What Does a Failed Rollout Supervise? Separating Policy Failure from Recoverability in Tool Agents

Anonymous authors
Paper under double-blind review

Abstract

When a tool-using agent fails to complete a task, does that mean its earlier actions left no way to succeed? A failed continuation may reflect the limitations of the policy rather than an unrecoverable state. This distinction matters when sampled outcomes are used to label intermediate actions for training. We develop an executable audit that separates policy-relative success from recoverability within a specified continuation family and budget. Successful continuations justify positive recoverability labels, complete bounded non-reachability certificates justify negative recoverability labels, and unresolved cases remain unlabeled. Exhaustive replay over finite action inventories under fixed continuation bounds supplies exact audit labels in small deterministic environments. Our analysis distinguishes population identifiability from finite-sample certification and connects target mismatch to probability loss and representation-dependent ranking. Audits of released supervision data and six target builders show why sampled failure and native negative labels cannot uniformly be interpreted as irrecoverability. In a three-seed, four-arm Retail intervention with restricted candidates and matched training budgets, exact relabeling reduces held-out recoverability log loss by 0.150 relative to forced one-rollout labeling, while certificate-based abstention reduces it by 0.050 relative to matched deletion. Task-held-out monotone calibration of all arms, using exact labels from other tasks, narrows these gaps to 0.024 and 0.011 while preserving within-task rankings. Effects on action ranking and terminal success remain inconclusive. These findings motivate target-explicit supervision and separate evaluation of probability fit, action ranking, and terminal task success.

1 Introduction

When an agent samples K continuations from a partial tool trajectory and all fail, should the preceding action receive a negative training label? The answer depends on the supervision target. Monte Carlo hard estimation takes the maximum sampled outcome (Xie et al., 2026), while AgentPro labels a step negative when all sampled search paths fail (Deng et al., 2025). All-failure outcomes describe policy behavior under a sampling budget, rather than establish irrecoverability.

This distinction matters for decisions. Ranking actions under a fixed policy and budget can use their success probabilities; declaring a branch unsolvable requires excluding every successful continuation in the declared family. For a Retail task requiring returns of orders A and B, returning A leaves a successful suffix—return B—even if every sampled continuation fails (Figure 1(a)).

These two signals also coexist in released process-supervision data. The expert-trajectory release accompanying Iterative Step-level Process Refinement (IPR) (Xiong et al., 2024) covers ALFWorld, InterCodeSQL, and WebShop. Our audit finds that 11,566 of the 41,560 step records with stored expert terminal reward one have a stored Monte Carlo step value of zero. The annotations answer different questions: the expert trajectory's terminal outcome records task completion, whereas the Monte Carlo field summarizes sampled continuation outcomes. Their coexistence motivates an explicit account of what failure-derived labels supervise.

We develop an executable audit of recoverability within a declared continuation family and budget. A verified successful continuation supports a positive label; a complete bounded non-reachability

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.

certificate supports a negative; cases with neither remain unlabeled (Figure 1(b)). Exhaustive replay over finite action inventories supplies exact bounded targets in small deterministic environments. Our analysis distinguishes population identifiability from finite-sample certification and connects target mismatch to probability loss and representation-dependent ranking. Auditing six released target builders then tests how their native negative labels relate to this recoverability target.

To test the learning consequences of these target distinctions, we conduct a three-seed, four-arm Retail study with shared candidate menus and matched training budgets. Fixed-row relabeling and certificate-based sample admission improve raw recoverability log loss against their respective controls. Rank-preserving calibration substantially narrows both gaps without changing action order (Figure 1(c)). This finding makes the distinction between probability fit and action selection central to evaluating process supervision.

Our contributions are as follows:

1. **Targets and theory.** We distinguish policy-relative success from bounded recoverability, characterize population identification and finite-sample certification, and derive consequences of target mismatch for probability loss and representation-dependent ranking.
2. **Executable audits.** We implement witness/certificate/abstain labeling and combine exact bounded replay, released-data analysis, and six native target builders to characterize how negative supervision relates to recoverability across different target definitions.
3. **Learning consequences.** A matched Retail study quantifies the raw probability-fit gains of exact relabeling and certificate-based sample admission; rank-preserving calibration shows that much of the loss separation can change without changing action order.

2 Related Work

Process-supervision targets. Step verifiers assess local validity and guide search (Lightman et al., 2024). Tool-agent supervision includes rollout value, progress, preference, and call quality (Xiong et al., 2024; Choudhury, 2025; Xi et al., 2025; Li et al., 2026a;c). Xie et al. (2026) address policy-dependent noise in Monte Carlo labels through reflection-aware correction and confidence-based training, targeting intrinsic step correctness. Deng et al. (2025) assign negative step labels when all three Monte Carlo tree search simulations fail. Other approaches use stage-aware values, hindsight relabeling, turn-level supervision, or verified rescue suffixes (Sa et al., 2026; Xu et al., 2026; Zhou et al., 2026; Xia et al., 2026). We study bounded environmental recoverability: whether a successful continuation remains within a specified family and budget. This differs from intrinsic step correctness, since a successful repair can establish recoverability without validating the original step.

Multiple valid tool trajectories. Stateful benchmarks evaluate environment states rather than agreement with a single reference trace (Yao et al., 2024; Lu et al., 2024; Trivedi et al., 2024). Set-valued evaluators represent valid alternatives through evidence banks, plans, merged traces, action graphs, or multiple executions (Kim et al., 2026; Wei et al., 2026; Sahoo et al., 2026; Qian et al., 2025; Huang et al., 2025). These approaches recognize that different trajectories can satisfy the same task. We complement these approaches with exact replay that checks whether a candidate action admits a successful continuation within the declared family and budget.

Credit assignment, recovery, and certification. Recent credit-assignment methods refine action-level feedback (Tao et al., 2026; Lu et al., 2026; Ren et al., 2026), while counterfactual approaches analyze failure attribution and repair (Li et al., 2026b; Bonagiri et al., 2026; Shah, 2026; Pona et al., 2025). ERR measures recovery regret (Vuddanti & Chittiprolu, 2026), and DART certifies rollback admissibility (Yang et al., 2026). Conformal risk control provides a different form of guarantee, using calibration data to bound expected loss (Angelopoulos et al., 2024). Our focus is per-instance evidence for bounded recoverability: a successful continuation establishes possibility, whereas a negative certificate excludes every successful continuation in the declared family. We use this distinction to audit supervision labels and test the learning consequences of changing their target.

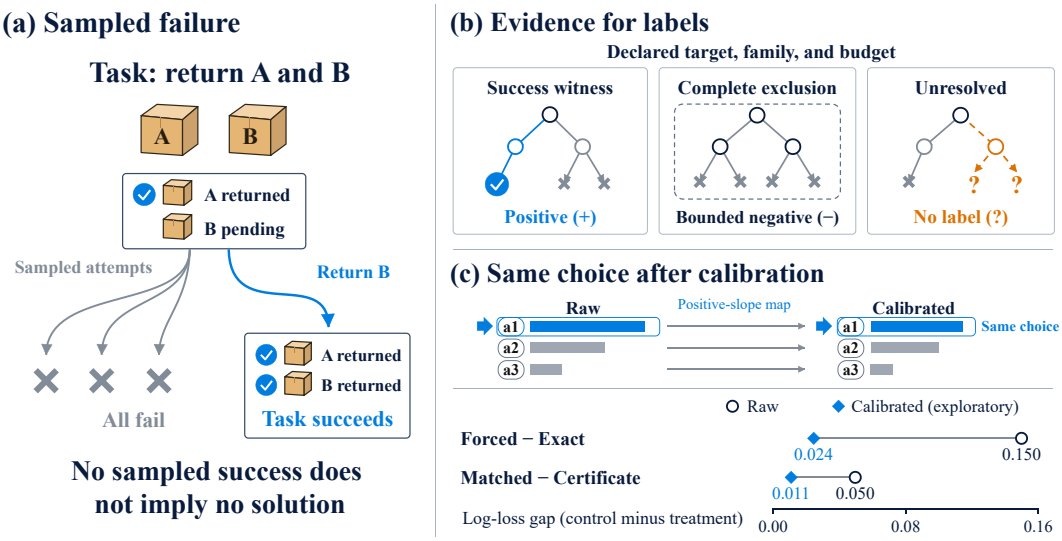

Figure 1: **From Rollout Failure to Supervision.** **(a)** A recoverable action can be followed by sampled failures. **(b)** Witnesses, bounded non-reachability certificates, and unresolved cases yield positive, negative, and no recoverability labels. **(c)** Schematic bars illustrate ranking preservation under monotone calibration; numerical results show smaller control-minus-treatment log-loss gaps after task-held-out calibration using other tasks' exact labels.

3 Recoverability Targets and Executable Certification

Our audit asks whether a candidate action leaves a successful continuation within a declared family and budget. It assigns a positive label when execution supplies a success witness, a negative label when complete bounded analysis rules out success, and no recoverability label when neither form of evidence is available.

3.1 Defining the Supervision Target

For task τ, let $s \in \mathcal{S}_\tau$ be the complete executor state, $a \in \mathcal{A}_\tau(s)$ a candidate tool action, and G_τ the goal predicate. The deterministic executor $F_\tau(s, a)$ returns a next state and tool output, or $\perp$ on execution failure. The audit contract specifies the goal, admissible continuations, resource bound, and treatment of execution errors. These choices determine what a label means.

A reference action $a_r(s)$ defines the point-matching label

$$y_{\text{point}}(s, a) = \mathrm{I}[a = a_r(s)], \tag{1}$$

where I denotes the indicator. Recoverability instead asks whether the goal remains reachable after a. Let $\Sigma^{\text{env}}(s, a)$ contain all admissible environmental continuations and let $\Sigma_B(s, a) \subseteq \Sigma^{\text{env}}(s, a)$ be a declared finite family satisfying resource bound B. For either family $\mathcal{F}$, define

$$R_\exists(s, a; \mathcal{F}) = \mathrm{I}\left[\exists \sigma \in \mathcal{F}(s, a) : G_\tau\left(F^{\text{st}}_\tau(s, (a, \sigma))\right) = 1\right]. \tag{2}$$

Here F^{st}_τ is the state reached by executing a and the continuation under the declared error contract. A family may contain fixed sequences or observation-contingent continuation trees. We write $R^{\text{env}}_\exists$ for the environmental target and $R^B_\exists$ for the declared-family target. Since $R^B_\exists \le R^{\text{env}}_\exists$, a bounded success witness establishes environmental recoverability, whereas a bounded negative excludes success only within the declared family.

A continuation policy π under per-rollout bound B defines a different target:

$$p_{\pi,B}(s, a) = \Pr_{\sigma \sim \pi_B(\cdot|s,a)}\left[G_\tau\left(F^{\text{st}}_\tau(s, (a, \sigma))\right) = 1\right]. \tag{3}$$

The distinction is between whether a successful continuation exists and how likely the policy is to find one. Policy success supports ranking attempts under that policy; non-reachability evidence supports excluding a branch within the declared family.

