earlier hypothesis	Block-S scores; logical gate
position confirm	HVE, Sep 13	exploratory probe; follow-up design fixed before testing	prospective confirmation; two checkpoints, Holm	fresh pools; earlier hypothesis	position-confirm scores; logical gate
full context	airline, Sep 18	truncation concern	primary natural transfer; three checkpoints	37 fresh pools	five-enhancement/experiment-1-full-context/analysis-r4; logical gate
matched C4K	airline, Sep 18	observation-policy sensitivity	paired secondary; three families, Holm	same 37 pools	five-enhancement/experiment-1-full-context/analysis-r4; logical gate
CPU strengthening	airline and release, Sep 19	confound audit	verified within CPU boundary	existing records; deterministic rerun	ars-cpu-strengthening-20260919; 21/21 logical gates

Table 10: Closure and follow-up provenance. Status labels do not promote secondary, exploratory, or descriptive evidence.

Analysis	Data / date	Hypothesis source	Status / multiplicity family	Fresh pools / reuse	Release artifact / validation
all-50 extension	airline, Sep 19	13 saturated tasks omitted	descriptive; two checkpoints	13 newly scored tasks; 50-task denominator	ars-r2-all-task-gpu-20260919; 18/18 gates
short-pool split	retail / airline, Sep 18	preregistered held-out transfer	primary held-out $n = 2$; development/sensitivity separate	seven eligible pools	five-enhancement/experiment-2-split-audit/analysis-r4; logical gate
ToolRM	airline / retail, Sep 18	specialized evaluator extension	secondary pairwise; degree-nine round robin	same natural pools	five-enhancement/experiment-3-toolrm-formal and split audit; logical gate
slot × metadata	constructed, Sep 18	position non-identification	secondary factorial; nine contrasts, Holm	32 fresh pools	five-enhancement/analysis-4-factorial; logical gate
nested factorial	HVE, Sep 20	dose confounding	33 planned contrasts, Holm	32 frozen pools; new variants	ars-r4-nested-factorial-20260920; 18/18 gates
controlled context	airline, Sep 20	informative C4K	9 score and 45 selection contrasts, Holm	same 37 pools; new projections	ars-r5-controlled-context-20260920; 20/20 gates
policy / resolution	airline, Sep 20	audit-triggered after full-context results	D-Scalar manifest reporting target; other rows sensitivity	reuses 1,110 scores	ars-p0-policy-resolution-20260920 v2; 20/20 gates
GPU closure	airline / retail, Sep 20–21	audit-triggered frozen closure plan	three seeds; repeatability, serialization and matched-slot families	three source–domain cells; feasible subset reported	ars-gpu-closure-20260920; internal logical gate
graph / length sensitivity	same 69 feasible pools, Sep 23–24	follow-up implementation checks	3 graph seeds × 2 orientations; paired padded–dynamic scoring	reuses pools and 996 padded calls; 996 new dynamic calls	ars-gpu-sensitivity-20260924; independent audit `VERIFIED`
non-GPU audit	retained records, Sep 23	support/ provenance audit	derived CPU evidence; no new model outputs	frozen Gate 4 unchanged	ars-nongpu-revision-20260923; hashes + rerun script
prompt and composition	airline, Sep 18	follow-up sensitivities	secondary paired contrasts; Holm	same pools; rerun held-out	five-enhancement/experiment-5 and split audit; logical gate
clipping	raw N64, Sep 11/18	processing audit	measurement-chain audit; no test	all N64 rows reused	D-Vector clipping receipt; logical counts

D NON-GPU PROVENANCE AND PAIR-SUPPORT AUDIT

The retained AgentSuite label is `AgentSuite/tau-bench-trajectories`, with snapshot date 2026-09-18. Preflight metadata list 4,950 rows, 165 tasks, and 30 generator files; the fixed ten-generator policy was written before outcome analysis. The released `agentsuite_task_generator_crosswalk.csv` maps each task split to each selected generator (1,650 rows). SHA-256 `10a4950364aeef846c12be75f9de5d06ae07999d5d054c114f4d38d5baa3d55d` authenticates the retained `agentsuite_data_audit.json` metadata record only. The preflight reports `hash_checks_performed=false`; the raw snapshot is absent locally, so we cannot provide or infer its content hash, upstream accession, or licence.

Run `python` with `scripts/audit_ars_nongpu_revision_20260923.py` for the CPU audit. The script reads frozen records and writes only `ars-nongpu-revision-20260923`: a

provenance manifest with input/output hashes, the crosswalk, reconstructed edge-level cycle metadata, and the summary below. It does not score a model or alter Gate 4. The release omits AgentSuite candidate-level labels, raw pair probabilities, weights, and adapters, so it supports only a bounded numerical audit.

D.1 Pairwise supervised support and test edges

Static inspection of `candidate_pairs_for_tasks` shows that every retained training, validation, and calibration pair contains one success and one failure within a task. The bounded release retains pair counts but not example manifests. Table 11 therefore combines constructor inspection with immutable dataset records; it is not a row-level revalidation.

Table 11: Retained pair counts. Every listed pair is success–failure; success–success and failure–failure counts are zero by construction.

Seed	Train	Validation	Calibration
20260920	232	78	66
20260921	234	68	66
20260922	234	62	66

For original airline, candidate IDs and executable labels permit an exact CPU reconstruction of the predeclared cycles using the released graph seed and orientation rule (Table 12). Same-outcome edges lie outside supervised pair support; the classifier was never trained or calibrated to order their members. The initial CPU-only release did not retain AgentSuite labels or edge outputs, so Table 12 remains original-airline only. The later audit-triggered freeze separately retains labels for the 69 feasible mixed pools used below; it does not recover excluded or saturated pools, for which we impute no composition.

Table 12: Original-airline cycle composition. "Off support" is the sum of success–success and failure–failure edges.

Cycle set	Pools	S–F	S–S	F–F	Off support
All full 10-edge cycles	37	136	117	117	234/370 (63.2%)
Feasible full cycles	16	56	71	33	104/160 (65.0%)
Completed top-5 cycles	16	36	22	22	44/80 (55.0%)

The audit-triggered freeze then constructed the complete opposite-outcome graph for each of the 69 feasible mixed pools before model loading: 257 edges from original airline, 671 from AgentSuite airline, and 239 from AgentSuite retail. All 1,167 ordered, padded calls completed, and every support graph was connected. Table 13 compares this outcome-conditioned graph with the primary sparse full cycle on the same pools.

Table 13: Label-conditioned support diagnostic. Agreement is with the primary sparse-cycle winner. The paired row reports support minus primary with a 95% pool-bootstrap interval.

Graph	Pools	Winner agreement	Success@1	Excess utility
Primary sparse full cycle	69	1.000	.377	−.164
Support-matched, ID-oriented	69	.377	.406	−.135
Paired difference	69	—	+.029 [−.058, .101]	+.029 [−.058, .101]

The returned identity changed in 43/69 pools, but only eight pools changed binary outcome: five from failure to success and three from success to failure. Because the support graph is denser, uses labels to choose edges, and fixes candidate-ID presentation order, this comparison does not identify a pure support effect or estimate an outcome-blind deployment policy.

E Detailed limitations and completed GPU sensitivities

The primary natural analysis uses mixed tasks from one airline source. The closure adds a dated internal AgentSuite source and two related transactional domains; retail contributes 15 mixed test

pools. Original-airline matched-slot levels use 16/37 pools: included/excluded mean success shares are $.619/.410$, median longest-single lengths are $19{,}186/48{,}675$ tokens, and median longest-pair lengths are $26{,}547/70{,}617$. Feasibility is length-dependent and associated with outcome. The original $n = 37$ MDE is .177, so non-detection is not evidence of equivalence.

Each legacy family is represented by one checkpoint. T-Verifier uses 140 training examples and one seed; the joint evaluator has three seeds and independent calibration but remains a local diagnostic. Exact supplied-trace checks do not address fuzzy or semantic overlap, undisclosed sources, or pretraining contamination. No mature agent-trajectory ORM/PRM is matched on the same pools, inputs, and policies (Xia et al., 2025; Agarwal et al., 2025; Li et al., 2026; Song et al., 2025); the claims are consequently restricted to this measured setting.

The policy/resolution audit followed the original results. Deterministic repeats do not measure stochastic decoding or software-version drift; stable IDs encode provenance. Audit-triggered scoring added three graph seeds, both orientations, a paired padded–dynamic comparison stratified by realized input length, and the label-conditioned support graph on the same 69 feasible mixed pools. Winner identities changed substantially across graph/orientation and support variants, and some changed between padded and dynamic calls, so the original policy comparison is implementation-conditional. These checks add no all-pairs reference; the support arm is outcome-conditioned and orientation-specific, and the $>$ 32k bin contains only nine edges. The evaluator's 8,192-token training support is therefore insufficient for a general long-context or selector-class claim.

Constructed and follow-up studies have separate boundaries. ToolSandbox repeats four short action forms; the earlier retail split has two pools; controlled projections change visible content. The 131k arm uses YaRN extrapolation beyond 40,960 positions, unlike native 4K/8K/16K projections. Factorial carriers are length-matched but not representationally equivalent, and cyclic, non-counterbalanced roles make role-H effects index-bound. These secondary and exploratory analyses do not broaden the primary natural-transfer claim.

F GPU CLOSURE DETAILS

The closure package contains metadata, summaries, and audit records. The three formal seeds used the same task split and checkpoint rule; seed-specific temperatures were fitted on the independent calibration tasks. Table 17 reports cell estimates without pooling across seeds. These are descriptive seed replications around the primary seed in Table 5; task/pool remains the inferential unit.

