When Sufficient Representations Fail under Deployment

Anonymous authors
Paper under double-blind review

Abstract

Exact sufficiency on a current support does not guarantee sufficiency for the same target after actions selected through the representation. We call the missing property *deployment closure*. For every $K \geq 2$, two minimum-cardinality K-state representations can agree on the certified support and induce the same deployment, while only one remains sufficient afterward. Repair changes the quantifier: one target-distinct collision refutes closure, whereas interface repair must resolve every such collision under the information actually available to the decision maker. In ABIDES, identical public books hide ownership distinctions that become consequential after a public-book-selected trade; a prospective 128-pair study yields a graded evidence–risk frontier, while reanalysis without pair identity shows that repairing designated pairs need not repair the retained interface. In a controlled learned-world-model assay, recurrent state retains a payload distinction that the native transition fails to use; a transition repair fixed before testing reduces real regret to 0.0044 of a q-only baseline on 128 fresh matched pairs, with history-erasure and carrier-transplant controls. Together, the results separate current certification, reached-support sufficiency, and access to decision-relevant repair information.

1 Introduction

A trading controller observes the same public order book in two worlds and chooses the same trade. Before that trade, a fixed future-response program gives the same answer in both worlds. Afterward, the public books still match, but that same program gives different answers: the trade has exposed queue-ownership differences that the public interface omits. Neither the controller nor the response program has changed. What has changed is where the controller has taken the system.

This example asks a representation-level question. A representation is sufficient for a task when it retains every distinction needed to recover that task's target on a specified support. When the representation also selects actions, can that current certificate justify using it on the states those actions reach? Let $q : X \to Z$ be the representation, $F : X \to Y$ the fixed task or future-response map, and

$$D_q(x) = T\big(x, \pi(q(x))\big) \tag{1}$$

the deployment selected through q. We call sufficiency for the same F on $S_1 = D_q(S_0)$ *deployment closure*. This tests whether the reached representation is still decodable, even allowing a new readout, rather than whether an old prediction head remains accurate.

The gap need not be a capacity deficit. We construct two minimum-cardinality encoders that agree everywhere on S_0 and induce the same deployment, yet have opposite closure outcomes. The minimum exact state budget is unchanged: the distinctions that must be retained have reorganized. Stronger predictive or transition-preserving abstractions already address relevant futures (Littman et al., 2001; Allen et al., 2021; Huang et al., 2022), and performative prediction already studies deployment-induced change (Brown et al., 2022; Mishler & Dalmasso, 2022). Our question is narrower: what remains unestablished by a current task certificate, and what evidence is needed to repair the interface actually used afterward?

Repair introduces a second certification issue. A test may tell the decision maker which two possible worlds it faces, withholding only the realized branch. Evidence that distinguishes those two worlds can then appear sufficient while remaining ambiguous among other worlds with the same