3.2 What failed rollouts establish

Let $Z_i \in \{0,1\}$ indicate success on attempt i. The soft label $\widehat{p}_K = K^{-1}\sum_{i=1}^{K} Z_i$ estimates single-attempt success when all attempts share marginal $p_{\pi,B}$. The hard label $H_K = \max_{i\le K} Z_i$ instead estimates group success: $\mathrm{E}[H_K \mid s,a] = 1 - P_K$, where $P_K = \Pr(\bigcap_{i=1}^{K}\{Z_i = 0\} \mid s,a)$ is the joint miss probability. Under conditionally i.i.d. attempts, the probability that a recoverable instance receives an all-failure label is

$$\mathrm{FIR}_K(s,a;\pi,B) = (1 - p_{\pi,B}(s,a))^K. \tag{4}$$

Under this sampling contract, the soft and hard labels target $p_{\pi,B}$ and $1-(1-p_{\pi,B})^K$, respectively, rather than the existential target $R^B_\exists$.

Shared randomness, reused search trees, and adaptive retries require the joint quantity P_K. For example, if attempts share a latent success probability Q and are i.i.d. conditional on Q, then $P_K = \mathrm{E}[(1-Q)^K] \ge (1-\mathrm{E}[Q])^K$ for $K \ge 2$. Dependence can therefore increase the miss probability relative to the independent-attempt calculation. For dependent attempts, we measure group-level misses directly.

Proposition 1 (support and finite-sample certification limits). Fix π_B. Suppose an environment class contains an irrecoverable M_0 with an unsupported, nonmerged, goal-distinguishing history, and permits an M_1 that makes this history successful while preserving every policy-supported transition, observation, and goal value. Without a revealing side channel, M_0 and M_1 induce identical rollout distributions but different $R^{\mathrm{env}}_\exists(s,a)$. Policy rollouts therefore do not identify global recoverability on this class.

For a nonempty finite family with exact full support, $\mathrm{supp}(\pi_B) = \Sigma_B(s,a)$, we instead have $p_{\pi,B} = 0$ if and only if $R^B_\exists = 0$. The population rollout distribution identifies bounded recoverability. Under conditionally i.i.d. attempts, a success is eventually observed almost surely whenever $p_{\pi,B} > 0$. Nevertheless, for every finite $K \ge 1$, an all-failure sample is possible both for an unrecoverable instance and for a recoverable instance with $0 < p_{\pi,B} < 1$. Over a recoverable class with $\inf p_{\pi,B} = 0$, the supremum finite-K miss probability is one.

Rollout budget. Under conditionally i.i.d. attempts, suppose the protocol separately establishes $R^B_\exists(s,a) = 1 \Rightarrow p_{\pi,B}(s,a) \ge p_{\min}$, where $0 < p_{\min} < 1$. For $0 < \delta < 1$, the all-failure probability on every declared-family-recoverable instance is at most δ when

$$K \ge \left\lceil \frac{\log \delta}{\log(1 - p_{\min})} \right\rceil. \tag{5}$$

Sample means, confidence bounds, and posteriors characterize policy success $p_{\pi,B}$. Our audit labels declared-family recoverability $R^B_\exists$ using executable success witnesses and bounded non-reachability certificates, leaving other cases unresolved.

3.3 Constructing executable witnesses and certificates

For the finite-inventory audit, let $J = \{1,\dots,k\}$ index discovered write-action occurrences, each usable at most once, and let $U \subseteq J$ contain the consumed indices. We represent a search node as $q = (U, C_\tau(s))$, where the canonicalizer C_τ must preserve every state distinction affecting future execution or goal evaluation. Including U distinguishes equal database states with different remaining actions. The Retail representation uses the complete database state together with consumed occurrences; its future sufficiency is an assumption about the executor.

We instantiate Eq. 2 on the finite inventory. At a fixed node $q = (U, C_\tau(s))$, a candidate $j \in J \setminus U$ leaves the suffix inventory $S_j = J \setminus (U \cup \{j\})$. Its continuation family consists of all ordered sequences of distinct occurrences from S_j, including the empty sequence, with a suffix-call budget $B = |S_j|$. The candidate itself consumes one additional occurrence. For this database-target audit, success means reaching the canonical target key $k^\star$. Under the *stop* contract, execution errors terminate at a non-target outcome, with $C_\tau(\bot) \ne k^\star$. A remaining occurrence j is therefore recoverable exactly when

$$j \in J \setminus U, \quad F_\tau(s,a_j) \ne \bot, \quad \exists \mathbf{c} \in \mathrm{Seq}(J \setminus (U \cup \{j\})) : C_\tau\big(F^{\mathrm{st}}_\tau(s, a((j,\mathbf{c})))\big) = k^\star. \tag{6}$$

Here $a((j, \mathbf{c}))$ denotes the candidate call followed by the calls indexed by $\mathbf{c}$. Reaching the target ends execution successfully, even when unused occurrences remain.

The audit enumerates every admissible unused occurrence and its successor, reusing representative states with the same product key while retaining concrete transitions. It then propagates viability backward from target nodes, following standard finite-state reachability (Clarke et al., 1986). A viable candidate receives a positive label and an executable successful suffix. Complete enumeration supplies a negative certificate for a nonviable candidate. An incomplete search without a verified witness leaves the candidate unlabeled. Successful no-ops consume their occurrences, so every transition reduces the number of remaining calls.

Proposition 2 (bounded witness and negative-certificate validity). Assume deterministic replay, future-sufficient canonicalization, early target termination, and complete enumeration of all admissible unused occurrences and their successors, with execution errors represented as terminal non-target outcomes. Every backward-viable edge then has an executable target-reaching witness, and every edge outside that set has no target-reaching sequence of remaining occurrences. Both claims follow by induction on the number of unused occurrences. The certificate applies to $R^{\mathrm{stop}}_{\exists,J}$, where each declared occurrence is used at most once.

The *continue* contract allows recovery after a state-preserving tool error by mapping $(U, C_\tau(s))$ to $(U \cup \{j\}, C_\tau(s))$. Provided such errors actually preserve the executor state, the same induction certifies $R^{\mathrm{continue}}_{\exists,J}$: errors also consume an occurrence. Our four-arm labels use continue, whereas terminal policy execution uses stop. The sensitivity analysis changes the label graph from continue to stop without retraining. Enumeration costs $O(k!k)$ without prefix reuse; the primary census uses inventories with $k \leq 5$ as an exact measurement instrument.

Grounded symbolic extension. The ALFWorld audit replaces the occurrence inventory with grounded operators. Fresh official replay verifies positive witnesses. Negative certification requires complete bounded search of a transition superset and an interface refinement under which official executions and successes are represented in the symbolic system. Incomplete search without a replay-verified witness yields ABSTAIN; restoration, mapping, or replay failures also prevent label admission. Appendix A.4 gives the construction and soundness conditions.

3.4 Consequences for learned scores

The audit determines which labels are justified; learning determines how well a scorer predicts those labels. Let X be the scorer's representation, $R = R^B_\exists \in \{0, 1\}$ the recoverability target, and $H = H_K$ the hard rollout label. Their conditional targets are $q_R(x) = \Pr(R = 1 \mid X = x)$ and $q_H(x) = \Pr(H = 1 \mid X = x)$. Under conditionally i.i.d. attempts, $q_H(x) = \mathrm{E}[1 - (1 - p_{\pi,B})^K \mid X = x]$; the definition of q_H also applies to dependent attempts.

Proposition 3 (proper-loss target mismatch). For any strictly proper binary loss, population-risk minimization over unrestricted predictors yields the unique Bayes predictors q_R and q_H for labels R and H, respectively, up to almost-sure equality. Under log loss, evaluating the rollout-target predictor against recoverability incurs excess risk

$$\mathcal{E}_R(q_H) = \mathrm{E}_X[\mathrm{KL}(\mathrm{Ber}(q_R(X)) \parallel \mathrm{Ber}(q_H(X)))] \,. \tag{7}$$

The excess is positive whenever the conditional targets differ on a set of positive measure. If $q_R(x_1) > q_R(x_2)$ but $q_H(x_1) < q_H(x_2)$, the Bayes-optimal rollout score also reverses the recoverability ordering. The proof is given in Appendix A.3.

The ranking implication depends on the representation. With the complete deterministic instance as X and rollouts confined to the declared family, an unrecoverable action has $q_H = 0$ and cannot outrank a recoverable action with positive policy success. Recoverable actions with zero policy success probability can tie at zero. For family-confined rollouts, strict reversals therefore require pooling distinct instances in X.

To distinguish the effects of label content and sample selection, the Retail study uses fixed-row relabeling and matched-exposure admission comparisons. We evaluate the trained scorers in terms of recoverability probability fit, action ranking, and terminal task success.

4 EVALUATION DESIGN

The evaluation connects supervision targets to their learning consequences. We first compare sampled failures with exact bounded recoverability, then examine what released target builders emit on the audited decisions, and finally change supervision in a controlled Retail learning study.

4.1 BOUNDED RECOVERABILITY AND ROLLOUT EVIDENCE

The Retail census uses deterministic τ-bench tasks with two to five indexed mutating calls and the official final database as the target (Yao et al., 2024). Complete enumeration over 38 frozen test tasks yields 306 recoverable edges and 33 bounded negatives. The action family permits each occurrence once and uses the stop contract. AgentDojo supplies two task-semantic replications: Workspace contains 465 positives and 13 structural negatives, and Slack contains 681 positives and 18 negatives (Debenedetti et al., 2024).

On the Retail census, we compare uniform ordering, a successful-witness control, and two frozen schema-relational continuation policies, Qwen3-32B and Qwen3-Next-80B. Each learned policy makes 16 independent, freshly reset attempts per positive edge. Its inputs exclude task instructions, literal argument values, targets, graph structure, and witness suffixes. For $K \in \{1, 2, 4, 8, 16\}$, we report the fraction of recoverable decisions lacking a sampled success, the recoverable fraction among forced negative labels, and the coverage of witness/certificate/abstain labeling. Forced and three-way labeling use identical rollout evidence and differ in their admission of negative labels.

4.2 RELEASED RECORDS AND NATIVE TARGET BUILDERS

The released-data audit compares stored expert terminal rewards with aggregate Monte Carlo values in Iterative step-level Process Refinement (IPR) (Xiong et al., 2024). Its denominator consists of records with expert terminal reward 1, and uncertainty is clustered by released ID.

The builder audit uses a fixed 113-decision panel containing the 49 recoverable failed decisions from the preliminary stress-test slice and all 64 bounded negatives from the three censuses: 56 Retail, 22 Workspace, and 35 Slack cases. Six released builders meet the frozen eligibility rules: AgentPRM Monte Carlo and InversePRM demonstration labels (Choudhury, 2025), IPR Monte Carlo (Xiong et al., 2024), ToolPRMBench pair construction (Li et al., 2026a), ToolRM pairwise targets (Li et al., 2026c), and AgentPRM shaped targets (Xi et al., 2025). Each receives its native failed-rollout, demonstration, pair, or step-judge input under a fixed implementation or checkpoint. Recoverability labels and successful witnesses are joined only after the native outputs are fixed. For each builder, we report the number of valid outputs, the native-negative rate, and the recoverable fraction among emitted negatives (N/A when no negatives are emitted).

4.3 CONTROLLED FOUR-ARM LEARNING INTERVENTION

The learning study uses the official 74/40 train/test split of τ^2 Retail (Barres et al., 2025). Its finite candidate menus contain all official mutating reference calls plus at most two deterministically ordered train-only donor calls. Reference calls retain a solution path, while donor calls introduce alternatives on a shared menu. Native execution and backward reachability produce 2,757 training rows and 1,099 test rows; the test rows span 36 candidate-bearing tasks. This design evaluates restricted-candidate reranking.