Padding implementation. We used the adapter tokenizer's pad token ID and fell back to EOS only when no pad token was available. Real tokens occupied positions 0 through $L - 1$; padding was appended on the right, with attention-mask entries set to one for real tokens and zero for padding. The forward call supplied `input_ids` and `attention_mask` but no explicit `position_ids`, so position construction came from the model implementation. The two pointwise calls used identical IDs, masks, and deterministic inference. They matched the exposure count and were averaged, not treated as independent samples. The audit-triggered paired check reused 996 matching padded calls with source provenance and ran 996 dynamic-unpadded calls. After accounting for edges shared by the full and top-five cycles, all 1,035 manifest edges were paired; no call failed, was truncated, or was excluded.

Table 14: Graph and orientation sensitivity on the same 69 feasible mixed pools. Winner agreement and mean within-pool Spearman rank correlation (ρ) are relative to seed 20260920 in the forward orientation, separately for each cycle. Excess utility is the mean over pools. These variants reuse task samples and are implementation diagnostics, not independent replications.

Graph seed	Orientation	Full agree	Full excess	Full ρ	Top-5 agree	Top-5 excess	Top-5 ρ
20260920	forward	1.000	−.164	1.000	1.000	−.217	1.000
20260920	reverse	.290	−.207	.602	.348	−.159	.426
20260921	forward	.304	−.178	.704	.522	−.159	.712
20260921	reverse	.348	−.207	.713	.478	−.174	.630
20260922	forward	.333	−.164	.713	.565	−.145	.674
20260922	reverse	.435	−.207	.719	.493	−.188	.641

Table 15: Paired padded–dynamic edge-score sensitivity by realized non-padding length. Drift is the absolute change in the probability assigned to candidate A; brackets are 95% pool-bootstrap intervals. Memory is median peak allocated GiB and latency is median wall seconds per call, shown as dynamic/padded. The $>$ 32k row has only nine edges.

Length bin	Edges	Mean drift [95% CI]	Max drift	Memory (GiB)	Latency (s)
$\leq$ 8k	365	.0121 [.0106, .0136]	.0922	8.54/9.79	.47/2.51
8–16k	536	.0182 [.0164, .0200]	.2141	9.19/10.12	.93/3.06
16–32k	125	.0199 [.0165, .0236]	.0844	10.29/11.32	1.85/5.78
$>$ 32k	9	.0143 [.0093, .0219]	.0417	12.57/23.56	4.30/28.66

Table 16: Cycle-level padded–dynamic comparison on the 69 paired pools. Coverage is 69/69 in every row; there were no OOMs, context exclusions, truncations, or non-finite outputs. Identical aggregate utility does not imply identical selected trajectories.

Cycle	Padded success@1	Dynamic success@1	Padded excess	Dynamic excess	Winner agreement
Full cycle	.377	.377	$-.164$	$-.164$	$62/69 = .899$
Top-5 cycle	.319	.319	$-.217$	$-.217$	$67/69 = .971$

Table 17: Joint-evaluator seed and cell summaries. Values are point estimates from the mixed pools. The source–domain cells are not pooled, and the three seeds are not additional task samples.

Seed	Cell	Mixed pools	Success@1	Excess utility	AUC	Spearman
20260920	Original airline	37	.216	$-.284$	.435	$-.094$
20260921	Original airline	37	.514	$+.014$	.587	$+.137$
20260922	Original airline	37	.432	$-.068$	.491	$-.005$
20260920	AgentSuite airline	38	.342	$-.129$	.506	.001
20260921	AgentSuite airline	38	.289	$-.182$	.513	.001
20260922	AgentSuite airline	38	.289	$-.182$	.485	$-.026$
20260920	AgentSuite retail	15	.533	$-.100$	.696	.227
20260921	AgentSuite retail	15	.800	$+.167$	.776	.343
20260922	AgentSuite retail	15	.667	$+.033$	.738	.282

Table 18: Seed-specific training and calibration records. The selected epoch minimizes joint validation negative log likelihood; test and external outcomes were not consulted.

Seed	Epoch	Joint val. NLL	Early stop	Point temp.	Pair temp.	Cal. n
20260920	3	.6147	no	3.3995	1.1605	268
20260921	2	.5449	yes	3.5925	.3126	268
20260922	2	.5380	yes	4.3090	2.3514	268

The training pointwise sample contained 800 balanced examples for every seed; the selected pairwise samples contained 232, 234 and 234 examples for seeds 20260920, 20260921 and 20260922, respectively. Validation contained 376, 366 and 360 joint examples, and calibration contained 268 examples per seed. No training, validation or calibration rendering was truncated.

Table 19: Stability and matched input-token-slot closure checks.

Quantity	Verified value
Process repeats	5
All-five winner agreement	1.000
Repeat ICC(2,1), ICC(2,k)	1.000, 1.000
Repeat maximum score drift / rank displacement	0 / 0
Serialization variants	3
All-template winner agreement / pairwise agreement	.568 [.405, .730] / .703 [.586, .811]
Serialization mean Pearson / Spearman	.974 / .969
Serialization ICC(2,1) / ICC(2,k)	.973 / .991
Serialization maximum drift / rank displacement	.1372 / 7
Serialization receipt	frozen variant label + file name/size/schema receipt; alternative winner recheck required
Matched-slot feasibility gates	102
Completed mixed pools / memory exclusions / saturated exclusions	69 / 21 / 12
Policy rows in completed pools	207
Numeric values compared by reproduction	7,136
Maximum absolute difference	0

The three matched-slot policies used identical padded token slots within each completed pool. Original-airline completion was 16/37 because 21 pools crossed the memory gate; AgentSuite airline contributed 38 mixed pools after 12 saturated exclusions; all 15 retail test pools completed. Runtime OOM exclusions were atomic. Aggregate wall time was 2,453/4,160/3,307 seconds for pointwise/pairwise/top-5; median peak allocated memory was 10.22 GiB (range 8.26–24.63 GiB). Matching slots did not equalize compute or latency. We did not rescore excluded pools with a fallback, so coverage and conditional utility have no all-pool fallback estimate.

Table 20: Secondary feasible-only matched-slot levels. Each row uses fresh padded rescoring and the same scheduled token slots within a pool. Brackets are 95% paired pool-bootstrap intervals. These binary-outcome contrast and policy-plus-graph diagnostics do not rank selector classes. Coverage was 16/37, 38/38 mixed, and 15/15 mixed pools; excluded pools have no fallback estimate.

Cell	Policy	n	Success@1	Excess utility	Regret
Original airline	Pointwise argmax	16	.188 [0, .375]	−.431 [−.675, −.131]	.813
	Pairwise Bradley–Terry	16	.250 [.063, .500]	−.369 [−.588, −.125]	.750
	Top-5 pairwise rerank	16	.125 [0, .313]	−.494 [−.694, −.244]	.875
AgentSuite airline	Pointwise argmax	38	.342 [.184, .500]	−.129 [−.271, .018]	.658
	Pairwise Bradley–Terry	38	.316 [.184, .474]	−.155 [−.292, −.011]	.684
	Top-5 pairwise rerank	38	.289 [.158, .447]	−.182 [−.321, −.045]	.711
AgentSuite retail	Pointwise argmax	15	.600 [.333, .867]	−.033 [−.213, .140]	.400
	Pairwise Bradley–Terry	15	.667 [.400, .867]	+.033 [−.140, .200]	.333
	Top-5 pairwise rerank	15	.600 [.333, .867]	−.033 [−.187, .107]	.400

Table 21: Complete paired contrasts for the matched input-token-slot comparison. P denotes pointwise argmax, B denotes Bradley–Terry aggregation, and T5 denotes pointwise top-5 followed by pairwise reranking. Each estimate is right minus left on the same feasible pools; brackets are 95% paired pool-bootstrap intervals, s_d is the paired standard deviation, and p_{Holm} is corrected within the three-policy family for each cell and metric. The final column gives the 80%-power MDE at $\alpha = .05$ and $\alpha = .05/3$.

Cell	Contrast	Metric	Estimate	95% CI	s_d	p_{Holm}	$\mathrm{MDE}_{.05}$ / $\mathrm{MDE}_{.05/3}$
Original airline	P–B	success@1	$-.063$	$[-.250, .125]$	.443	1.000	.332/.397
Original airline	P–B	excess	$-.063$	$[-.250, .125]$	.443	1.000	.332/.397
Original airline	P–B	regret	$+.063$	$[-.125, .250]$	.443	1.000	.332/.397
Original airline	P–T5	success@1	$+.063$	$[0, .188]$	.250	1.000	.187/.224
Original airline	P–T5	excess	$+.063$	$[0, .188]$	.250	1.000	.187/.224
Original airline	P–T5	regret	$-.063$	$[-.188, 0]$	.250	1.000	.187/.224
Original airline	B–T5	success@1	$+.125$	$[0, .313]$	.342	1.000	.256/.306
Original airline	B–T5	excess	$+.125$	$[0, .313]$	.342	1.000	.256/.306
Original airline	B–T5	regret	$-.125$	$[-.313, 0]$	.342	1.000	.256/.306
AgentSuite airline	P–B	success@1	$+.026$	$[-.132, .184]$	.492	1.000	.230/.269
AgentSuite airline	P–B	excess	$+.026$	$[-.132, .184]$	.492	1.000	.230/.269
AgentSuite airline	P–B	regret	$-.026$	$[-.184, .132]$	.492	1.000	.230/.269
AgentSuite airline	P–T5	success@1	$+.053$	$[-.079, .184]$	.462	1.000	.216/.252
AgentSuite airline	P–T5	excess	$+.053$	$[-.079, .211]$	.462	1.000	.216/.252
AgentSuite airline	P–T5	regret	$-.053$	$[-.211, .105]$	.462	1.000	.216/.252
AgentSuite airline	B–T5	success@1	$+.026$	$[-.079, .132]$	.367	1.000	.171/.200
AgentSuite airline	B–T5	excess	$+.026$	$[-.079, .132]$	.367	1.000	.171/.200
AgentSuite airline	B–T5	regret	$-.026$	$[-.132, .079]$	.367	1.000	.171/.200
AgentSuite retail	P–B	success@1	$-.067$	$[-.267, .133]$	.458	1.000	.356/.427
AgentSuite retail	P–B	excess	$-.067$	$[-.267, .133]$	.458	1.000	.356/.427
AgentSuite retail	P–B	regret	$+.067$	$[-.133, .267]$	.458	1.000	.356/.427
AgentSuite retail	P–T5	success@1	.000	$[-.200, .200]$	.378	1.000	.294/.353
AgentSuite retail	P–T5	excess	.000	$[-.200, .200]$	.378	1.000	.294/.353
AgentSuite retail	P–T5	regret	.000	$[-.200, .200]$	.378	1.000	.294/.353
AgentSuite retail	B–T5	success@1	$+.067$	$[0, .200]$	.258	1.000	.201/.241
AgentSuite retail	B–T5	excess	$+.067$	$[0, .200]$	.258	1.000	.201/.241
AgentSuite retail	B–T5	regret	$-.067$	$[-.200, 0]$	.258	1.000	.201/.241

Table 22: Feasible-versus-excluded composition for the original-airline matched-slot gate. Length summaries are median [IQR] over pools; success share is the mean within-pool share.