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

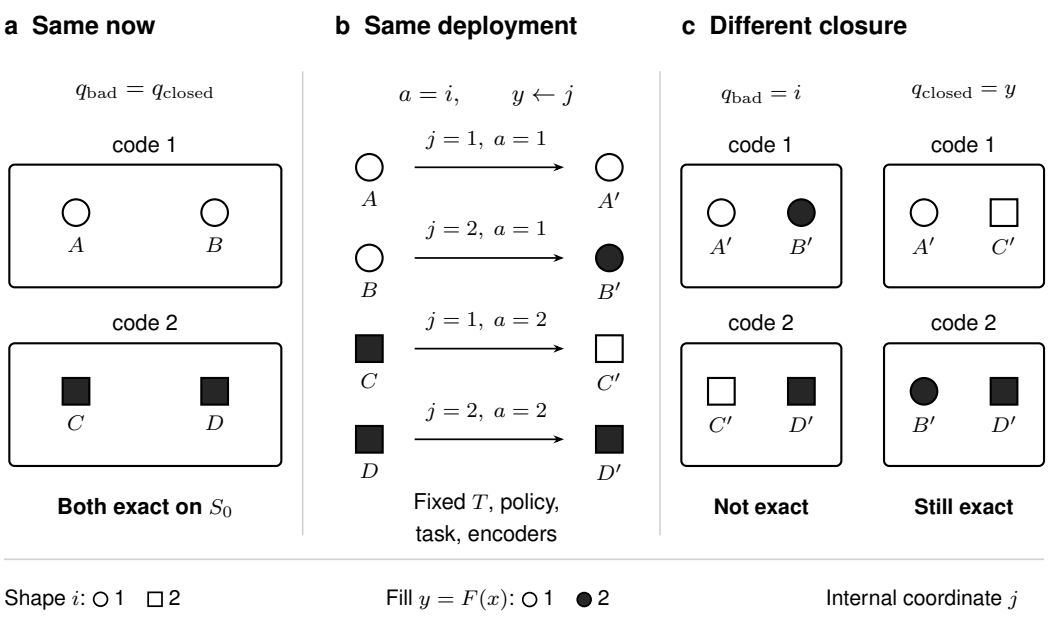

Figure 1: **The same current certificate can conceal different deployment outcomes.** Exact $K = 2$ construction, not empirical data. Shape denotes i, fill denotes the unchanged target $y = F(x)$, and j is an internal coordinate. Initially $y = i$, so the fixed encoders $q_{\mathrm{bad}} = i$ and $q_{\mathrm{closed}} = y$ agree. Both select action $a = i$, and the same transition sets $y \leftarrow j$. On the reached support, q_{bad} merges different target values while q_{closed} remains exact. Primes denote successor states; no encoder is updated.

deployed observation. Exact repair must resolve every target-distinct collision allowed by the retained information, not only the designated pairs. We measure the resulting difference in decision loss on the same market cohort.

Two certification gaps. The *deployment gap* is between current-support sufficiency and sufficiency after representation-selected deployment. The *repair gap* changes the quantifier: one target-distinct collision refutes closure, whereas exact repair must eliminate every such collision allowed by the retained interface. Pair identity or other side information can resolve a designated witness while leaving interface-compatible worlds ambiguous.

The paper makes three connected contributions. First, Theorem 1 gives the fixed-system, equal-budget separation and its support boundary: two equally small encoders can agree on every certified state and induce the same deployment yet have opposite closure outcomes. Second, Section 3 characterizes evidence-constrained repair and the decision cost of unresolved information, including the distinction between known-pair repair and repair of the deployed interface; Appendix A records the separate preservation cost of retaining old distinctions. Third, Section 4 supplies a direct ABIDES mechanism, causal controls, and a randomized evidence study, while Section 5 isolates a distinct learned-system failure in which useful history remains in recurrent state but is not used by the transition predictor. The latter is a transition-interface repair result, not another claim of information loss in the full recurrent representation. A native continuous-control audit is retained as a boundary result in Appendix F.

2 From current sufficiency to deployment closure

2.1 A task-relative property of the reached support

We use deterministic maps and exact sufficiency. Write $\ker(g|_S)$ for the equivalence relation induced by g on S. The representation q is sufficient for F on S_0 when

$$\ker(q|_{S_0}) \subseteq \ker(F|_{S_0}). \tag{2}$$

It is *deployment-closed relative to* (S_0, T, π, F) when the same inclusion holds on $S_1 = D_q(S_0)$. Equivalently, there exists a readout $\bar{F}_1 : q(S_1) \to Y$ such that

$$F \circ D_q = \bar{F}_1 \circ q \circ D_q \quad \text{on } S_0, \qquad \ker(q \circ D_q) \subseteq \ker(F \circ D_q). \tag{3}$$

These are elementary factorization conditions: a target is recoverable exactly when it is constant on representation fibers.

Information preservation is not frozen-readout validity. The readout $\bar{F}_1$ may differ from the current readout. For example, if $q = 0$ on two disjoint supports, $F = 0$ on the first and $F = 1$ on the second, each support is separately decodable but no single readout works on both. A common readout requires sufficiency on $S_0 \cup S_1$; correctness of an already fixed readout requires testing that readout itself. Throughout, deployment closure means preservation of *decodability*, not an unchanged prediction head.