Here $K = 1$ means one frozen pseudorandom ordering of the remaining menu occurrences after the candidate action. The ordering is generated from the candidate identifier and shared across training seeds. Both this rollout and the exact training target use the continue contract. The four supervision arms are: *Forced*, which labels each row by its single rollout outcome; *Exact*, which retains every row and replaces its label with bounded recoverability; *Certificate/abstain*, which admits positives witnessed by the fixed $K = 1$ rollout and negatives supported by bounded non-reachability certificates, while omitting $K = 1$ failures without negative certificates; and *Matched deletion*, which uses the deletion quotas induced by Certificate/abstain within each task–source stratum and selects rows for removal only among $K = 1$ negatives, without consulting their exact recoverability.

All arms use Skywork-Reward-V2-Qwen3-4B with its scalar head and final two transformer blocks trainable. They share the candidate universe, optimizer settings, and initial trainable state within each of three paired seeds. Every run receives 2,757 example exposures and 345 optimizer updates; the two shorter arms cycle their 2,540 retained rows. Certificate/abstain and Matched deletion also match assigned positive and negative exposures within each task–source–$K = 1$ group. Exact–Forced therefore changes labels on fixed rows, whereas Certificate–Matched compares admission rules with matched deletion counts and label exposures. In the latter comparison, retained candidate identities and exact-label composition can differ.

4.4 Outcomes, uncertainty, and follow-up analyses

Recoverability log loss measures probability fit by averaging binary cross-entropy over candidate rows within each task and then equally across tasks. Top-1 measures whether the highest-scored candidate is recoverable, averaging states within tasks and then tasks; states without a recoverable candidate contribute zero. Both metrics cover 36 test tasks, while pairwise accuracy covers 35 tasks with eligible comparisons. Terminal evaluation covers all 40 official test tasks: the policy greedily selects the highest-scored unused action and stops on success, tool error, or menu exhaustion. A 10,000-draw paired bootstrap resamples official tasks after averaging the three seeds within each task.

To examine probability scaling independently of within-task ranking, we calibrate each arm and seed using leave-one-task-out fits on the other 35 tasks. Bias-only calibration changes the logit intercept; positive-slope logistic calibration fits $\widetilde{p} = \sigma(a \operatorname{logit}(p) + b)$ with $a > 0$. All arms use identical folds and exact calibration labels.

5 Results

5.1 Successful expert records can have zero sampled value

Our audit of the complete IPR release covers 43,897 records. Among the 41,560 with stored expert terminal reward 1, 11,566 (27.83%) have aggregate Monte Carlo value zero. The corresponding fractions are 33.27% in ALFWorld, 11.05% in InterCodeSQL, and 15.26% in WebShop (Table 6). The expert reward records observed trajectory completion, whereas the Monte Carlo value summarizes sampled continuation outcomes at an intermediate step. Zero sampled value therefore coexists with expert success across all three environments. This cross-environment pattern motivates our target-explicit audit: interpreting failure-derived supervision requires distinguishing policy-relative sampled outcomes from executable evidence of recoverability.

5.2 Finite rollout budgets can miss recoverability witnesses

We distinguish per-attempt success (successful attempts divided by all attempts, including invalid outputs), positive-edge miss rate (positive edges without a witness after K attempts divided by all audited positive edges), and negative-pool contamination (recoverable cases labeled negative divided by all emitted negatives).

The exact Retail census reveals why aggregate rollout success can hide recoverable failures. Of 306 positive edges, 283 succeed under every valid ordering of the remaining indexed calls. The other 23, spread across 11 tasks, each have uniform success mass $1/2$. Under independent uniform sampling with replacement, the all-failure probability for each of these edges is $(1/2)^K$: 50%, 25%, 6.25%, 0.3906%, and 0.00153% for $K = 1, 2, 4, 8, 16$. The frozen uniform panel records zero misses at $K = 8$ and 16. The witness control succeeds immediately, while all 33 bounded negatives remain unsuccessful.

Qwen3-32B and Qwen3-Next-80B achieve success rates of 97.04% and 97.10%, respectively, over 4,896 attempts per policy. On the 23 fallible edges, the rates are 62.50% and 61.41%. Figure 2(a) compares the exact uniform expectation with observed learned-policy miss rates.

The AgentDojo replications show how such misses enter a training-label pool. At $K = 1$, the recoverable fraction among forced negatives is 13.33–23.53% in Workspace and 28.00–57.14% in Slack across the two policies. A small positive-edge miss rate can produce substantial negative-pool contamination when certified negatives are sparse in the audited population. Restricting negative

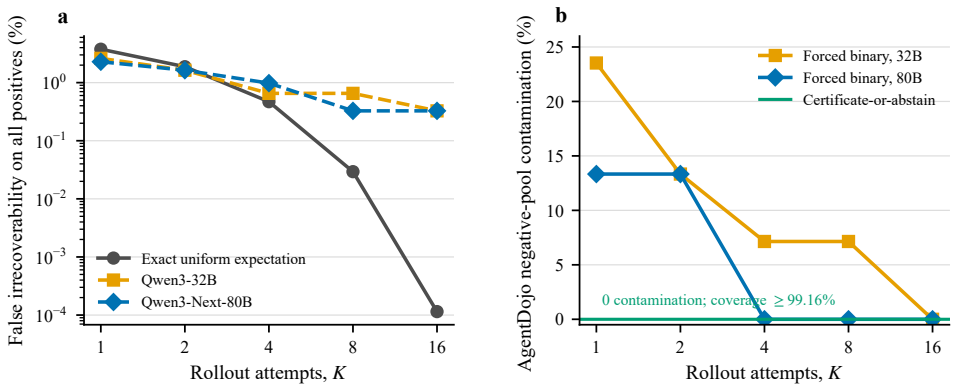

Figure 2: **Sampling budget and label quality. (a)** Positive-edge miss rates over 306 Retail positives. **(b)** Negative-pool contamination in Workspace under forced labeling and certificate/abstain (465 positives and 13 bounded negatives).

admission to bounded certificates gives zero contamination by construction on these audited populations, while retaining 96.57–99.58% label coverage. Figure 2(b) shows the Workspace comparison; Tables 4 and 5 give the domain-level counts.

5.3 Native negative labels have different meanings

On a fixed panel of 49 recoverable failed decisions and 64 bounded negatives, all six builders produce valid outputs using their native input formats. Native-negative rates span 0–100% (Table 1). Four rules emit negatives on all decisions, including the 49 recoverable cases. ToolRM emits 48 negatives, ten of which are recoverable; the shaped AgentPRM target emits none. These outputs reflect different native input–target contracts on the same case mix.

Table 1: Native labels from six builders on 113 audited decisions: 49 recoverable failed decisions and 64 bounded negatives. Native-negative rates use all 113 decisions as the denominator; recoverable fractions condition on emitted negatives.

Builder	Native target	Native negative(%)	Recoverable|negative (%)
AgentPRM Monte Carlo	policy success	100.00	43.36
InversePRM demonstration	demonstration identity	100.00	43.36
IPR MC	policy success	100.00	43.36
ToolPRMBench pair construction	pair construction	100.00	43.36
ToolRM released pairwise	pair preference	42.48	20.83
AgentPRM shaped target	shaped progress	0.00	N/A

For example, AgentPRM's Monte Carlo target is explicitly policy-relative (Choudhury, 2025); a failed sampled continuation can receive a negative outcome even when another continuation succeeds. Demonstration identity, pair preference, and shaped progress answer other questions again.

5.4 Target correction improves raw probability fit

In the controlled three-seed Retail study, both supervision interventions improve recoverability probability fit against their specified controls. Exact relabeling reduces log loss by 0.150 relative to Forced $K = 1$, while Certificate/abstain reduces it by 0.050 relative to Matched deletion.

Exact–Forced compares supervision targets on the same 2,757 training rows. Certificate–Matched compares sample-retention rules under matched assigned-label exposures, with different retained candidates and recoverability composition (Appendix Tables 9 and 10).

Top-1 ranges from 15.54% to 15.91% across arms, and terminal success ranges from 61.67% to 65.00%; the paired intervals for the reported decision contrasts include zero. Top-1 counts all-negative states as failures, giving a uniform-selection baseline of 9.97% and an exact-label oracle ceiling of 21.41% on this population (Appendix Table 12).

Table 2: Primary Retail contrasts with paired 95% task-bootstrap confidence intervals in brackets, conditional on three realized seeds and fixed rollout labels.

Contrast	Log loss ↓	Top-1 (pp)	Terminal (pp)
Exact relabel − Forced	-0.150 [-0.330,-0.026]	+0.37 [-0.91,+1.49]	+3.33 [-5.00,+11.67]
Certificate − Matched drop	-0.050 [-0.082,-0.030]	-0.07 [-0.51,+0.34]	+0.83 [-2.50,+5.00]

5.5 Rank-preserving calibration narrows the loss gaps

Table 3 compares raw log loss with leave-one-task-out calibration across 36 tasks. For each arm and seed, the calibrator is fitted using the same exact labels from the other 35 tasks, preserving rankings within the held-out task.

Table 3: Recoverability log loss before and after task-held-out, rank-preserving calibration, averaged over 36 tasks and three seeds.

Arm	Raw	Bias-only	Positive-slope logistic
Forced $K = 1$	0.4813	0.3410	0.3326
Certificate/abstain	0.4328	0.3289	0.3202
Exact relabel	0.3314	0.3175	0.3081
Matched drop	0.4823	0.3394	0.3310

Positive-slope calibration narrows the Forced–Exact log-loss gap from 0.1499 to 0.0245 and the Matched–Certificate gap from 0.0495 to 0.0108. Bias-only calibration yields similar gaps of 0.0235 and 0.0106. Exact retains the lowest point-estimate loss under both calibrations. All four positive-slope calibrated predictors also have lower loss than the common task-held-out constant baseline of 0.4217, fitted using the same exact-label tasks (Appendix E). The comparison shows that substantial changes in log-loss separation can occur while each model's within-task action ordering remains fixed. Probability fit and action selection therefore provide distinct evidence about the effects of supervision.

5.6 Additional sensitivity analyses

The four completed rollout-budget runs compare Exact with hard $K = 1, 4, 8$; hard $K = 4, 8$ attain lower recoverability log loss than Exact. In a separate experiment using a different candidate population and a frozen 0.6B backbone, the five $K = 1$ sampled-label heads also attain lower loss than the exact-label head. These comparisons show that a valid recoverability target need not yield the lowest test loss in every training setting, separating target validity from the performance of a model trained on that target. Full results and experimental protocols are provided in Appendix F.

6 Limitations and Conclusion

The certificates apply to deterministic environments and declared action families, with the ALFWorld diagnostic additionally conditioned on transition-refinement assumptions. The Retail experiments use reference-containing candidate menus, and calibration holds out tasks within a previously examined task collection. The tool-error sensitivity analysis rescores existing runs. Future work can extend the audit to generated actions, partial observability, and stochastic dynamics, and evaluate transfer to new task collections and native action spaces.

We turn the distinction between policy failure and bounded recoverability into an executable supervision rule based on witnesses, certificates, and abstention. Released-data and native-builder audits demonstrate how different supervision targets produce different interpretations of failure. The controlled Retail intervention measures the learning consequences of changing those targets, while rank-preserving calibration shows why probability fit and action selection require separate evaluation. Together, these results provide a concrete framework for defining, auditing, and evaluating failure-derived supervision.

Reproducibility statement

The paper and appendices document the environments, targets, data splits, prompts, seed rules, training settings, and aggregation procedures. They also provide theorem assumptions, proofs, and supporting analyses. The coverage of saved-graph checks and executor replay is reported separately.