Group	Pools	Success share	Longest single / pair tokens
Included	16	.619	19,186 [13,258, 22,542] /26,547 [21,445, 33,132]
Excluded	21	.410	48,675 [38,827, 82,654] /70,617 [62,976, 122,737]
Included − excluded	–	$+.209$ $[+.023, +.385]$	$-29{,}490$ $[-59{,}593, -22{,}450]$ / $-44{,}070$ $[-85{,}789, -34{,}470]$

G NAMED-POLICY AND SCORE-RESOLUTION AUDIT

The P0 audit uses the 37 task pools as independent units. The formal and reproduction runs each generated 15 analysis artifacts. CSV files were byte-identical, while JSON files matched after normalizing only the attempt ID. An independent implementation reconstructed 444 policy-pool rows, 3,000 permutation draws, 555 epsilon rows, 592 rounding rows, 111 margin rows, and all 12 power rows. All 20 integrity gates passed.

The full release includes policy-specific bootstrap intervals, epsilon-set membership, rounding results, raw and normalized margins, explicit candidate-order sequences, and all 1,000 order-permutation draws. The validation report marks the analysis `VERIFIED` and records cautions about the post-result hierarchy, mixed-outcome conditioning, checkpoint scope, identifier provenance, and planning-model assumptions.

Table 23: Fixed epsilon grid. Each evaluator block reports mean epsilon-top-set size $|A_i^\epsilon|$, the fraction of pools with a singleton epsilon top set, and random-within-set utility U_{rnd}. Percentiles use NumPy's linear rule, and zero-spread pools include every candidate.

	D-Scalar			G-Rubric			T-Verifier		
ϵ	$\lvert A^\epsilon\rvert$	singleton	U_{rnd}	$\lvert A^\epsilon\rvert$	singleton	U_{rnd}	$\lvert A^\epsilon\rvert$	singleton	U_{rnd}
0	1.000	1.000	.486	3.676	.162	.412	1.000	1.000	.514
.01	1.108	.892	.500	3.676	.162	.412	1.297	.703	.527
.02	1.243	.811	.500	3.676	.162	.412	1.541	.595	.536
.05	1.432	.703	.486	3.676	.162	.412	2.054	.459	.517
.10	1.703	.568	.473	3.676	.162	.412	2.784	.297	.541

Table 24: Rounding sensitivity on each native scale. Cells give singleton rate / mean top-set size. A dash denotes a resolution not applied to the native integer G-Rubric scale. No jitter is added.

Family	Native scale	exact	6 decimals	4 decimals	3 decimals	2 decimals	1 decimal	0 decimals
D-Scalar	probability	1.000/1.000	1.000/1.000	1.000/1.000	1.000/1.000	.892/1.135	.514/2.054	.000/7.000
G-Rubric	integer 1 to 5	.162/3.676	–	–	–	–	–	.162/3.676
T-Verifier	probability	1.000/1.000	1.000/1.000	.946/1.054	.946/1.162	.649/1.892	.324/4.514	.000/7.730

H Secondary and exploratory follow-ups

All-task population. This descriptive $\mathbb{E}[U]$ estimand is separate from mixed-only natural discrimination and feasible-only matched-slot utility. The released `task_intake.csv` lists all 50 airline tasks (500 candidates; 225 successes); the 13 additional pools comprise four all-success and nine all-failure tasks. D-Scalar and G-Rubric all-task manifest-order utilities are .440 (95% task-bootstrap CI $[.300, .580]$) and .460 ($[.320, .600]$), with paired difference $-.020$ ($[-.200, .160]$). Saturated tasks add no discrimination information. T-Verifier is excluded because those 13 IDs form its training/validation tasks, and the matched-slot arm has no measured fallback outcome for excluded pools.

H.1 Nested factorial decomposition

Table 26 reports the planned dose contrast and the detected component terms discussed in the main text. Sign conventions are content minus matched empty turn, carrier B minus carrier A, post minus pre, and 16 minus 8 tokens. The remaining planned contrasts and per-pool effects are released. The design identifies these terms only for the two carriers, two positions, two token increments, 32 HVE pools, and three fixed checkpoints.

H.2 Controlled visible context

The controlled arm projects the same 370 candidates with one terminal-preserving head-tail rule at 4,096, 8,192, and 16,384 complete prompt tokens. It truncates 229, 118, and 58 candidates at the respective budgets. Formal and reproduction attempts each contain 3,330 scores. Seven of nine planned score contrasts survive Holm correction; none of the 45 selection contrasts does.

This intervention changes the visible content selected by a fixed projection. It does not isolate an architectural context-window effect. The older matched C4K arm uses a different terminal-preserving event cut and remains an observation-policy sensitivity.

The controlled runner records `max_position_embeddings` = 40,960, `rope_scaling` = `none`, the nested-terminal-head-tail projection, and 3,330 score rows. It retains the opening context and terminal event at every budget; 229, 118, and 58 candidates are truncated at 4K, 8K, and 16K. Seven of nine score contrasts survive one Holm family, while none of the 45 prespecified selection contrasts does. Within this native-context diagnostic, score, rank, winner identity, and utility can move differently.

H.3 Short-pool retail and pairwise endpoints

The prespecified held-out split contains `retail_68` and `retail_77`; five pools remain development exploration. Each held-out pool has .9 success share, and all pointwise checkpoints plus ToolRM select a success (utility 1.0, excess $+.1$). The registered decision is `INCONCLUSIVE_SMALL_N`; the seven-pool analysis is sensitivity. Section AA reports the pairwise ToolRM, prompt, and APIGen audits.

Table 25: SESOI and power audit. The 95% intervals use a one-sample t model for transparent planning; the main policy intervals in Table 4 use the pool bootstrap. Required n targets $|\delta| = .10$ at 80% power.

Family	Policy	n	Mean excess	95% t CI	MDE	Required n	Bonferroni n
D-Scalar	all four	37	$-.0135$	$[-.1381, .1110]$	.1768	112	149
G-Rubric	manifest-order	37	$+.0135$	$[-.1886, .2156]$	.2869	291	388
G-Rubric	random-within-tie	37	$-.0884$	$[-.2459, .0690]$	.2235	177	237
G-Rubric	stable-ID	37	$-.1757$	$[-.3539, .0026]$	.2530	227	303
G-Rubric	abstain-on-tie	6	$-.7167$	$[-1.0446, -.3887]$	.4483	79	106
T-Verifier	all four	37	$+.0135$	$[-.1399, .1669]$	.2178	169	225

Table 26: Selected nested-factorial contrasts. All 33 planned tests share one Holm family.

Family	Contrast	Effect	95% CI	p_{Holm}
D-Scalar	content presence	$-.03297$	$[-.04715, -.01724]$	.00330
D-Scalar	carrier main	$+.02232$	$[+.01783, +.02702]$	.00330
D-Scalar	position main	$-.02548$	$[-.03377, -.01757]$	.00330
D-Scalar	dose main	$-.01002$	$[-.02146, +.00193]$	1.00000
G-Rubric	dose main	$-.01953$	$[-.03906, -.00195]$	.73643
T-Verifier	carrier main	$+.05198$	$[+.03773, +.06656]$	.00330
T-Verifier	empty position	$-.05399$	$[-.07450, -.03236]$	.00330
T-Verifier	content presence × position	$+.04844$	$[+.02230, +.07404]$	.00480
T-Verifier	dose main	$-.02791$	$[-.05988, +.00133]$	1.00000

I CANDIDATE-ROLE PERTURBATIONS

The released compiler manifests retain the base form of each role perturbation. Candidate indices are zero-based and cycle through (`base`, `low`, `high`, `noop`). In HVE, the literal strings also vary with index parity: role `high` uses `[N] please` at odd indices and `[N] please`, with two spaces, at even indices. Because `high` occurs only at indices congruent to 2 (mod 4), the odd-parity high branch is unreachable in the balanced grid. Role `low` alternates between `guess [N]` and `guess [N]`.

Under this fixed cyclic assignment, role-H is an index-bound descriptive diagnostic, not a counterbalanced causal estimate. The ToolSandbox canonicalization below also applies only to the low and noop positions in this cycle.

At odd candidate indices, ToolSandbox substitutes a canonicalized action string before applying the role perturbation. Under the cyclic assignment, these branches correspond to `low` and `noop`; `base` and `high` retain the original form. Canonicalization also reverses argument order in two of four scenarios. For `set_cellular_service_status`, it expands the action from one whitespace word to five. The resulting role factor combines whitespace perturbation with canonical re-spacing.