A necessary support boundary. If q is sufficient on S_0 and $D_q(S_0) \subseteq S_0$, then it is automatically deployment-closed. Exact failure therefore requires leaving the certified support. The converse is false: a globally decodable $F = g \circ q$ remains decodable under any support shift. The relevant question is not simply whether deployment leaves S_0, but whether it reaches a target-distinct collision inside a q-fiber.

2.2 Identical current behavior, different closure

Theorem 1 (Current sufficiency does not identify deployment closure)**.** *For every $K \geq 2$, there are fixed K-state representations q_{bad} and q_{closed} that agree pointwise on S_0, are both minimum-cardinality exact for the same F there, and induce the same deployment under a fixed policy. On their common reached support, q_{bad} is not sufficient for F while q_{closed} is. The transition and task are fixed and time-homogeneous.*

Construction. Let $X = [K]^3$, with coordinates (i, j, y), and set

$$S_0 = \{(i, j, y) : y = i\}, \quad F(i, j, y) = y, \quad q_{\mathrm{bad}}(i, j, y) = i, \quad q_{\mathrm{closed}}(i, j, y) = y.$$

Use actions in $[K]$, policy $\pi(z) = z$, and

$$T((i, j, y), a) = \begin{cases} (i, j, j), & a = i, \\ (i, j, y), & a \neq i. \end{cases} \tag{4}$$

On S_0, both encoders equal $F = i$ and select $a = i$. Both deployments reach $S_1 = \{(i, j, y) : y = j\}$. Now $F = j$, so $q_{\mathrm{bad}} = i$ discards the required distinction, whereas $q_{\mathrm{closed}} = y$ retains it. Since F takes K values on each support, K states suffice and are necessary both before and after deployment. Appendix A gives the complete argument.

The theorem is an unrestricted off-support non-identifiability result, not an impossibility under known dynamics or constrained encoder families. Observations confined to S_0 cannot distinguish these two fixed encoders; inspecting their behavior on S_1, their implementations, or a valid transition-level guarantee can. The construction also does not claim that all encoders fail: it explicitly supplies a closed one.

A *direct same-initial-fiber witness* additionally requires that the two initial states share q. Under current sufficiency, such a witness exists exactly when

$$\ker q \ \cap \ \ker(q \circ D_q) \not\subseteq \ker(F \circ D_q) \quad \text{on } S_0. \tag{5}$$

This is stronger than failure of Equation (3): deployment may also merge states from different initial fibers. Our construction and direct market example satisfy the stronger condition.

Relation to transition-level certificates. Suppose $F = \bar{F} \circ q$ and $q(T(x, a)) = \bar{T}(q(x), a)$ hold on a domain containing S_0 and closed under admissible transitions. Then the same domain covers the deployment and its continuation, so current sufficiency and deployment closure follow. The reverse implication fails: one-step task decodability need not preserve transitions or longer futures. Deployment closure fills the gap between a current task certificate and such stronger guarantees; it does not replace them.

3 REPAIR UNDER A DECLARED INFORMATION INTERFACE

Let S_1 be finite and nonempty, q_0 the retained representation there, and $\tau : S_1 \to Y$ the target to recover; in the basic setting, $\tau = F|_{S_1}$. An auxiliary observation e repairs this interface if τ factors through (q_0, e). Both the retained information and the allowed observations must be specified.

3.1 WHICH OBSERVATIONS SUFFICE?

Proposition 2 (Evidence required for exact repair)**.** *Define*

$$m^*(q_0, \tau) = \max_{z \in q_0(S_1)} \left|\tau\left(q_0^{-1}(z) \cap S_1\right)\right|. \tag{6}$$

Among unrestricted finite evidence maps $e : S_1 \to E$, the smallest alphabet allowing exact repair has $|E| = m^*$, or $\lceil \log_2 m^* \rceil$ bits for a fixed-length binary label. For a specified catalog $\{e_1, \ldots, e_M\}$ with costs $c_j \geq 0$, let

$$\mathcal{C} = \{\{s, s'\} \subseteq S_1 : q_0(s) = q_0(s'),\ \tau(s) \neq \tau(s')\},$$

and let $\mathcal{C}_j$ be the pairs separated by e_j. Then exact repair is possible with a subset J precisely when $\bigcup_{j \in J} \mathcal{C}_j = \mathcal{C}$, and its minimum catalog cost is

$$C^*_{\mathcal{E}} = \min_{J : \bigcup_{j \in J} \mathcal{C}_j = \mathcal{C}} \sum_{j \in J} c_j, \tag{7}$$

with $C^*_{\mathcal{E}} = \infty$ if no subset separates all collisions.