References

Anastasios N. Angelopoulos, Stephen Bates, Adam Fisch, Lihua Lei, and Tal Schuster. Conformal risk control. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2208.02814.

Victor Barres, Honghua Dong, Soham Ray, Xujie Si, and Karthik Narasimhan. τ^2-bench: Evaluating conversational agents in a dual-control environment. *arXiv preprint arXiv:2506.07982*, 2025. URL https://arxiv.org/abs/2506.07982.

Akash Bonagiri, Devang Borkar, Gerard Janno Anderias, Setareh Rafatirad, and Houman Homayoun. Causalflow: Causal attribution and counterfactual repair for llm agent failures, 2026. URL https://arxiv.org/abs/2605.25338.

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

Edmund M. Clarke, E. Allen Emerson, and A. Prasad Sistla. Automatic verification of finite-state concurrent systems using temporal logic specifications. *ACM Transactions on Programming Languages and Systems*, 8(2):244–263, 1986. doi: 10.1145/5397.5399.

Edoardo Debenedetti, Jie Zhang, Mislav Balunović, Luca Beurer-Kellner, Marc Fischer, and Florian Tramèr. AgentDojo: A dynamic environment to evaluate prompt injection attacks and defenses for LLM agents. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/97091a5177d8dc64b1da8bf3e1f6fb54-Abstract-Datasets_and_Benchmarks_Track.html.

Yuchen Deng, Shichen Fan, Naibo Wang, Xinkui Zhao, and See-Kiong Ng. AgentPro: Enhancing LLM agents with automated process supervision. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 9981–10006. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.506. URL https://aclanthology.org/2025.emnlp-main.506/.

Wei-Jin Huang, Yuan-Ming Li, Zhi-Wei Xia, Yu-Ming Tang, Kun-Yu Lin, Jian-Fang Hu, and Wei-Shi Zheng. Modeling multiple normal action representations for error detection in procedural tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 27794–27804, 2025.

Wonjoong Kim, Sangwu Park, Yeonjun In, Sein Kim, Dongha Lee, and Chanyoung Park. Beyond the final answer: Evaluating the reasoning trajectories of tool-augmented agents. In *Proceedings of the International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2510.02837.

Dawei Li, Yuguang Yao, Zhen Tan, Huan Liu, and Ruocheng Guo. Toolprmbench: Evaluating and advancing process reward models for tool-using agents. In *Findings of the Association for Computational Linguistics: ACL 2026*, 2026a. doi: 10.18653/v1/2026.findings-acl.602. URL https://aclanthology.org/2026.findings-acl.602/.

Mukai Li, Qingcheng Zeng, Tianqing Fang, Zhenwen Liang, Linfeng Song, Qi Liu, Haitao Mi, and Dong Yu. Verified critical step optimization for llm agents, 2026b. URL https://arxiv.org/abs/2602.03412.

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. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 8613–8640, 2026c. URL https://aclanthology.org/2026.findings-acl.419/.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2305.20050.

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, 2024. URL https://arxiv.org/abs/2408.04682.

Yijun Lu, Rui Ye, Jiajun Wang, Yuwen Du, Tian Jin, Songhua Liu, and Siheng Chen. ABSeeker: Training long-horizon search agents via answer-backtracked credit assignment. arXiv preprint arXiv:2608.05102, 2026. URL https://arxiv.org/abs/2608.05102.

Edoardo Pona, Milad Kazemi, Yali Du, David Watson, and Nicola Paoletti. Abstract counterfactuals for language model agents, 2025. URL https://arxiv.org/abs/2506.02946.

Yaoyao Qian, Yuanli Wang, Jinda Zhang, Yun Zong, Meixu Chen, Hanhan Zhou, Jindan Huang, Yifan Zeng, Xinyu Hu, Chan Hee Song, and Danqing Zhang. WebGraphEval: Multi-turn trajectory evaluation for web agents using graph representation. *arXiv preprint arXiv:2510.19205*, 2025.

Bo Ren, Yirong Mao, Yi Yang, and Wenhui Que. IAPO: Influence-aware policy optimization for credit assignment in multi-turn service agents. arXiv preprint arXiv:2608.24588, 2026. URL https://arxiv.org/abs/2608.24588.

Inkyu Sa, Konstantin Stulov, and Rajat Bhageria. ValueFormer: A causal transformer value function with stage-aware labels for semi-autonomous vision-language-action policies. *arXiv preprint arXiv:2608.02958*, 2026. URL https://arxiv.org/abs/2608.02958.

Priyam Sahoo, Gaurav Mittal, Xiaomin Li, Shengjie Ma, Benjamin Steenhoek, Pingping Lin, and Yu Hu. Agentlens: Revealing the lucky pass problem in swe-agent evaluation, 2026. URL https://arxiv.org/abs/2605.12925.

Jaineet Shah. Causal agent replay: Counterfactual attribution for llm-agent failures, 2026. URL https://arxiv.org/abs/2606.08275.

Leitian Tao, Baolin Peng, Wenlin Yao, Tao Ge, Hao Cheng, Mike Hang Wang, Jianfeng Gao, and Sharon Li. TRACE: Turn-level reward assignment via credit estimation for long-horizon agents. arXiv preprint arXiv:2607.13988, 2026. URL https://arxiv.org/abs/2607.13988.

Harsh Trivedi, Tushar Khot, Mareike Hartmann, Ruskin Manku, Vinty Dong, Edward Li, Shashank Gupta, Ashish Sabharwal, and Niranjan Balasubramanian. AppWorld: A controllable world of apps and people for benchmarking interactive coding agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pp. 16022–16076, 2024. doi: 10.18653/v1/2024.acl-long.850. URL https://aclanthology.org/2024.acl-long.850/.

Sri Vatsa Vuddanti and Satwik Kumar Chittiprolu. Recoverability has a law: The ERR measure for tool-augmented agents. *arXiv preprint arXiv:2601.22352*, 2026.

Rongzhe Wei, Ge Shi, Min Cheng, Na Zhang, Pan Li, Sarthak Ghosh, Vaibhav Gorde, and Leman Akoglu. Long-horizon plan execution in large tool spaces through entropy-guided branching, 2026. URL https://arxiv.org/abs/2604.12126.

Zhiheng Xi, Chenyang Liao, Guanyu Li, Yajie Yang, Wenxiang Chen, Zhihao Zhang, Binghai Wang, Senjie Jin, Yuhao Zhou, Jian Guan, Wei Wu, Tao Ji, Tao Gui, Qi Zhang, and Xuanjing Huang. Agentprm: Process reward models for llm agents via step-wise promise and progress, 2025. URL https://arxiv.org/abs/2511.08325.

Xu Xia, Jinghua Piao, Min Yang, Xiaochong Lan, Jiaju Chen, and Yong Li. From scoring to acting: Outcome-verified comparative self-distillation for LLM agents. *arXiv preprint arXiv:2607.27937*, 2026. URL https://arxiv.org/abs/2607.27937.

Bin Xie, Bingbing Xu, Xueyun Tian, Yilin Chen, and Huawei Shen. Towards robust process reward modeling via noise-aware learning. *arXiv preprint arXiv:2601.12748*, 2026.

Weimin Xiong, Yifan Song, Xiutian Zhao, Wenhao Wu, Xun Wang, Ke Wang, Cheng Li, Wei Peng, and Sujian Li. Watch every step! llm agent learning via iterative step-level process refinement, 2024. URL https://arxiv.org/abs/2406.11176.

Iris Xu, Sunshine Jiang, John Marangola, Nitish Dashora, Richard Li, Thomas Liu, Zexue He, Yuheng Zhi, Alex Pentland, Pulkit Agrawal, and Zhang-Wei Hong. Learning more from less: Reinforcement learning from hindsight. *arXiv preprint arXiv:2607.09042*, 2026. URL https://arxiv.org/abs/2607.09042.

Ke Yang, Panpan Li, Zonghan Wu, Kejin Xu, Huaxi Huang, and Xiaoshui Huang. Dart: Semantic recoverability for structured tool agents, 2026. URL https://arxiv.org/abs/2605.23311.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A benchmark for tool-agent-user interaction in real-world domains, 2024. URL https://arxiv.org/abs/2406.12045.

Yuhang Zhou, Kai Zheng, Haoling Li, Dengyun Peng, Can Xu, and Jingjing Chen. TurnOPD: Making on-policy distillation turn-aware for efficient long-horizon agent training. *arXiv preprint arXiv:2607.05804*, 2026. URL https://arxiv.org/abs/2607.05804.

A Theoretical Results and Proofs

A.1 Support and finite-sample certification

We prove Proposition 1 for a fixed state–action pair (s, a) and continuation policy π_B. Suppose the environment class contains an irrecoverable environment M_0 and an unsupported continuation $\sigma^\star$ reaching a history that is not merged with any policy-supported history. Suppose it also contains M_1, which makes that history successful while preserving all policy-supported transitions, observations, and goal values. The construction requires both environments to exist, including the irrecoverable M_0. Without an environment-revealing side channel, coupling the policy choices gives identical observed traces in both environments. Their rollout distributions are therefore identical for any number of observations, although $(R^{\mathrm{env}}_{\exists})^{M_0}(s, a) = 0$ and $(R^{\mathrm{env}}_{\exists})^{M_1}(s, a) = 1$. Global recoverability is thus not identifiable from these rollout distributions.

For the finite-family statement, assume $\mathrm{supp}(\pi_B) = \Sigma_B(s, a)$ is nonempty and finite. Write its members as σ_j, their probabilities as $w_j > 0$, and their success indicators as $g_j \in \{0, 1\}$. Then $p = \sum_j w_j g_j$ is zero exactly when every family member fails, giving $p = 0 \iff R^B_{\exists} = 0$. Conditionally i.i.d. attempts give $P_K = (1 - p)^K$. This probability equals one at $p = 0$, zero at $p = 1$ for $K \geq 1$, and remains positive for finite K when $0 < p < 1$. The population distribution therefore identifies bounded recoverability, whereas a finite all-failure sample can arise from either an unrecoverable instance or a recoverable instance with $0 < p < 1$. When $p > 0$, an infinite i.i.d. sequence eventually observes success almost surely.

For a class $\mathcal{M}$ of recoverable instances with $\inf_{M \in \mathcal{M}} p_M = 0$,

$$\sup_{M \in \mathcal{M}} \Pr_M(\text{all } K \text{ fail}) = \sup_{M \in \mathcal{M}} (1 - p_M)^K = 1.$$

A lower bound on success mass is a separate protocol premise. For $0 < p_{\min} < 1$ and $0 < \delta < 1$, if the protocol establishes $p_M \geq p_{\min}$ for every recoverable instance, then $(1 - p_M)^K \leq (1 - p_{\min})^K$. Solving this bound for a miss probability at most δ gives Equation 5. Any smaller integer fails uniformly over the Bernoulli family containing the endpoint $p = p_{\min}$, although a restricted family with larger success mass may require fewer attempts. These statements concern class-conditional witness misses, rather than the posterior probability that an all-failure instance is unrecoverable. For dependent attempts, the joint miss probability P_K replaces the product law.

A.2 Finite-inventory certificate soundness

For Proposition 2, fix the deterministic executor F_τ, a future-sufficient canonicalizer C_τ, an indexed action inventory J, and the target key $k^\star$. Product nodes are $q = (U, C_\tau(s))$, where U records

consumed occurrences. Assume every admissible occurrence $j \in J \setminus U$ is executed from a recorded full-state representative and its deterministic successor is retained. Equality under C_τ must preserve action applicability, future execution, and goal evaluation. Under the stop contract, errors lead to terminal non-target outcomes. Reaching the target ends a successful suffix even when unused occurrences remain.