J KNOWN-ANSWER AND SPACING CONTROLS

Block P uses fresh pools as a known-answer control. Its candidates are matched on episode length and differ in whether they reach the target. Block S is a spacing-matched, five-arm control. Neither block includes candidate permutations, so we report argmax counts under all three tie policies. The permutation policy follows the predeclared candidate-index permutation with seed 20260912 used in the main run. Table 29 reports Block P's preregistered primary estimand, computed per pool as the solved-candidate mean minus the unsolved-candidate mean.

Only G-Rubric is sensitive to the tie policy. D-Scalar has no tied maximum across its 32 pools, and D-Vector has one class-pure tie, so their winner counts do not change across policies. G-Rubric ties in every pool, with 18 top sets containing both groups. The returned candidate is solved in 32 of 32 pools under ascending order, 14 under descending order, and 31 under the seeded permutation; the first count comes from the index-to-label construction described above. Block S is unchanged under the tested policies because no unmodified or prepended arm reaches a pool maximum for either family. This is a property of this score surface, not a general selector invariant.

Table 27: Controlled evaluator-visible budget contrasts. Cells report mean paired score difference, 95% pool-bootstrap interval, and Holm-adjusted p. Native score scales differ across rows.

Family	8K minus 4K	16K minus 8K	16K minus 4K
D-Scalar	$-.07424\ [-.09127, -.05840],\ p = .00045$	$-.02838\ [-.03644, -.02070],\ p = .00045$	$-.10262\ [-.12321, -.08282],\ p = .00045$
G-Rubric	$-.01892\ [-.02568, -.01216],\ p = .00045$	$-.00811\ [-.01622, .00000],\ p = .17449$	$-.02703\ [-.03784, -.01622],\ p = .00045$
T-Verifier	$-.00157\ [-.01699, +.01330],\ p = .84396$	$+.04046\ [+.02228, +.06481],\ p = .00045$	$+.03889\ [+.01783, +.06356],\ p = .00150$

Table 28: Context-regime diagnostic on the same 37 airline pools. The native arm uses 40,960 positions with no RoPE scaling; the 4K/8K/16K rows are nested visible-context budgets under the same native configuration. Score is the mean native score, rank is mean Spearman association, winner is the unique-winner rate, and utility is random-within-top-set success. These rows are diagnostics of the frozen projection protocol, not estimates of a context-architecture effect.

Evaluator	Visible budget	Score	Rank	Winner	Utility
D-Scalar	4,096	.832	.046	.973	.486
D-Scalar	8,192	.758	.122	1.000	.432
D-Scalar	16,384	.730	.139	1.000	.459
G-Rubric	4,096	.680	.029	.135	.385
G-Rubric	8,192	.661	.043	.189	.392
G-Rubric	16,384	.653	.067	.270	.402
T-Verifier	4,096	.670	.138	.973	.514
T-Verifier	8,192	.668	.181	.946	.500
T-Verifier	16,384	.709	.228	.946	.500

K THE COMPLETE PRIMARY GRID

Figure 4 summarizes the complete balanced-grid response cells and their score-scale heterogeneity.

L FRESH-POOL CONFIRMATION BY ACTION-FORM COUNT

Figure 5 reports the fresh-pool confirmation as arm-level counts and contrasts.

M WHAT THE COMPILER INSERTS, AND WHERE

Figure 6 traces the original insertion labels, the orthogonal slot-by-metadata follow-up, and exact-token compilation. For transparency, the realized carrier words were `contextual`, `descriptive`, and `filler`. For $q > 0$, the inserted text is the selected carrier word repeated $q - 4$ times followed by a trailing `0`; the word-repetition counts for $q = 16, 32, 64, 128, 256$ are $12, 28, 60, 124, 252$. The serialized assistant event contributes the remaining three tokens, and the certificates report an exact total token delta of q for each of the D-Scalar, D-Vector, and G-Rubric tokenizers. At $q = 0$, no event or carrier string is inserted.

N WHAT VARIES BETWEEN CANDIDATES, BY DOMAIN

Within-pool candidate variation differs by domain, which limits interpretation of the omnibus domain factor. HVE assigns each candidate a separately seeded action order, producing variation in steps to target and action content. ToolSandbox candidates replay the scenario-specific action sequence and stop on the same terminal action; only action-string formatting varies.

The released base trajectories quantify this contrast. All 2,048 ToolSandbox trajectories have three turns and contain 9 to 18 tokens. The 2,048 HVE trajectories span eleven turn counts from two to twelve and contain 96 to 1,562 tokens. In this construction, ToolSandbox supplies short, repeated action sequences, and evaluators rarely isolate one candidate at $N{=}64$.

Repeated role strings impose an additional mechanical constraint. Each ToolSandbox pool contains 16 copies of four strings, so a deterministic evaluator's top set must contain complete 16-candidate blocks. Across all 5,184 ToolSandbox cells, top-set size is restricted to $\{16, 32, 48, 64\}$. A unique winner is impossible by construction, although D-Scalar and D-Vector distinguish the four forms. Averaged over 54 conditions and 32 pools, ToolSandbox's within-pool score SD is 0.081, 0.150, and 0.125 of the corresponding HVE value for D-Scalar, D-Vector, and G-Rubric. The ToolSandbox scores presented to the selector are correspondingly less dispersed. At $N{=}64$, all three ToolSandbox

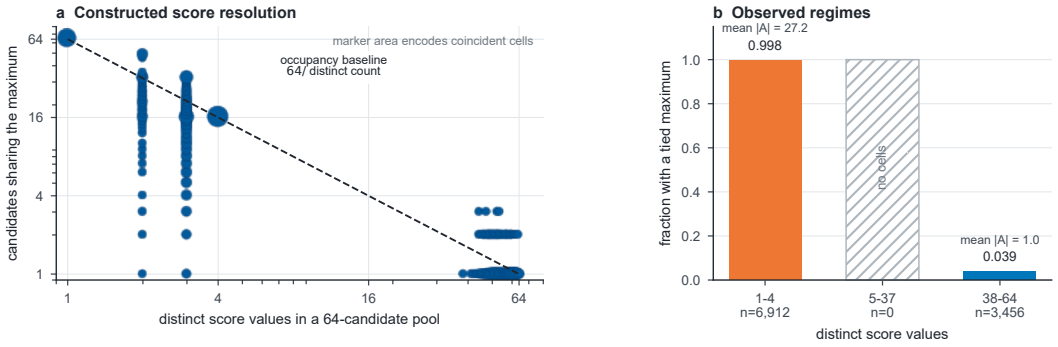

Figure 2: Operator-induced score coarsening pattern. **a**, top-set size versus distinct-value count in 10,368 constructed cells; marker area encodes the number of coincident cells. The dashed curve is the occupancy baseline, computed as pool size divided by distinct count. **b**, the production grid occupies two regimes across six effective checkpoint–domain cells; the middle range contains no observed cells and is not a zero-valued estimate. Rank coarsening fills the gap while mechanically placing about m/k candidates in a top bin. This is an operator-induced pattern conditional on pool size and score distribution, not a general score-resolution law or independent checkpoint evidence.

Table 29: Block P preregistered primary estimand. Intervals are 10,000-draw pool bootstraps under the main analysis convention. The D-Scalar interval contains zero, and no equivalence margin was preregistered. The other two checkpoints separate the groups.

Family	Estimate	95% CI	p	Pools favoring solved
D-Scalar	$+0.0097$	$[-0.0058, +0.0263]$	0.237	15 / 32
D-Vector	$+0.0091$	$[+0.0086, +0.0095]$	$< 10^{-4}$	32 / 32
G-Rubric	$+0.1553$	$[+0.1484, +0.1621]$	$< 10^{-4}$	32 / 32

checkpoint cells tie in every decision, with mean top-set sizes of 16.6, 18.0, and 60.1. The two non-degenerate HVE cells have tie prevalences of 1.4% and 6.4%.

These pool properties alone can produce a family-by-domain interaction. The ToolSandbox selection cells also cannot show that the selector ignored role-H. The supplement reports within-pool spreads of trajectory depth, turn count, and token count by domain; we treat them as properties of the candidate pools.

O Outcome preservation, by factor

The dose, insertion-condition, and carrier factors are certified as outcome-preserving. Each of 221,184 compilation certificates checks equality of action, tool result, termination, completion, and task identity between base and variant. The compiler fails closed on any change to these fields, and every certificate passed. This rule checks more invariants than the manuscript's outcome-preservation claim requires.

Compilation certificates exclude the candidate-role factor because it modifies the action string. For these perturbations, outcome preservation comes from the two domain parsers. HVE extracts a bracketed integer from the action, so leading or trailing words, tabs, and padding preserve the guess. The ToolSandbox subset normalizes whitespace and parses each call into a keyword dictionary. Padding, assignment spacing, and spacing before the closing parenthesis consequently parse identically. Outcomes are preserved whenever those parsers are the only consumers of the action string.

The supplement reports termination and completion by domain and role for all 4,096 base trajectories. Of these, 4,095 terminate successfully and complete. One trajectory, `traj-hve-iclr-balanced-n64-p018-c25-low`, reaches the step cap after 11 actions and terminates `failure/completed` without finding the target. Role assignment does not cause this outcome. HVE draws guess order from `Random(seed*1009 + candidate_index*9176)`, which excludes role. The same candidate index would follow the same action sequence under any role, so the failure belongs to the sampled pool.

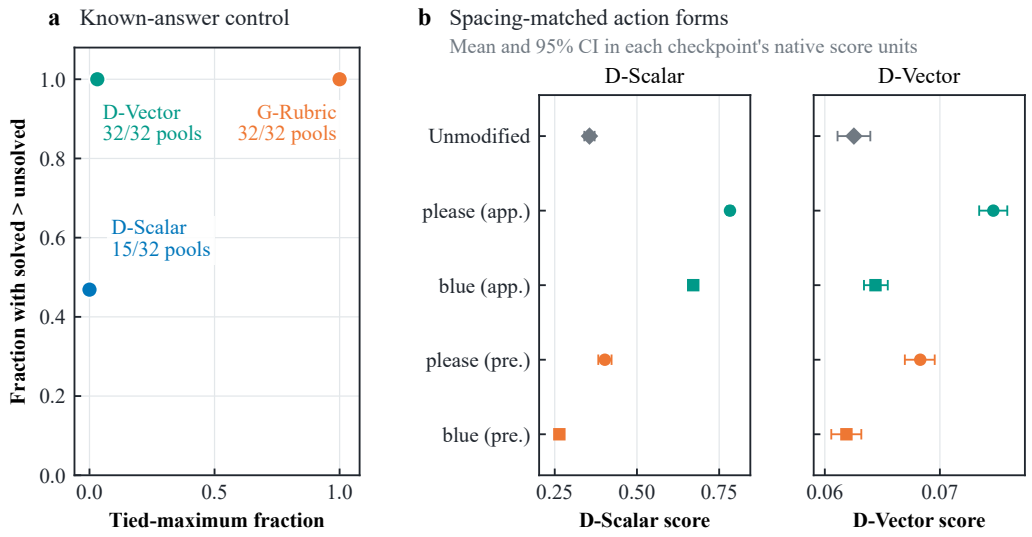

Figure 3: Controls on fresh pools, disjoint from all earlier pools. **a**, tied-maximum prevalence versus the fraction of pools in which solved candidates receive a higher mean score than unsolved candidates. D-Scalar has no tied maximum but tracks the known group difference in only $15/32$ pools; D-Vector and G-Rubric track it in $32/32$ despite opposite tie regimes. Tie prevalence and outcome tracking differ across these checkpoints. **b**, spacing-matched group means with 95% pool-bootstrap intervals. Each checkpoint uses its own native-score axis. Appended forms beat prepended forms for both words and checkpoints after Holm correction. No unmodified or prepended form wins under any policy.

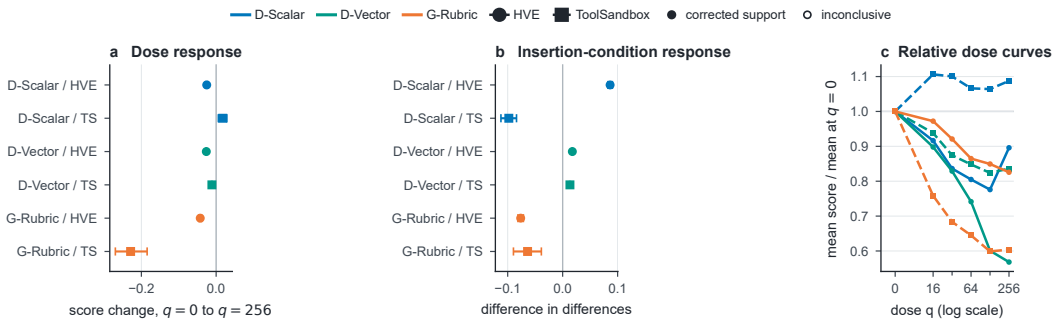

Figure 4: Constructed score responses are heterogeneous. **a**, dose slope; **b**, insertion-condition interaction, which does not identify a pure physical-slot effect because two labels share slot 1. Intervals are 95% pool-bootstrap intervals; filled markers pass the corrected rule. Native units are not comparable across checkpoints. **c**, response relative to each cell's $q{=}0$ mean. These are secondary construction diagnostics, not evidence of a domain-general response pattern.

P Exploratory mechanism probes

These exploratory probes lie outside every multiplicity-corrected family; their records are included with the supplementary evidence. For D-Scalar, sensitivity concentrates at the response tail. Changing only the final action recovers 52% of the all-action effect, compared with 7% for the first action. Terminal-character effects do not follow a simple semantic or length ordering: an appended closing parenthesis shifts the score by $+2.4$ baseline SD, while a period or hyphen has almost no effect. D-Vector is insensitive to the final action in the same probe and responds only to word identity.

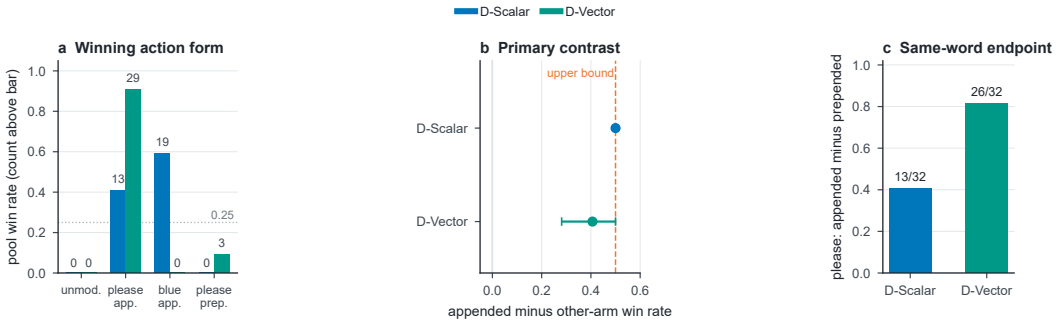

Figure 5: Fresh-pool confirmation reported as counts. **a**, win rate for each action form. The unmodified action never takes the pool argmax for either checkpoint; the prepended form wins in 0 and 3 pools. Only 4 of 64 decisions contain a tie, including one cross-arm tie. **b**, the preregistered primary contrast between appended arms and the other two arms. D-Scalar reaches the upper bound of $+0.5$, yielding a zero-width interval and a p-value at the bootstrap resolution floor. Panel **a** reveals arm-level variation hidden by the saturated mean. **c**, the same-word position contrast at fixed word. This preregistered descriptive endpoint lies outside the multiplicity correction. The checkpoints select different appended words. Position replicates across checkpoints; word identity does not.

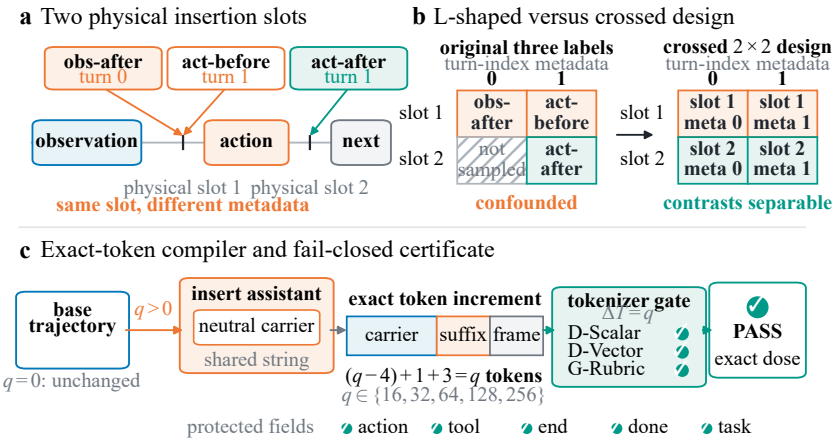

Figure 6: Insertion geometry, orthogonalization, and exact-token compilation. **a**, `observation-after` and `action-before` occupy the same physical gap but have different turn-index metadata; `action-after` occupies the second gap. **b**, the original labels fill three slot-by-metadata cells and confound the two factors. The crossed follow-up fills all four cells so their contrasts can be separated. **c**, for $q > 0$, the compiler searches for one shared carrier whose serialized increment is q under all three evaluator tokenizers. A mismatch or protected-field change fails closed; $q = 0$ inserts no event.

Q Selection sensitivity and claim trace

The machine-readable selection table contains 72 rows from six family–domain cells, four values of N, and three tie policies. Each pool uses one frozen candidate permutation across policies and depths; the requested prefix is exactly `candidate_ids[:N]` for $N \in \{1, 4, 16, 64\}$. Thus $N = 1$ is the head of the frozen permutation, not a newly sampled sub-pool. Primary inference uses permutation order; ascending and descending ID are sensitivity policies. The realized role-H share

at $N = 1$ is 0.2292 in HVE and 0.2500 in ToolSandbox, so the HVE value is not the design share $16/64$. Each estimand cell includes a `claim_state` of supported positive, supported negative, or inconclusive, together with the `decision_rule` that assigned it. The logical-repeat audit treats repeated compilations as nested checks, not additional independent samples.

R AUDIT CHECKLIST

The public-chain audit found no missing row, replay mismatch, abstention, parser error, quarantine, or silent exclusion. A separately implemented internal verifier checked schemas, row counts, ID-set equality, effective sample sizes, Holm decisions, tie-policy sensitivity, logical repeats, and reproducibility. It also passed all 11 fallacy-coverage checks. These checks concern the internal consistency of the measurement chain.

S FRESH-POOL MODIFIER-LOCATION CONFIRMATION

Exploratory word-class and modifier-position probes weakened a social-pragmatics account of the role-H word and localized the effect to modifier position. We then tested that hypothesis on fresh pools after fixing the design, estimands, multiplicity rule, and acceptance gates.

Each of 32 fresh HVE pools contains 64 candidates in four action forms: unmodified, `please` appended, `blue` appended, and `please` prepended. Seeds 7401–7432 are disjoint from all earlier pools. This single-domain experiment does not test ToolSandbox, where the constructed dose and insertion-condition signs reverse. The preregistered primary estimand subtracts the mean win rate of the unmodified and prepended arms from that of the two appended arms. Both families support this contrast after Holm correction. D-Scalar is $+0.5000$ (95% CI $[+0.5000, +0.5000]$), and D-Vector is $+0.4062$ ($[+0.2812, +0.5000]$). Both have $p_{\text{Holm}} = 2.0 \times 10^{-4}$.

The estimand has an upper bound of $+0.5$, which D-Scalar attains exactly. One candidate wins in every pool, always from an appended arm, so each pool contributes $+0.5$. The interval is a point and the p-value reaches the bootstrap resolution floor. Arm-level counts show why: the unmodified action wins in 0 of 32 pools for both families, while the prepended form wins in 0 and 3 pools. Four of 64 decisions contain a tie, including one cross-arm tie. The tie policy does not explain the overall count.