Define V recursively: every target node belongs to V, and a non-target node belongs to V if and only if it has an enumerated non-error successor in V. We prove by induction on $r = |J \setminus U|$ that $q \in V$ exactly when some sequence of remaining occurrences, including the empty sequence, reaches $k^\star$. A target node has the empty witness for every r. A non-target node with $r = 0$ has neither a successful suffix nor a viable outgoing edge. For $r > 0$, a viable first edge reaches a node in V with one fewer unused occurrence, so induction supplies a successful suffix from that node. Conversely, any successful suffix from a non-target node is nonempty and begins with a non-error edge to a node with a successful suffix. By induction, that successor belongs to V. Complete enumeration includes the first edge, so the original node also belongs to V.

Conditioning on a candidate's first execution gives the edge statement: backward-viable edges have executable success witnesses, and other edges have no successful sequence of remaining occurrences. This certifies $R^{\mathrm{stop}}_{\exists,J}$ for the declared inventory, with each occurrence usable at most once. Under the continue contract, a state-preserving error instead leads to $(U \cup \{j\}, C_\tau(s))$. Because errors also consume an occurrence, the same induction establishes the corresponding result for $R^{\mathrm{continue}}_{\exists,J}$, provided the errors actually preserve the executor state.

A.3 Proper-loss target mismatch

For Proposition 3, let $R = R^B_\exists$, $H = H_K$, $q_R(x) = \Pr(R = 1 \mid X = x)$, and $q_H(x) = \Pr(H = 1 \mid X = x)$. Strict propriety makes the conditional mean the unique conditional-risk minimizer. Applying this result pointwise to R and H gives their Bayes predictors among unrestricted predictors, up to almost-sure equality. The learning studies evaluate finite-data training separately from these population identities.

For log loss, the conditional recoverability risk of a prediction u is $-q_R(x)\log u - (1 - q_R(x))\log(1-u)$. Subtracting its value at $u = q_R(x)$ from its value at $u = q_H(x)$ yields $\mathrm{KL}(\mathrm{Ber}(q_R(x)) \| \mathrm{Ber}(q_H(x)))$. Taking expectation over X proves Equation 7. The identity uses the extended-real convention at probabilities zero and one. Because binary Bayes entropy is bounded by $\log 2$, it requires no subtraction of infinity from infinity. Gibbs' inequality then gives nonnegativity and the equality condition.

A Bayes-optimal rollout score orders candidates by q_H, so $q_R(x_1) > q_R(x_2)$ and $q_H(x_1) < q_H(x_2)$ imply a strict reversal of the recoverability ordering. With complete deterministic instances as X and family-confined rollouts, however, $H \le R$: an unrecoverable action has $q_H = 0$, while a recoverable action with positive policy success has $q_H > 0$. Zero-support recoverable actions can tie at zero. Strict reversals under family-confined rollouts therefore require representation pooling.

A.4 Grounded symbolic certificate soundness

Fix a reproduced post-candidate ordered SAS state s_0, an integer continuation budget $B \ge 0$, and a deterministic symbolic system $\mathcal{M} = (\mathcal{S}, \mathrm{App}, T, G)$. The connection between symbolic search and official execution rests on the following interface-refinement assumptions. The planner interface enumerates every applicable grounded raw operator, implements its deterministic successor, exposes a Markov-sufficient SAS state, and evaluates the compiled goal correctly. At every relevant reachable state, each executable official command must refine to one enumerated raw transition preserving its successor, and official success must imply the compiled goal.

Proposition 4 (grounded bounded-certificate soundness). Start breadth-first search at s_0 at depth zero and check the root goal. Suppose search exhausts its queue normally, expands every discovered state at depth below B, enumerates every applicable raw operator there, and checks every generated child through depth B. If it finds no goal, no raw-operator sequence of length at most B reaches G, and the refinement assumptions exclude an official successful continuation of that length. A positive path must have length at most B and pass fresh official replay from the same reproduced

root. Replay checks admissibility, executed raw operator IDs, ordered-state continuity, counters, and goal channels. An empty path at a goal root still requires official candidate/root replay. Timeouts, resource exhaustion, incomplete enumeration, mapping faults, restoration faults, and replay faults prevent certificate admission.

For the negative direction, breadth-first search discovers each state at its minimum depth. At $B = 0$, the root check handles both alternatives. If a raw path prefix of length $d < B$ reaches a discovered state, search expands that state at depth at most d and enumerates the next edge. Induction covers every path through depth B, including states also reachable by longer paths. A bounded raw goal would therefore contradict complete negative search. An official successful path would yield the same contradiction, because it would refine to a raw path of the same length ending at a compiled goal.

For the positive direction, menu coverage is checked by grounded operator name, and selected positive paths check their actual raw IDs. This requires no separate public command for every same-name raw ID. The raw transition system can conservatively contain additional transitions that have no official rendering. Negative certificates rely on this raw transition superset and the refinement assumptions. Successful official replay from the reproduced root supplies the witness for a positive label.

B Bounded Replay and Rollout Audits

B.1 Action inventories and exact targets

The primary Retail census uses deterministic τ-bench replay. Each product node records the complete database state and the subset of indexed action occurrences already consumed. A suffix succeeds as soon as the database matches the official `final_database`, even if some occurrences remain unused. Successful no-ops also consume their occurrences. Under the stop contract, the 38-graph test population contains 232 viable states, 306 viable edges, and 33 bounded-negative candidate pairs. The negative pairs cover every source/action combination in which the action occurrence is unused, is absent from the source's backward-viable set, and has no viable outgoing edge. To assess target sensitivity, we also require all indexed writes to be consumed. This stricter target yields 230 viable states, 305 viable edges, and 34 bounded negatives.

For each of the 306 viable edges, we enumerate every permutation of the remaining occurrences after the candidate and stop execution at the first goal. Exactly 283 edges succeed under every valid permutation. The remaining 23 fallible edges span 11 tasks, and each has uniform success mass $1/2$, with either one of two or three of six permutations succeeding. Uniform attempts use independent shuffles for each pair and attempt, derived from fixed seeds. The witness control follows a stored shortest successful suffix and appends unused actions. We report exact combinatorial expectations alongside the realized 16-attempt panel. Wilson intervals summarize variation across the 23 edge pairs descriptively, as some pairs share tasks.

B.2 Misses, contamination, and coverage

Let M_K count recoverable edges with no success among the first K attempts. Forced-negative contamination on the Retail panel is $M_K/(33 + M_K)$. The exact uniform curve replaces the observed counts with their expectations, giving $\mathrm{E}[M_K]/(33+\mathrm{E}[M_K])$. For independent panels, the probability of mislabeling at least one fallible positive is $1 - \prod_{i=1}^{23}(1 - (1 - p_i)^K)$.

B.3 Continuation-policy inputs and outputs

The Retail schema-relational policy input exposes tool names, argument keys, anonymous equality relations, completed labels, the focal action, and remaining labels. It excludes task instructions, read observations, literal argument values, user identifiers, graph edges, viable sets, targets, and successful suffixes. We collect the relation set for each graph, apply a fixed graph-level shuffle, and assign opaque tokens so that equal relations receive equal tokens. Action labels and menu presentation are randomized independently for each attempt. The resulting input supports reasoning over the exposed schema relations.

We evaluate Qwen3-32B and Qwen3-Next-80B on all 306 positive edges, using 16 attempts per edge for a total of 4,896 attempts per policy. Their overall success rates are 97.04% and 97.10%, respectively. On the 23 fallible edges, the corresponding rates are 62.50% and 61.41%. Each policy leaves one positive edge without a successful continuation at $K = 16$, as shown in the full-census results in Figure 2.

B.4 Workspace and Slack accounting

Workspace contains 465 positive decisions and 13 structural negatives across 25 official tasks. Its certificates are constructed using fresh replay of all 240 released full orders in the official runtime, together with checks of prefix bounds. At $K = 1$, Qwen3-32B and Qwen3-Next-80B miss four and two positives, respectively. These misses give forced-negative contamination of 4/17 and 2/15, with three-way coverage of 99.16% and 99.58%. In the realized panels, all positives acquire witnesses by $K = 16$ and $K = 4$, respectively.

Slack contains 681 positives and 18 bounded negatives. Table 5 reports the fixed and adaptive results at each specified budget.

A post-hoc task-cluster bootstrap with 10,000 draws summarizes sensitivity to task composition within the censuses. At $K = 1$, the Retail contamination intervals for the two policies have centers and half-widths of $18.77 \pm 8.51\%$ and $16.81 \pm 7.72\%$, respectively. The corresponding Workspace values are $50.00 \pm 50.00\%$ and $11.11 \pm 11.11\%$.

B.5 Additional census populations and construction cost

The Retail training census enumerates all post-edge permutations for 1,254 positive edges in 151 task-disjoint training graphs, each with at most five occurrences. It identifies 132 fallible edges across 32 tasks. Ten edges in four tasks have success mass below $1/2$, with a minimum of $3/24 = 1/8$. The full histogram of rational success masses and the edge-level records show how success mass varies within this executor and target.

The Airline census contains 191 positives and 11 bounded negatives in 15 official-test graphs. Five edges across two tasks are fallible, with a minimum uniform success mass of $1/6$. We compare forced binary labels, witness-only positives, witness positives combined with bounded negatives, and exact bounded labels. Combining witnesses with certificates achieves 98.60% coverage at $K = 1$ and zero selective error by construction on this census.

Exact-label construction is an offline preprocessing step whose cost is accounted for separately from model training and evaluation. Retail and Airline enumerate permutations of at most five indexed mutating occurrences after a candidate, yielding at most $5! = 120$ local continuations before product-state reuse. Workspace replays 240 released full orders, while Slack enumerates its declared indexed-reference family. The number of permutations grows factorially with inventory size. Replay counts measure the work performed within these bounded families, and wall time reflects the cost of each execution and the extent of state reuse.

C Released Data and Native Target Builders

C.1 Stored fields in the IPR release

The expert-trajectory release accompanying Iterative Step-level Process Refinement (IPR) contains 43,897 step records across ALFWorld, InterCodeSQL, and WebShop. We compare the stored expert terminal reward with the stored aggregate Monte Carlo step value. Among the 41,560 records with expert terminal reward one, 11,566 have Monte Carlo value zero (Table 6).

C.2 Complete native-builder panel

We fix a panel of 113 decisions before inspecting builder outputs. It contains 49 bounded-recoverable failed decisions and 64 bounded negatives, distributed across 56 Retail, 22 Workspace, and 35 Slack cases. Each of the six released builders receives its native failed-rollout, demonstration, pair, or step-judge input. We join successful witnesses and exact recoverability labels after the native outputs

Table 4: Rollout accounting on the Retail and Workspace censuses. P/Ncert denotes recoverable/certified-negative decisions, and Miss counts positives without a witness through K. Contamination and coverage are reported as percentages. Three-way labeling abstains on unresolved cases.