Earlier analyses generated the hypothesis before the fresh-pool test. The fresh test is therefore a prospective replication that was absent from the original preregistration. The exploratory analysis used ascending-ID tie-breaking, whereas balanced-N64 uses permutation order. Ties are rare, but the protocols differ. The checkpoints also favor different appended words. D-Scalar divides win rate between the two words (0.406 and 0.594), while D-Vector concentrates on `please` (0.906 and 0.000). Appended position replicates across checkpoints; word identity does not. The same-word position contrast is 0.406 for D-Scalar and 0.8125 for D-Vector. This preregistered descriptive endpoint lies outside the multiplicity correction.

Appendix P reports D-Scalar's response-tail probe and D-Vector's word-identity response. For D-Vector, a mean-score separation of about $+0.08$ baseline SD between prepended and appended forms reorders almost every pool maximum.

T EVALUATOR INPUT PROTOCOL AND READ-OUT REDUCTION

The three public pointwise checkpoints read the same public trajectory projection, serialized as JSON under one key after a family-specific instruction. D-Scalar receives "Score the quality of the agent trajectory as a response evaluator. Return no text." D-Vector receives "Score the quality of the agent trajectory as a response evaluator." G-Rubric receives "Evaluate the public agent trajectory. Return a final JSON object with a numeric score from 1 to 5 under the key score. Do not include private labels or facts not present in the trajectory." Each call contains one candidate, so the model sees neither pool position nor neighboring candidates. D-Scalar is the Skywork-Reward-V2 checkpoint (Liu et al., 2026), D-Vector is the ArmoRM scalar head (Wang et al., 2024), and G-Rubric is the R3 checkpoint (Anugraha et al., 2025). The manifest records the exact local files and adapters.

D-Scalar maps its raw logit through a sigmoid to $[0, 1]$. D-Vector clips ArmoRM's native scalar head to $[0, 1]$, although selection uses the native head. The common diagnostic schema retains its 19 objective heads as four latent dimensions. G-Rubric parses the final numeric value. Integers from 1 to 5 map to $(\text{score} - 1)/4$, values from 0 to 100 map to $\text{score}/100$, and values in $(0, 1)$ remain

unchanged. This monotonic normalization preserves every tie. The checkpoint emits only part of its 1–5 scale, producing at most three distinct values.

T-Verifier receives "Score whether this tool-use agent trajectory reaches a successful outcome. Return no text." The serializer replaces the task ID with one neutral value and otherwise uses the public JSON representation used in training. The frozen LoRA adapter was trained for three fixed epochs on 140 examples: 90 from nine airline tasks (70 failures and 20 successes) plus 50 positive APIGen conversations. Validation uses 40 examples from four disjoint airline tasks. Epoch-wise validation accuracies are .550, .825, and .800, with losses of .8946, .5767, and .5945. The training loop saves only the terminal epoch and has no early-stopping or best-checkpoint branch. The 37 mixed-outcome evaluation task IDs are disjoint from both airline splits. Exact normalized conversation overlap is zero across supplied splits and between all 5,000 supplied APIGen conversations and the 500 airline conversations. This audit does not cover fuzzy or semantic matches, undisclosed sources, or Qwen3 pretraining. The separate no-APIGen ablation uses 40 training examples and has validation accuracies of .500, .725, and .650; it is a separate checkpoint from the full-context model. The 4,096-token value is the training/inference cap for this local diagnostic, not a native cap for the public checkpoint families. For the natural full-context arm, each Qwen3 config started at 40,960 positions and was extended to 131,072 by setting YaRN `factor` $= 3.2$ and `original_max_position_embeddings` $= 40{,}960$, with cache disabled and MLP sequence chunking. This arm is a context-extension diagnostic, not a native-context claim. The checkpoint manifest records base and adapter version labels, file names, and file sizes for all four pointwise checkpoints.

U D-VECTOR CLIPPING AUDIT

We inspected D-Vector's native scalar head before the documented $[0, 1]$ clipping operation in all three deterministic repeats. Each repeat contains 221,184 rows. All 663,552 values lie strictly within $(0, 1)$, ranging from 0.0065307617 to 0.1181640625; none reaches or crosses either boundary. Hence $\mathrm{clip}(x, 0, 1) = x$ for every row. Clipping leaves distinct-value counts, ties, top-set sizes, winning margins, argmax identities, AUCs, and Spearman correlations unchanged. The supplement records row and boundary counts for each repeat. D-Vector's exact ties predate clipping.

V CROSS-RUN DIRECTIONAL CHECK

The earlier and current runs use disjoint pool seeds and agree in direction for all 18 family–domain–estimand cells. This descriptive count is unweighted, and the same 64 pools underlie all three estimands, so the signs are dependent. At least two cells are indistinguishable from zero in both runs, including G-Rubric/ToolSandbox selection ($+0.0052$ versus $+0.0098$); directional agreement there is uninformative. Comparison depth also differs. The earlier ToolSandbox run used 32 candidates per pool, placing its selection arm at $N = 16$; the current arm uses $N = 64$. We use this result as a consistency check and keep the sample sizes separate.

W RELEASE CONTENTS AND VALIDATION SCOPE

The anonymous supplement contains aggregate and pool-level inputs, all-50 and mixed-outcome records, context projections, held-out reruns, ToolRM choices, factorial designs, prompt/composition sensitivities, and the GPU closure. The non-GPU audit adds reconstructed original-airline cycle edges, pair-support summaries, a 1,650-row AgentSuite task–generator crosswalk, retained-input hashes, and a rerun script. The sensitivity package adds the frozen feasible-pool labels, graph/length/support manifests, and completed edge-score receipts. Tables 9–10 map each block to its artifact. The release excludes private or sealed artifacts, model weights, adapters, logits, caches, credentials, the raw AgentSuite snapshot, candidates outside the frozen pools, and raw constructed-production streams.

The independent unit is always the pool or task. Each constructed domain and factorial arm has $n = 32$; the mixed-outcome natural contrasts have $n = 37$; the descriptive all-task extension has $n = 50$; and the legacy short-pool primary split has $n = 2$. The GPU closure separately contributes 16 original-airline, 38 AgentSuite-airline, and 15 AgentSuite-retail feasible mixed pools. Bootstrap and permutation seeds, draw counts, correction rules, tie policies, and denominators are specified in the relevant methods text and each package's completion records. A separately implemented internal full-context verifier re-derived the aggregate rows and multiplicity decisions. The remote focused

test suite passed 19/19 tests. Score rows for the constructed grid decompose as $(4{,}096$ baseline $+$ $4{,}096$ replay $+$ $221{,}184$ variants$) \times 3$ repeats $\times$ 3 families $= 2{,}064{,}384$. The package reports row counts for all three deterministic re-executions, which are nested checks and do not constitute independent samples.

We release the selection layer in two reduced forms. `selection_pool_summary.csv` contains 2,304 rows, one per family, pool, depth, and tie policy, with role-H rate and mean tie count. `selection_decisions_r0.csv.gz` contains 373,248 rows, one per decision, with condition, permutation, depth, tie policy, role-H indicator, and tie count. The summary reproduces all 72 released selection-curve rows and the $N{=}64$ tie counts to floating-point precision. All three repeats produce identical selection decisions, so the archive stores one repeat.

The release includes standard-library logical gates. They fail closed when required paths are missing, completion fields are incorrect, row counts or ID sets drift, files are empty, structured files are invalid, or schema parsing fails. R2, R4, and R5 add direct row-level equality against independent scoring reruns and deterministic analysis reruns. Their validators check schema, denominator, ID-set, and row equality; they do not use cryptographic or Git-identity checks. Rebuilding the omitted raw stream still requires the public checkpoints and compute. Author and repository metadata remain withheld for double-blind review.

X SELECTION-ENDPOINT VISUALIZATION

Figure 7 plots the secondary endpoint documented in Appendix Q.

Y KNOWN-ANSWER CONTROLS

On fresh, length-matched known-answer pools, tied-maximum fractions were 0.000, 0.031, and 1.000 for D-Scalar, D-Vector, and G-Rubric. Ties were not forced in these controls, so the fractions describe the tested checkpoint–pool combinations. D-Scalar identifies a unique winner in every pool but selects a solved candidate in only 15 of 32 pools. G-Rubric never identifies one member, although its top sets are 72–97% solved across pools. The index-to-label audit found that indices 0–31 are solved and 32–63 are unsolved; 18 pools contain mixed-class ties. Ascending-ID therefore returns 32/32 solved candidates by construction, not through a label-independent selector property. Descending-ID and seeded policies return 14/32 (.4375) and 31/32 (.96875), while random-within-tie utility is .93743. The apparent spread $1.000 - .4375 = .5625$ is an order-alignment diagnostic. Scores leave members within a top set unordered; executable labels determine its composition and the utility of each deployment policy.

A second control matches spacing and expands arm coverage for the modifier-location confirmation in Appendix S. The original arms had unmatched spacing and no prepended counterpart for the second word. Fresh pools contain five single-spaced arms. After Holm correction, appended forms beat prepended forms for both words and evaluators. Adjusted p-values are 2×10^{-4} for `please` and 2×10^{-4} and 2×10^{-3} for `blue`. No unmodified or prepended form wins under any tie policy. The modifier-location effect persists; the word-identity effect does not replicate.

Z SCORE-RESOLUTION INTERVENTION

Across $10{,}368$ pool-by-condition cells, top-set size tracks pool size divided by the number of distinct score values ($r = -0.974$ in log–log; Figure 2). These 10,368 cells are dependent: the distinct-value count is nearly constant within each of six checkpoint–domain cells, and the construction leaves five to thirty-seven distinct values unobserved. With fewer than four distinct values, 99.8% of cells tie. At thirty-eight or more values, 3.9% tie. G-Rubric/ToolSandbox contributes 1,503 of the 1,506 all-equal cells.