Domain	Policy	K	P/Ncert	Miss	Forced contam.	3-way cov.	3-way contam.
Retail	Qwen-32B	1	306/33	8	19.51	97.64	0.00
Retail	Qwen-32B	16	306/33	1	2.94	99.71	0.00
Retail	Qwen-80B	1	306/33	7	17.50	97.94	0.00
Retail	Qwen-80B	16	306/33	1	2.94	99.71	0.00
Workspace	Qwen-32B	1	465/13	4	23.53	99.16	0.00
Workspace	Qwen-32B	16	465/13	0	0.00	100.00	0.00
Workspace	Qwen-80B	1	465/13	2	13.33	99.58	0.00
Workspace	Qwen-80B	4	465/13	0	0.00	100.00	0.00

Table 5: Preregistered Slack accounting at every frozen budget. Miss is the number of oracle-positive decisions without a witnessed success. Forced contamination is the percentage of forced-binary negative labels that are oracle-positive. Three-way labeling admits the 18 bounded certificates as negatives and abstains on unresolved positives. Adaptive calls are reported against the same PK cap as fixed K and may be lower only after all positives resolve. Percentages are complete-census micro values.

Policy	K	Fixed miss	Adapt. miss	Fixed contam.	Adapt. contam.	Fixed cov.	Adapt. cov.	Adapt. calls/cap
Qwen-32B	1	24	24	57.14	57.14	96.57	96.57	681/681
Qwen-32B	2	2	0	10.00	0.00	99.71	100.00	707/1362
Qwen-32B	4	0	0	0.00	0.00	100.00	100.00	707/2724
Qwen-32B	8	0	0	0.00	0.00	100.00	100.00	707/5448
Qwen-32B	16	0	0	0.00	0.00	100.00	100.00	707/10896
Qwen-32B	32	0	0	0.00	0.00	100.00	100.00	707/21792
Qwen-80B	1	7	7	28.00	28.00	99.00	99.00	681/681
Qwen-80B	2	0	0	0.00	0.00	100.00	100.00	688/1362
Qwen-80B	4	0	0	0.00	0.00	100.00	100.00	688/2724
Qwen-80B	8	0	0	0.00	0.00	100.00	100.00	688/5448
Qwen-80B	16	0	0	0.00	0.00	100.00	100.00	688/10896
Qwen-80B	32	0	0	0.00	0.00	100.00	100.00	688/21792

are fixed. All builders complete every decision, producing 678/678 valid outputs with no missing outputs, parse failures, or runtime failures. Table 7 breaks down the overall target behavior reported in the main text by environment.

The six builders retain their native target semantics and emit negative labels at rates ranging from 0 to 100% on the fixed panel. Among builders that emit negatives, the recoverable fraction ranges from 20.83 to 43.36%. This conditional fraction is undefined for a builder that emits no negatives.

Earlier conditional slice. The preliminary stress test uses the 49 recoverable decisions subsequently included in the complete panel, comprising 23 Retail, nine Workspace, and 17 Slack cases. AgentPRM Monte Carlo, InversePRM demonstration labeling, IPR Monte Carlo, and ToolPRMBench pair construction each emit 49 negatives. ToolRM emits ten, and AgentPRM shaped emits none. This outcome-conditional slice is part of the 113-decision panel and uses the same cases, so it contributes neither an additional replication nor a different denominator to the main results.

C.3 Related decision objects

Table 8 compares the decision objects addressed by related methods with per-instance bounded-recoverability certification.

Table 6: Stored-field audit of the IPR release. E denotes records with stored expert terminal reward one, and Both denotes E together with Monte Carlo value zero. The last column gives the point estimate and a released-ID-cluster bootstrap 95% interval as center $\pm$ half-width.

Domain	Records	MC=0	E	Both	Both/E (point; 95% CI)
ALFWorld	30,657	10,201	30,657	10,201	33.27%; $33.25 \pm 1.93\%$
InterCodeSQL	7,088	783	7,088	783	11.05%; $11.05 \pm 1.24\%$
WebShop	6,152	748	3,815	582	15.26%; $15.29 \pm 1.33\%$
All	43,897	11,732	41,560	11,566	27.83%; $27.83 \pm 1.53\%$

Table 7: Complete builder audit by environment. Each cell is $n^-/N\,[r/n^-]$: native negatives among eligible decisions, followed by bounded-recoverable decisions among emitted negatives. "–" applies when $n^- = 0$.

Builder	Retail (56)	Workspace (22)	Slack (35)
AgentPRM Monte Carlo	56/56 [23/56]	22/22 [9/22]	35/35 [17/35]
InversePRM demonstration	56/56 [23/56]	22/22 [9/22]	35/35 [17/35]
IPR MC	56/56 [23/56]	22/22 [9/22]	35/35 [17/35]
ToolPRMBench pair construction	56/56 [23/56]	22/22 [9/22]	35/35 [17/35]
ToolRM released pairwise	10/56 [3/10]	15/22 [2/15]	23/35 [5/23]
AgentPRM shaped	0/56 [–]	0/22 [–]	0/35 [–]

D Four-Arm Retail Learning Study

D.1 Candidate construction and label realization

The intervention uses the official 74/40 training/test split of τ^2 Retail at a fixed benchmark release. Each menu contains the official mutating reference calls and at most two calls drawn from training-only donors in a deterministic order. All candidates are retained regardless of execution outcome. We construct the finite graphs through native tool execution and compute bounded targets by backward reachability under the executor assumptions in Proposition 2. The exact labels and sampled training labels both use the continue contract.

The dataset contains 2,757 training rows (422 exact positives and 2,335 negatives) and 1,099 test rows (161 positives and 938 negatives). For each candidate, a single rollout shuffles the remaining occurrence indices with a deterministic seed that is fixed across training seeds. The reproduction configuration records the seed derivation and ordering rule. This rollout misses 217 training positives and 81 test positives.

Forced $K = 1$ and Exact retain all 2,757 training rows. Certificate/abstain omits the 217 recoverable rows whose fixed $K = 1$ rollout fails. Matched deletion applies the same task-by-source quotas to remove 217 $K = 1$-negative rows, comprising 182 exact negatives and 35 exact positives. Each shorter arm cycles its 2,540 retained rows to reach 2,757 example exposures. All arms use the same model, optimizer, and endpoint definitions, receive 345 optimizer steps, and share initialization within each of three paired seeds.

D.2 State representation and verification

Each graph key includes the full serialized database and the consumed candidate occurrences. Tools execute from cloned full-state representatives, and histories with the same key share one expanded representative. The saved-graph audit checks remaining-action coverage, record consistency, and backward reachability for all 3,856 candidate transitions and labels. It reads stored graphs rather than rerunning native tools. A separate, earlier τ-bench package replays 190 candidate pairs from 183 source nodes across 70 tasks and records zero source-state mismatches or positive/negative suffix-label violations. Evaluating those 190 pairs in both orders produces 380 scorer cases.

Table 8: Related methods and the decisions they draw from failure evidence. Our contract distinguishes policy value, bounded certificates, and abstention. We evaluate this distinction through audits of released builders and a training intervention on the supervision target.

Work	Decision object	Licensed use of failure evidence	Evaluation
NoiseAware	Intrinsic reasoning-step correctness	Reflection-aware correction and confidence-based label refinement	Reasoning process rewards
AgentPro	MCTS state label	Emit a negative when all three sampled paths fail	Tool-agent training
Failed Traces / SENTINEL	Failure signature or recurring error	Route rescue or generate a failure curriculum	Cross-family rescue / Retail
ERR / DART	Recovery regret or rollback boundary	Predict recovery cost or admit safe rollback	Recovery and safety audits
Ours	State–action plus declared continuation family	Declare the estimand; certify a bounded negative or abstain	Exact censuses, six builders, matched Retail training

D.3 Deletion strata and training exposures

Task identity and candidate source (official or training-only donor) define 134 strata across 67 candidate-bearing training tasks. Certificate omissions determine each stratum's deletion quota. Matched deletion fills that quota from $K = 1$ negatives using a deterministic seeded ordering based on the task, source, candidate identifier, context prompt, and candidate action. The reproduction configuration specifies this ordering rule. Quota sizes therefore use Certificate information, while selection within each quota does not consult exact recoverability.

We align rows by task, source, and assigned $K = 1$ label, then apply a common seeded shuffle and cycle the rows. Each shorter arm presents 2,323 retained rows once and 217 twice. The exposure check uses six saved post-training multiplicity files and schedule reconstruction, as no per-forward trace was stored. The files confirm equal positive and negative exposures within every matching group (Table 9), and the reconstructed schedule reproduces both the multiplicities and the ordered matching-group sequence.

Table 9: Actual assigned-label exposures after cycling. Each arm contains 205 distinct positive rows and 2,335 distinct negative rows. The reported exposure counts include repetitions.

Seed	Certificate/abstain		Matched deletion	
	Positive	Negative	Positive	Negative
2026082817610	217	2,540	217	2,540
2026082817611	226	2,531	226	2,531
2026082817612	224	2,533	224	2,533

D.4 Retained-sample composition

A descriptive audit joins the original rows to the saved multiplicities to examine both distinct candidates and exposure-weighted samples (Table 10). Certificate and Matched deletion share 2,358 of their 2,540 retained identifiers. Task, source, and assigned-label counts match in every stratum, while exact-positive prevalence is 205/2,540 versus 387/2,540 (8.07% versus 15.24%). The corpus has no native task-type field, so the audit groups rows by task identity. Across the four views, tool-name total variation is 0.00508–0.00689 and the standardized content-length difference is $+0.084$–$+0.091$, with length measured in saved Unicode characters before tokenization. Under matched assigned-label exposure, the two arms thus differ in their admission rules and the recoverability composition of their retained samples.

Table 10: Retained-set balance. C/M denotes Certificate/Matched deletion. The Unique view gives each retained candidate unit weight, and views a–c use the three saved exposure schedules (seeds 2026082817610–12). Exact positives denote recoverable candidates rather than assigned training positives. Content length counts Unicode characters in the saved context and action before chat wrapping or tokenization. SMD divides the C–M mean difference by the pooled population SD. Tool TV measures total variation over six exact tool names. Task, source, and assigned-label TV are zero in every view.

View	Total/arm	Exact positives C/M	Mean chars C/M	SMD	Tool TV
Unique	2,540	205/387	3595.8/3460.5	+0.091	0.0059
Seed a	2,757	217/417	3603.4/3467.7	+0.091	0.0051
Seed b	2,757	226/424	3597.4/3466.1	+0.088	0.0069
Seed c	2,757	224/419	3600.4/3474.8	+0.084	0.0069

Table 11: Main four-arm 4B training configuration, distinct from the frozen-representation experiment. All arms share these settings.

Setting	Value
Checkpoint	Skywork-Reward-V2-Qwen3-4B
Trainable scope	Scalar head and final two transformer blocks
Precision / loss	BF16 backbone; FP32 logits for binary cross-entropy
AdamW	Learning rate 10^{-5}; betas $(0.9, 0.95)$; weight decay 0.01
Batch / accumulation	Micro-batch 1; accumulate 8; final 5-example partial batch stepped
Gradient clipping	Global norm 1.0
Input	Maximum 4,096 tokens; tokenizer truncation enabled and right padding
Training budget	2,757 exposures; 345 optimizer steps
Seeds	2026082817610, 2026082817611, 2026082817612

D.5 Training configuration

Table 11 gives the 4B fine-tuning configuration. The trainer inherits truncation behavior from the checkpoint tokenizer, and the checked Forced run records zero truncated training or test examples. The frozen 0.6B representation study in Section F.3 uses a different dataset and trainable scope.

D.6 Metrics and uncertainty

Recoverability log loss averages candidate losses within each task and then gives each task equal weight. Top-1 first averages over states within each task, assigning zero to all-negative states. Its 465-state test population gives a task-macro uniform-selection baseline of 9.97% and a label-informed oracle ceiling of 21.41%. Pairwise accuracy requires both classes and covers 35 tasks. NDCG excludes states with zero ideal DCG and covers 36 tasks, as do log loss and Top-1. Terminal evaluation covers all 40 tasks under the stop contract, while offline labels use continue. It greedily selects the highest-scored unused action until success, tool error, or inventory exhaustion.

We average each task's results over the three realized seeds before taking 10,000 paired task-bootstrap draws. The confidence intervals describe task variation conditional on these training seeds and the fixed rollout labels. Table 12 reports absolute performance for all arms, with seed standard deviations shown separately from the task-bootstrap intervals in Table 13. Table 14 gives each seed's effects for all original paired contrasts.

Outputs from the six additional runs were fixed before analysis. The Exact–Forced contrast in recoverability log loss is -0.14987, with a paired 95% interval of $[-0.32988, -0.02605]$. The corresponding Certificate–Matched contrast is -0.04952, with interval $[-0.08227, -0.03010]$, and the secondary Certificate–Exact contrast is $+0.10139$, with unadjusted interval $[0.00082, 0.24236]$. For all three contrasts, the intervals for Top-1, pairwise accuracy, NDCG, and terminal success include zero.

Table 12: Absolute performance in the four-arm Retail intervention. Entries give the mean ± sample SD across the same three realized seeds. SD describes variation across seeds rather than a confidence interval. Log loss, Top-1, and NDCG average 36 tasks, pairwise accuracy averages 35, and terminal success covers all 40. Non-loss metrics are percentages. Top-1 assigns zero to all-negative states. The uniform random Top-1 baseline is 9.97%, and the label-informed oracle ceiling is 21.41%. Paired task-bootstrap contrasts appear in Table 13.

Arm	Exact loss ↓	Top-1	Pairwise	NDCG	Terminal
Forced $K = 1$	0.481 ± 0.055	15.54 ± 0.18	75.69 ± 0.42	86.22 ± 0.26	61.67 ± 1.44
Certificate/abstain	0.433 ± 0.131	15.80 ± 0.39	75.59 ± 1.71	86.50 ± 0.96	65.00 ± 2.50
Exact relabel	0.331 ± 0.011	15.91 ± 0.04	75.42 ± 0.50	86.56 ± 0.26	65.00 ± 0.00
Matched drop	0.482 ± 0.164	15.87 ± 0.24	75.94 ± 0.87	86.76 ± 0.56	64.17 ± 1.44

Table 13: Four-arm Retail contrasts with paired task-bootstrap 95% intervals, conditional on the three realized seeds. Non-loss endpoints are percentage-point changes. The two primary loss contrasts appear in bold, and the secondary Certificate–Exact contrast uses an unadjusted interval. Training exposures and update counts are matched as specified in Table 11.

Contrast	Exact loss ↓	Top-1	Pairwise	NDCG	Terminal
Exact relabel − forced	**-0.150 [-0.330,-0.026]**	+0.37 [-0.91,+1.49]	-0.27 [-6.13,+5.29]	+0.34 [-2.56,+2.90]	+3.33 [-5.00,+11.67]
Certificate − matched drop	**-0.050 [-0.082,-0.030]**	-0.07 [-0.51,+0.34]	-0.35 [-1.71,+0.77]	-0.26 [-1.10,+0.32]	+0.83 [-2.50,+5.00]
Certificate − exact relabel	+0.101 [+0.001,+0.242]	-0.11 [-1.19,+1.10]	+0.17 [-5.05,+5.65]	-0.06 [-2.66,+2.81]	-0.00 [-5.83,+6.67]

E Probability Calibration and Diagnostics

E.1 Task-held-out calibration

Using saved predictions on the 36 candidate-bearing Retail test tasks, we fit $\widetilde{p} = \sigma(a \operatorname{logit}(p) + b)$ separately for each arm and seed. Each fold holds out one complete task for scoring and fits on the exact labels of the other 35 tasks, giving each fitting task equal weight. Bias-only calibration fixes $a = 1$, while positive-slope logistic calibration fits $a > 0$ and b. We constrain the parameters to $a \in [10^{-6}, 100]$ and $b \in [-100, 100]$, and none of the 864 fitted maps reaches a bound. The maps are task-specific, so pooled cross-task rankings may change. Both transformations preserve rankings within each held-out task, arm, and seed.

Table 3 reports loss averaged first over tasks and then over the three realized seeds. We compute gaps from unrounded losses, so subtracting rounded table entries can change the final digit. Positive-slope calibration reduces the Forced–Exact gap from 0.149873 to 0.024473 and the Matched–Certificate gap from 0.049523 to 0.010846. Bias-only calibration gives comparable gaps of 0.0235 and 0.0106 while preserving rankings. The logistic-calibrated seed SDs for Forced, Certificate, Exact, and Matched are 0.002260, 0.000588, 0.003616, and 0.001654. Calibrated Forced reaches a loss of 0.332576, close to the uncalibrated Exact loss of 0.331419.

E.2 Constant predictors and calibration-label cost

For held-out task t, let n_u be the number of rows in fitting task u, and let y_{ui} be their exact labels. The common cross-fitted constant is

$$c_{-t} = \frac{1}{35} \sum_{u \neq t} \left(\frac{1}{n_u} \sum_{i=1}^{n_u} y_{ui} \right). \tag{8}$$

This constant gives each fitting task equal weight and uses the same exact fitting labels as the calibrators. Within a fold, it is identical across arms and seeds. Labels from the held-out task are used only for scoring, giving a task-macro log loss of

$$L_{\text{const}} = \frac{1}{36} \sum_{t=1}^{36} \frac{1}{n_t} \sum_{i=1}^{n_t} [-y_{ti} \log c_{-t} - (1 - y_{ti}) \log(1 - c_{-t})] = 0.421690. \tag{9}$$

The cross-fitted constant assigns the same score to all candidates, whereas the calibration maps preserve their rankings. It uses exact labels from the fitting tasks, as do the calibrators. We also

Table 14: Paired effects for all three original contrasts across the realized Retail seeds. Seeds a, b, and c are 2026082817610, 2026082817611, and 2026082817612. Loss differences use original units, and other differences are percentage points.

Contrast	Seed	Exact loss	Top-1	Pairwise	NDCG	Terminal
Exact − forced	a	−0.148	+0.24	−1.32	−0.10	+2.50
Exact − forced	b	−0.214	+0.58	+0.25	+0.80	+5.00
Exact − forced	c	−0.088	+0.29	+0.27	+0.33	+2.50
Certificate − matched	a	−0.087	+0.37	+1.25	+0.91	+2.50
Certificate − matched	b	−0.011	+0.15	+0.41	+0.18	+2.50
Certificate − matched	c	−0.050	−0.73	−2.71	−1.88	−2.50
Certificate − exact	a	+0.244	+0.33	+2.55	+1.18	+2.50
Certificate − exact	b	+0.046	−0.09	−0.35	−0.19	+0.00
Certificate − exact	c	+0.013	−0.55	−1.69	−1.18	−2.50

evaluate separate constants based on each arm's training-label prevalence. On the same 36-task metric, these yield losses of 0.445937 for Forced ($p = 205/2757$) and 0.416251 for Exact ($p = 422/2757$).

E.3 Brier score, signed bias, and ECE

The probability diagnostics cover all 1,099 candidates at 465 states across every arm and all three seeds. Their definitions were fixed after the primary log-loss results were known. Brier score averages $(p - y)^2$ within each task and then equally over tasks and seeds. We compute intervals from 10,000 paired task-bootstrap draws conditional on the realized seeds, without multiple-comparison adjustment. Certificate has a lower Brier score than Matched for all three seeds, with the task-bootstrap interval for the Certificate–Matched contrast lying entirely below zero. The intervals for the other two contrasts include zero (Table 15).

For signed bias and ECE, each candidate–seed observation in task t receives weight $1/(36 \cdot 3 \cdot n_t)$. Bias subtracts the weighted mean exact label from the weighted mean probability. ECE_{10} uses ten fixed equal-width bins on $[0, 1]$ and sums the product of each bin's mass and its absolute probability–label gap. Only the last bin includes its right endpoint, and empty bins contribute zero. The bins pool observations from the three models using task weights, rather than averaging seedwise ECE or scoring ensemble predictions. Alongside the decision metrics, these statistics describe the saved probability predictions and show nonzero signed bias in every arm.

F Sampling-Budget and Representation Sensitivity

F.1 Nested rollout-budget labels

The budget analysis uses the same 2,757 training and 1,099 test candidates, action inventories, and exact targets as the four-arm study. For each candidate, 16 pseudorandom permutations of the remaining actions are sampled with replacement. Labels at $K = 1, 4, 8, 16$ use nested prefixes of these attempts, with the first attempt reproducing the original single-rollout realization. Subsequent attempts use deterministic seeds indexed by candidate and attempt and recorded in the reproduction configuration. Each attempt shuffles an initially ascending list of remaining indices and executes actions until it reaches the goal or exhausts the inventory. Hard targets indicate whether any attempt succeeds, while soft targets give the fraction of successful attempts.

Table 16 reports missed positives and within-state positive–negative pairs in which both candidates receive zero hard labels. These pairs represent target ties, distinct from scorer inversions. Each post-candidate state is cached once, and transition counts include all K attempts, including those after a success, to support soft labels. The counts measure suffix evaluation on saved graphs. They do not measure the cost of stopping at the first success for hard labels or the throughput of native tools.

Table 15: Post-hoc probability-quality diagnostics on the same Retail predictions. Brier measures task-macro squared probability error (lower is better), and SD describes variation across the three realized seed means. Bias is mean probability minus mean exact label. ECE_{10} uses a task-weighted mixture of the three models. The lower panel reports all three ordered Brier contrasts with unadjusted 95% task-bootstrap intervals, conditional on those seeds.

Arm	Brier	Seed SD	Bias	ECE_{10}
Forced $K = 1$	0.12589	0.00610	-0.11371	0.11371
Certificate/abstain	0.11552	0.01651	-0.08879	0.09593
Exact relabel	0.10521	0.00472	+0.05442	0.05446
Matched drop	0.12209	0.01529	-0.10085	0.10085

Ordered contrast	Difference	95% interval
Exact minus forced	-0.02068	[-0.07302, +0.01373]
Certificate minus matched	-0.00656	[-0.01157, -0.00336]
Certificate minus exact	+0.01032	[-0.01957, +0.05542]

Table 16: Nested rollout-label accounting on the original four-arm candidate population. Positive misses are counted among 422 training and 161 test positives. Zero-label ties are counted among 820 training and 317 test positive–negative pairs within states. Transition counts include all attempts through K.

K	Positive misses		Zero-label ties		Suffix transitions	
	Train	Test	Train	Test	Train	Test
1	217	81	427	162	5,256	1,931
4	98	38	193	76	21,024	7,709
8	38	18	75	36	42,039	15,373
16	18	4	36	8	84,056	30,747

F.2 Completed training comparisons

Study execution. The original plan comprised eight arms (Exact, hard $K = 1, 4, 8, 16$, and soft $K = 4, 8, 16$) across three seeds, for a total of 24 runs. The first queue block completed Exact and hard $K = 1, 4, 8$ with seed 2026082817610. After inspection of these results, the plan was reduced to one seed and then stopped during manuscript revision. The other four runs for the first seed were interrupted, and the remaining 16 were cancelled. This outcome-aware stopping produced an incomplete exploratory study rather than a prospectively fixed four-arm, single-seed experiment. Table 17 reports all completed runs.