We fill the empty band by binning the same released ranks into at most $k \in \{2, \ldots, 64\}$ levels while preserving raw ties. Across 228,096 derived rows, the resulting slope is 1.05 and $r = 0.96$. These values largely follow from the operator, which places roughly m/k of m ranks in the top bin. Applying it to uniform noise gives slope 1.08 and $r = 0.99$. At this pool size, the constructed transition lies between twenty and thirty-two distinct values. The threshold is conditional on pool size and the candidate-score distribution and has not been estimated for other checkpoints.

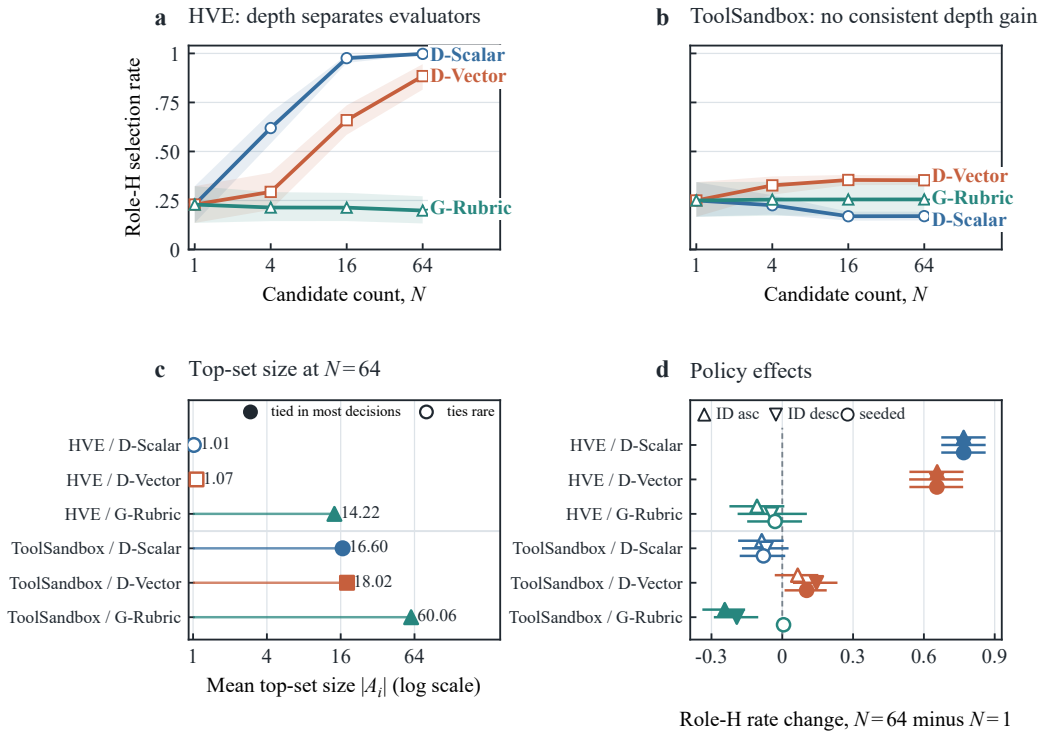

Figure 7: Selection depth, score resolution, and policy sensitivity. **a–b**, descriptive role-H selection rate and 95% pool-bootstrap interval versus candidate count under the seeded-permutation policy, shown separately for HVE and ToolSandbox. Curve-end labels identify evaluator families. **c**, mean top-set size $|A_i|$ at $N = 64$ on a log axis; the baseline is a unique maximum, so filled markers denote checkpoints whose maximum is shared in nearly every decision and hollow markers those where ties are rare. **d**, change in role-H rate from $N = 1$ to $N = 64$ under three named tie policies; filled markers denote 95% intervals excluding zero. These are secondary mechanism diagnostics because role is index-bound and ToolSandbox has no unique winner under this construction.

AA BLOCK T: REWARD DECOMPOSITION, STRATA AND SENSITIVITY

The airline reward combines a database check with a communication check, and some tasks require no database write. We stratify binary reward by write requirement and treat database and communication recovery as secondary quantities outside the primary endpoints. This block measures whether selection recovers the trajectory judged successful by the environment. General transactional correctness, intent, deception, and hidden-utility harm lie outside its scope.

The 37 pools contain 185 successes among 370 candidates, with per-pool shares from 0.1 to 0.9. Table 30 reports all four selector estimands for each deployable policy in the matched C4K arm. By definition, random-within-tie utility equals top-set quality. Deterministic policies can differ when identifiers align with reference labels. Every pool contains a success, so regret equals $1 - U$. G-Rubric identifies one winner in only 0.135 of C4K pools, and its top sets are 0.402 successful. Its manifest-order utility is 0.622, compared with 0.402 under random-within-tie selection.

Truncation is associated with outcome in the matched arm: $91/143 = 63.6\%$ of untruncated candidates succeed, compared with $94/227 = 41.4\%$ of truncated candidates; the difference is .222 (95% pool-bootstrap CI $[.076, .358]$). This imbalance makes an untruncated-only subset non-comparable. The released audit lists all 1,110 rows with checkpoint, pool, candidate, original and retained event counts, omitted events, token lengths and executable outcome. All terminal events are retained. This is an unadjusted association analysis. The full-context arm contains all 37 pools for the three eligible checkpoints. D-Vector is excluded from this paired arm because of its native context cap. Table 31 reports the matched observation-policy sensitivity.

Table 30: Matched Block T C4K policy audit. Pool success is uniform expected utility; top-set quality equals random-within-tie utility U_{rnd}. U_{manifest} is utility under the deterministic manifest-order policy used in the paired comparison; R_{rnd} is random-policy regret. Intervals and per-pool records are released.

Family	Pool success	Top-set quality	U_{manifest}	U_{rnd}	R_{rnd}
D-Scalar	.500	.541	.541	.541	.459
G-Rubric	.500	.402	.622	.402	.598
T-Verifier	.500	.405	.405	.405	.595

Table 31: Full-context minus matched C4K differences on all 37 paired pools. The first success@1 column uses manifest-order tie resolution; the adjacent column uses random-within-tie selection. Winner change is reported as a rate. Brackets are shared pool-bootstrap 95% intervals; Holm correction is over the three eligible checkpoints within each metric, and every adjusted test is inconclusive. Scores use native within-checkpoint units, precluding comparisons across rows.

Family (n)	mean score	winner changed	tied	success@1 (manifest)	success@1 (random)	Spearman	AUC
D-Scalar (37)	$-.014[-.132, +.105]$	.405[.000, .811]	.000[.000, .000]	$-.054[-.189, +.081]$	$-.054[-.189, +.081]$	$+.091[-.004, +.193]$	$+.072[-.008, +.160]$
G-Rubric (37)	$+.014[-.178, +.203]$	.270[.081, .459]	$-.027[-.135, +.081]$	$-.108[-.243, +.000]$	$+.010[-.058, +.083]$	$+.021[-.029, +.070]$	$+.008[-.028, +.042]$
T-Verifier (37)	$+.014[-.132, +.159]$	.243[.000, .486]	.000[.000, .000]	$+.108[-.027, +.243]$	$+.108[-.027, +.243]$	$+.053[-.035, +.140]$	$+.049[-.020, +.120]$

For G-Rubric, the apparent full-minus-C4K utility contrast changes from $-.108$ under manifest order to $+.010$ under random-within-tie selection. Manifest order is associated with generator scaffold, so the former contrast cannot be attributed to context projection alone. Neither policy-specific interval excludes zero after the registered multiplicity correction.

E1: tool-specialized pairwise endpoint. We evaluate Li et al.'s ToolRM-Gen (Li et al., 2025), released as `RioLee/ToolRM-Gen-Qwen3-4B-Thinking-2507`. It differs from Agarwal et al.'s ToolRM outcome model (Agarwal et al., 2025). ToolRM-Gen scores ordered response pairs instead of assigning independent scalar rewards. Each pool requires all $\binom{10}{2} = 45$ unordered comparisons, and each candidate appears in nine. We parsed all $1{,}665$ choices and converted them to within-pool win counts. We report this endpoint separately because its integer 0 to 9 scale and induced ties are not directly comparable with Table 4's pointwise scores.

Table 32: E1 ToolRM-Gen under two deterministic tie policies and swapped-order sensitivity. "Tied" is the fraction of 37 pools with a shared maximum. Brackets give 95% pool-bootstrap intervals for *success@1*. Order-consistent rows retain the $1{,}299$ pairs that choose the same candidate in both presentations.

Pair set	Tie policy	Tied	success@1 [95% CI]
Single order	Lexicographic ID	0.081	$0.5135\ [0.351, 0.676]$
Single order	Manifest order	0.081	$0.5676\ [0.405, 0.730]$
Order-consistent	Lexicographic ID	0.297	$0.4324\ [0.270, 0.595]$
Order-consistent	Manifest order	0.297	$0.4865\ [0.324, 0.649]$

The single-order ranking has pool-mean Spearman $+0.129\ [-0.066, 0.322]$ and AUC $0.596\ [0.457, 0.726]$ under either policy. A tie policy changes only the selected member of a shared maximum. Three pools tie; only pools 12 and 24 return different winners under the two conventions. Both *success@1* intervals contain 0.5, the AUC interval contains 0.5, and the Spearman interval contains zero. Under the preregistered E1 rule, this endpoint shows no detectable separation.

The position control scores all $1{,}665$ pairs in both orders. Slot 1 wins $1{,}621$ of $3{,}330$ judgments, a rate of $0.4868\ [0.4697, 0.5039]$. The preferred candidate changes in 366 pairs, giving a flip rate of $0.2198\ [0.2000, 0.2402]$. With greedy decoding, these flips reflect presentation order, not sampling variation. Aggregate slot balance coexists with pairwise order dependence. Removing unstable pairs creates 11 tied pools, and both order-consistent *success@1* intervals in Table 32 still contain 0.5.