The four completed runs share the same initialization and shuffled training order, each with 2,757 exposures and 345 optimizer updates. They use the 4B backbone, trainable layers, BCE loss, and optimizer specified in Table 11. All run on a common CPU backend with 16 operator threads and one inter-op thread per model and complete the specified schedule without representation caching, additional layer freezing, or test calibration. Exact and hard $K = 1, 4, 8$ receive 422, 205, 324, and 384 assigned positive exposures, respectively, with negative labels for the remaining exposures.

Evaluation uses the same previously examined 40-task inventory. Metrics cover 1,099 candidates at 465 states in the 36 tasks containing candidates. Log loss clips probabilities to $[10^{-8}, 1 - 10^{-8}]$ and averages first over rows within each task and then over tasks. Top-1 similarly averages first over states within each task and then over tasks, with ties resolved by a fixed ordering of candidate identifiers. Brier score is averaged over rows in this table, unlike the task-macro diagnostic in Table 15. Checks of all four runs cover completion records, inputs, and complete predictions, with recomputed metrics agreeing within 10^{-12}.

Hard $K = 4$ and $K = 8$ reduce loss relative to Exact by 0.02290 and 0.01806, respectively. All three arms have the same aggregate Top-1 rate, although their individual action choices may differ. These point estimates show that the loss ordering relative to Exact changes as the rollout budget increases from $K = 1$.

Table 17: Completed exploratory Retail rollout-budget runs on a common CPU backend. Each uses seed 2026082817610, 2,757 example exposures, and 345 updates. Scoring covers 1,099 candidates in 36 tasks, using task-macro log loss and Top-1 and row-averaged Brier score. The accompanying text describes execution status and stopping.

Training target	Exact log loss ↓	Exact Top1 (%) ↑	Brier ↓
Exact	0.3389	16.33	0.1073
Hard $K = 1$	0.4911	16.13	0.1344
Hard $K = 4$	0.3160	16.33	0.1071
Hard $K = 8$	0.3209	16.33	0.1063

Computation. With all four runs sharing the CPU, training/evaluation take 25.89/8.40 wall hours for Exact, 27.56/9.06 for $K = 1$, 29.53/9.65 for $K = 4$, and 29.60/9.40 for $K = 8$. Training time is measured from the end of model loading and input rendering. These durations describe individual run phases rather than summed CPU-core hours or total allocation cost. Table 16 reports label-generation work separately.

F.3 Learning with frozen representations

The head-only study uses 1,411 training candidates from 151 task-disjoint Retail graphs (1,254 positives and 157 negatives) and 339 test candidates from 38 graphs (306 positives and 33 negatives). Inputs contain the original instructions, read-context excerpts, executed writes, candidate calls, and relevant schemas. Prompt checks confirm the exclusion of graph edges, viable sets, state and target identifiers, witness suffixes, success masses, and labels. Training updates only the scalar head of the frozen Skywork-Reward-V2-Qwen3-0.6B backbone for one epoch, using BCE, batch size four, AdamW at 10^{-4}, BF16, and a maximum length of 4,096. The Exact condition, five $K = 1$ rollout-label panels, and the $K = 4$ and $K = 16$ conditions share the same training seed, initialization, optimizer, data order, and test targets.

The exact-label head achieves task-macro recoverability log loss of 0.4648 and AUROC of 0.558. The untrained head and the constant training-prevalence predictor have losses of 1.687 and 0.306, respectively. The five $K = 1$ heads attain losses of 0.351–0.401, while the hard $K = 4$ and $K = 16$ heads attain 0.4590 and 0.4647. The pooled $K = 1$ hard-minus-exact loss difference is -0.0925, with a 95% task-cluster percentile bootstrap interval of $[-0.1571, -0.0312]$ from 10,000 draws over 38 tasks, retaining all five label panels in each draw. Saved predictions, heads, and metrics pass consistency checks. All five $K = 1$ heads assign lower scores to fallible positives while achieving lower aggregate loss than the exact-label head, contradicting the prespecified aggregate-degradation hypothesis in this setting.

The head-only experiment changes labels while retaining every candidate and freezing the representation. In the primary four-arm study, a 4B model is fine-tuned on a separately constructed τ^2 corpus, with sample-admission interventions and terminal execution. The two studies therefore report losses for different candidate populations and training configurations.

F.4 Sparse relational-label diagnostic

A preceding diagnostic uses the same 1,411/339 census rows with an L2 logistic model, $C = 1$, a 5,000-iteration cap, and fixed sparse relational features. Its protocol was fixed before the blinded-policy outcomes were inspected. Parameters are represented by anonymous relation keys, with features excluding state and target identifiers, graph viability, witnesses, success masses, instructions, observations, literal values, and action indices. The diagnostic compares exact labels with hard uniform-rollout labels at $K \in \{1, 4, 16\}$ across 20 specified panel seeds. The audit recomputes the census, feature exclusions, sampled labels, model fits, and metrics, confirming that labels and features remain unchanged.

F.5 TOOL-ERROR TARGET SENSITIVITY

The four-arm label graph consumes an occurrence after a state-preserving tool error and permits continuation, whereas terminal policy evaluation stops at the error. Switching from continuation to stopping reduces training positives from 422 to 415 by changing seven labels and test positives from 161 to 159 by changing two, with no additional downstream-only flips. One sampled-$K = 1$ positive also changes in each split. This post-hoc sensitivity analysis rescores existing predictions against the revised evaluation labels without retraining. The resulting Exact–Forced and Certificate–Matched loss contrasts are -0.13728 and -0.04734, respectively, and their ranking intervals still include zero.

G ADDITIONAL ENVIRONMENT STUDIES

G.1 GROUNDED ALFWORLD SEARCH

The development full-menu search evaluates 2,018 candidate commands at 64 states reached across 22 tasks, with checkpoints after 24, 36, or 40 executed prefix actions. Search allows up to four continuation actions after the focal command, with a 1,500-second deadline. It identifies 308 positives and 1,596 bounded negatives, yielding 94.35% binary coverage (Table 18). The remaining 114 cases are unknown because of controlled timeouts, and no engineering failures are recorded. With 32 workers, the run takes 8.87 wall-clock hours and a summed elapsed time of 282.97 hours across worker processes.

A mixed state contains at least one certified positive and one certified negative. The fixed selection criteria require complete binary coverage overall, within every state, and within each represented action family, with a positive fraction in $[0.1, 0.9]$. They also require at least 18 reached tasks and mixed labels in at least 50% of reached states. Only 11/64 states are definitely mixed. Even assigning unknowns favorably yields at most 17 potentially mixed states, with the proportion below 29% in every cell. Consequently, no cell or two-checkpoint recipe qualifies. Heat-then-place accounts for 109 of the 114 unknowns and has coverage of 73.80%. These results distinguish the coverage of command-level certification from the availability of informative comparisons within states.

G.2 TELECOM TERMINAL INTERVENTION

The four-arm Telecom trial fixes a population of 949 training tasks and 42,791 candidate rows before training, including 1,174 certified negatives and 858 recoverable $K = 1$ failures. Forced and Exact retain every row. Certificate omits the 858 unresolved rows, while Matched removes 858 certified negatives selected deterministically within specified strata. Arms with fewer retained rows cycle through them to reach 42,791 example exposures. A microbatch size of one with accumulation over eight examples gives 5,349 optimizer steps, including a final partial batch of seven examples. All arms share the same model, tokenizer, initialization, optimizer, prompt, parser, candidates, and terminal evaluator within each of the 12 seed blocks.

Each checkpoint is evaluated on 1,288 official `ENV_ASSERTION` tasks with fresh resets. Checks of 48 training and 48 evaluation records confirm 61,824 task rows. Forced, Certificate, Exact, and Matched obtain 15,406, 15,408, 15,411, and 14,383 successes, respectively, from 15,456 evaluations per arm. The Certificate–Forced difference is $+0.0129$ percentage points, with a bootstrap interval center and half-width of $+0.0162 \pm 0.1197$ and positive effects in 25.00% of seed blocks. The Certificate–Matched difference is $+6.63$ points, with an interval center and half-width of $+6.75 \pm 3.14$.

Table 18: ALFWorld development certification at budget 4. Checkpoint counts executed prefix actions, and jobs correspond to candidate commands. Mixed states contain at least one certified positive and one certified negative.

Checkpoint	Jobs	Positive	Negative	Unknown	Mixed/reached
24	697	151	542	4	3/22
36	664	78	530	56	4/21
40	657	79	524	54	4/21

Before training, an amendment replaced infeasible fine-grained matching with a hierarchy that used coarse cross-requestor fallbacks for 168/858 matches. The reported Matched result uses this amended construction, for which no sensitivity analysis restricted to the original match levels was supplied. Its lower terminal success therefore characterizes this particular admission control and is reported separately from the primary comparison against Forced.

G.3 TELECOM LEARNED-TRIAGE TRANSFER

The triage study uses programmatic Telecom development graphs with all 40 official test identifiers excluded. The population contains 949 graphs, 18,717 nodes, 41,617 positive edges, and 1,174 negatives. Checks of 6,439 negative suffix orders record zero failures. The conditional fitting and calibration sets contain 1,660 and 352 rows, respectively. A sparse model and three Skywork-Reward-V2-Qwen3-4B models are fixed before test access. Each neural model trains its final two blocks and scalar head for one epoch. Both methods achieve development AUROC 1.000.

Of the 40 official test tasks, 16 satisfy the two-to-five-action bound, and three contribute 49 conditional all-failure rows comprising 19 hidden recoverable cases and 30 exact negatives. Fresh replay checks 238 nodes, 483 positive edges, 30 negatives, and 137 negative suffix orders, recording zero failures. Test AUROC/log loss are 0.6368/1.1598 for the sparse model and 0.6246/4.0372 for the neural ensemble, compared with a constant-prior loss of 0.6691. Their selective contamination rates are 31.82% and 27.78%, with one-sided 95% upper bounds of 45.19% and 42.55%, respectively. Inference checks pass, but the prespecified confirmation criterion is not met. The three task clusters give a smallest attainable exact sign-flip p-value of $1/8$. Thresholds, splits, rollouts, and training recipes remain unchanged after test exposure.

Table 19: Telecom triage performance on programmatic development and official test tasks. Loss is averaged over rows. Coverage is the fraction of conditional all-failure rows admitted as negatives; contamination is the fraction of these admitted negatives that are bounded-recoverable.

	Development		Official test			
Method	AUROC	Loss	AUROC	Loss	Coverage	Contam.
Sparse	1.000	0.0061	0.6368	1.1598	89.80%	31.82%
4B ensemble	1.000	0.0002	0.6246	4.0372	73.47%	27.78%

Table 20 reports results for individual Telecom tasks. Both models show limited discrimination on the two mixed-label tasks, with AUROC ranging from 0.571 to 0.655. The all-negative task contains only two examples, on which both models have near-zero losses.

Table 20: Per-task Telecom triage results. P/N denote hidden-recoverable and exact-negative counts. AUROC is undefined for the all-negative task.

Task	P	N	Sparse AUROC	Sparse loss	4B AUROC	4B loss
Service: airplane/bill	12	14	0.655	1.226	0.589	4.674
Service: bill/SIM	7	14	0.571	1.187	0.582	3.633
MMS: APN/roaming	0	2	–	0.002	–	0.00002