E2: balanced APIGen ablation. The ablation keeps T-Verifier's split seed, Qwen3-8B base, learning rate, batch size, 4,096-token cap, and the same three-epoch training schedule. It removes every APIGen positive. To satisfy the training script's class-balance gate, it also subsamples 70 held-out τ^2-bench failures to 20. The resulting training set contains 20 successes and 20 failures, while

validation retains a separate balanced set of 40 examples. No APIGen example remains. Validation accuracy is 0.500, 0.725, and 0.650 across epochs 0–2. Loss falls to 0.686 at epoch 1 and rises to 0.754 at epoch 2. Both validation accuracy and loss deteriorate from epoch 1 to epoch 2.

The matched C4K rows in Table 30 belong to the newer five-enhancement arm and use its paired score files and terminal-preserving projection. The E2 and E3 rows below are independent archived audits using legacy Block T score files and the legacy projection/tie protocol. They are not replicate estimates from one common scoring arm, so their baseline values should not be compared as such.

Table 33: E2 and the legacy 37-pool E3 prompt audit on the same Block T pools. Success@1 uses manifest-order tie resolution; brackets give 95% pool-bootstrap intervals, while Spearman and AUC are pool means. E3 rescored all three prompt conditions in one matched pass; the released split-audit package reports the preregistered short-pool E3 extension.

Audit	Condition	success@1 [95% CI]	Spearman	AUC
E2	T-Verifier baseline	0.4595 $[0.297, 0.622]$	$+0.0432$	0.5322
E2	No-APIGen balanced	0.4324 $[0.270, 0.595]$	$+0.0098$	0.5106
E3	Original instruction	0.4324 $[0.270, 0.595]$	$+0.0197$	0.5061
E3	Minimal instruction	0.3243 $[0.189, 0.486]$	-0.0316	0.4740
E3	Detail instruction	0.4865 $[0.324, 0.649]$	$+0.0250$	0.5172

On paired pools, the E2 ablation-minus-baseline difference is -0.0270 $[-0.0811, 0.0000]$ for *success@1*. Differences are -0.0333 $[-0.1249, 0.0552]$ for Spearman and -0.0216 $[-0.0859, 0.0385]$ for AUC. Each point estimate decreases slightly, and every interval includes zero. At the resolution of 37 pools, primary-endpoint change is undetectable and equivalence remains unestablished.

E3: instruction sensitivity. The original instruction is "Score the quality of the agent trajectory as a response evaluator. Return no text." The minimal instruction is "Return a score." A detailed instruction rewards complete, detailed, well-explained responses and penalizes incorrect or unnecessary tool use. We held the checkpoint, shared projection, C4K budget, and candidate order constant and rescored all prompt conditions together. The three conditions select successful winners in 16, 12, and 18 of 37 pools. Because the conditions share pools, independent interval overlap is not a valid contrast. Cochran's omnibus gives $Q(2) = 6.222$, $p = 0.0446$. Table 34 uses exact McNemar/binomial tests for success@1 and shared resampling for rank metrics.

Table 34: E3 paired prompt contrasts (second minus first). Each endpoint's three contrasts form one Holm family. Success@1 uses an exact McNemar/binomial p; Spearman and AUC use paired sign-flip p-values. Intervals use the same pool-bootstrap indices across contrasts.

Endpoint	Contrast	Difference	95% CI	p_{Holm}
success@1	minimal − original	$-.108$	$[-.243, .000]$	.438
success@1	detail − original	$+.054$	$[-.081, +.189]$	.688
success@1	detail − minimal	$+.162$	$[+.054, +.297]$	.094
Spearman	minimal − original	$-.051$	$[-.092, -.014]$	.0268
Spearman	detail − original	$+.005$	$[-.034, +.045]$	.798
Spearman	detail − minimal	$+.057$	$[+.017, +.097]$	.0268
AUC	minimal − original	$-.032$	$[-.060, -.006]$	.0437
AUC	detail − original	$+.011$	$[-.019, +.043]$	.486
AUC	detail − minimal	$+.043$	$[+.015, +.072]$	.0153

Cochran's omnibus indicates variation in *success@1* across prompt conditions, although no pairwise binary contrast survives Holm correction. The minimal prompt also changes the ranking summaries. The legacy 37-pool audit had no preregistered equivalence margin, so its non-significant contrasts remain inconclusive. The short-pool E3 extension follows its preregistered SESOI and held-out small-n guard. We release winner IDs, success labels, and all 370 candidate-level score deltas. The original-prompt rescore changes one of 37 winners relative to the cached D-Scalar row, and the matched comparison excludes all cached rows.

Pool strata. Each pool crosses five generators (`gpt-5`, `claude-sonnet-4-5`, `gemini-2.5-pro`, `Kimi-K2`, and `Qwen3-Coder`) with two dataset versions. Among

185 `original` candidates, 115 succeed; among 185 `code-generation` candidates, 70 succeed. D-Scalar, G-Rubric, and T-Verifier select the lower-base-rate version in 26, 30, and 22 of 37 pools, compared with 12 for D-Vector. These counts describe an association with scaffold version; a causal version effect remains unidentified. For `original`/`code-generation`, per-version *success@1* is 0.649/0.270 for D-Scalar and 0.595/0.324 for D-Vector. The corresponding values are 0.514/0.243 for G-Rubric and 0.676/0.405 for T-Verifier.

The full-context arm covers all 37 pools for three eligible checkpoints; D-Vector appears only in the short-pool C4K arm. Matched C4K often changes winners, but its corrected contrasts remain inconclusive and measure a change in observation policy. The controlled 4K/8K/16K arm detects seven score contrasts and none of 45 selection contrasts. Legacy C4K pointwise comparisons use manifest-order tie resolution; the full-context P0 audit reports four named policies. E1 adds a specialized pairwise checkpoint, E2 changes the retained τ^2-bench failures while removing external positives, and E3 changes one checkpoint's instruction. The two-pool held-out split is a guarded ceiling case. Each sensitivity is interpreted only for the checkpoint and projection it tests.

AB Omnibus heterogeneity tests

All three omnibus family-by-domain heterogeneity tests pass Holm correction. Dose slope gives Wald $\chi^2_2 = 394.4$ and $p_{\text{Holm}} = 4.6 \times 10^{-86}$. The insertion-condition interaction gives $\chi^2_2 = 397.0$ and 1.9×10^{-86}; role-H concentration gives $\chi^2_2 = 285.4$ and 1.0×10^{-62}. The insertion-condition statistic leaves physical slot and turn-index metadata conflated. These tests use the baseline standardization described in Appendix AC, whose constants span more than one order of magnitude. Native score scales are incomparable, and the omnibus statistic depends on scaling. Scale dependence restricts these tests to heterogeneity diagnosis and precludes cross-checkpoint ordering of effect magnitudes.

AC Additional methods and interpretation

Related work. reWordBench applies meaning-preserving transformations to reward-model inputs and finds sharp degradation, sometimes below random, after small wording changes (Wu et al., 2025). Judge and reward-shaping studies further document sensitivity to presentation and evaluation context (Shi et al., 2025; Fein et al., 2026). Reward models can favor spurious attributes, including length and format (Shen et al., 2023; Huang et al., 2024; Wang et al., 2026a; Zhang et al., 2024). The preference objective can also create content-independent distance bias (Xie et al., 2026). Other studies address length bias through causal interventions or explicit fitting (Kim et al., 2026; Zhao et al., 2026b; Bu et al., 2025). Concurrent evidence indicates that process-reward-model scores can change under label-preserving transformations (Shihab et al., 2026).

Intervention certificates. Each certificate records the realized string, tokenizer-specific deltas, and semantic invariants. The compiler rejects any compilation with truncation, token spill-over, or a change to state, action, tool result, termination, completion, or task identity. All 221,184 certificates passed. At dose q=0, no string is inserted; carrier labels map to one compilation, producing 24,576 duplicate variants.

Inference. Confidence intervals use 10,000 paired, domain-stratified pool-bootstrap draws with NumPy PCG64 seed 20260913. Families within a domain share resampling indices, while the primary permutation policy uses seed 20260912. Holm correction covers six family–domain cells per estimand and three estimands in the omnibus tests at $\alpha = 0.05$. Two-sided bootstrap p-values use a plus-one correction and a 1×10^{-4} floor. For the 13 of 18 primary cells that reach this floor, the intervals determine the available inferential resolution; the floor itself carries no information about a larger effect. Between-pool standard deviations of mean baseline score are 0.1431, 0.0131, and 0.3382. A directional result requires an interval excluding zero and rejection by the corrected test. Alternative tie policies remain sensitivity analyses.

Known-answer controls. D-Scalar identifies a unique winner in all 32 pools but favors the solved candidate in only 15. This family also has the largest HVE selection effect, which fails to certify recovery of a known difference. G-Rubric's mean scores separate the groups in every pool, while its maximum contains both groups in 18 pools. In those 18 pools, the tie policy determines the returned candidate. Appendix J reports tie-policy counts, the fixed-layout analysis, and the second control.

Interpretation. We report every implementable tie policy because one cell spans a policy-dependent range of 0.249. Baseline standardization also matters because native score units are incomparable. These checks concern argmax selection from a filled candidate pool under an implementable tie policy; top-k, rerank-and-keep-k, and group-relative consumers are outside their scope. D-Vector's mean score differs by only about $+0.08$ baseline SD between prepended and appended forms, yet this shift moves almost every pool argmax to an appended candidate. A mean-only probe would miss this nearly deterministic selection pattern.

AD ETHICS STATEMENT

The study uses public benchmark-style trajectories without human subjects or personal data. Neutral probes provide no instructions for bypassing real services. Protected reference-outcome artifacts remain outside the public audit and statistical workflow. Every release must preserve upstream licenses and notices.