The unrestricted label can be designed with full knowledge of the target partition; it is an information lower bound, not necessarily a physically available sensor. The catalog restriction supplies that operational constraint. Its covering form is the standard minimum test collection problem (Halldórsson et al., 2001). Crucially, $\mathcal{C}$ contains *all* target-distinct state pairs inside a retained fiber, not only the pairs selected for an experiment. Thus the quantifier changes at repair time: the existence of one pair in $\mathcal{C}$ witnesses failure, whereas exact repair requires separation of every pair in $\mathcal{C}$.

Preserving every old distinction imposes a different cost. If the retained partition and the coarsest exact target partition each have K cells, their common refinement has at most K^2 cells, and the construction in Theorem 1 attains this bound even though a fresh exact representation uses only K states on either support. This is a preservation constraint rather than intrinsic task complexity; Appendix A states and proves the result and contrasts it with auxiliary evidence.

3.2 DECISION COST AND PAIR CONTEXT

Exact collisions identify an information deficit, but their cost depends on the decision. For the market audit, let $Y \in \mathbb{R}^4$ be the future dealer inventory in shares and $h \in \mathbb{R}^4$ an offline hedge. Use

$$L(Y, h) = \frac{\|Y + h\|_2^2 + \lambda \|h\|_2^2}{100^2}, \qquad \lambda = 1. \tag{8}$$

The 100-share scale makes loss dimensionless. For a declared information set $\mathcal{I}$, decisions must be $\mathcal{I}$-measurable. Define excess loss relative to the full-inventory optimum $h^*(Y) = -Y/(1 + \lambda)$.

Lemma 3 (Information-set-dependent excess loss)**.** *For square-integrable Y and $\lambda \geq 0$, the optimal admissible hedge and its expected excess loss are*

$$h^*_{\mathcal{I}} = -\frac{\mathbb{E}[Y \mid \mathcal{I}]}{1 + \lambda}, \qquad R(\mathcal{I}) = \frac{\mathbb{E}\,\|Y - \mathbb{E}[Y \mid \mathcal{I}]\|_2^2}{(1 + \lambda)100^2}. \tag{9}$$

For nested information sets $\mathcal{I} \subseteq \mathcal{J}$,

$$R(\mathcal{I}) - R(\mathcal{J}) = \frac{\mathbb{E}\,\|\mathbb{E}[Y \mid \mathcal{J}] - \mathbb{E}[Y \mid \mathcal{I}]\|_2^2}{(1 + \lambda)100^2} \geq 0. \tag{10}$$

Table 1: **A direct same-task deployment-closure witness.** The same public interface selects the same deployment; the unchanged future program F differs only afterward. Depths are in shares.

Recorded quantity	World A	World B
Public depths before deployment	$(1600, 1600)$	$(1600, 1600)$
Response to F without deployment	$(580, 220)$	$(580, 220)$
Public depths after deployment	$(1600, 1200)$	$(1600, 1200)$
Response to F after deployment	$(700, 100)$	$(685, 130)$
Transferred-holdings equalized	$(700, 100)$	$(700, 100)$
Inventory-insensitive successors	$(800, 800)$	$(800, 800)$

This conditional-variance identity follows by completing the square; monotonicity with more information is classical (Blackwell, 1953). On a finite support with positive mass at every state, zero risk is equivalent to exact recovery of the inventory target. This separates information loss from the additional error of a fitted predictor that does not attain the information-constrained optimum.

If P identifies a known matched pair but not its realized branch, it changes the decision information:

$$R\big(\sigma(q_0, e, P)\big) \leq R\big(\sigma(q_0, e)\big).$$

Under an equal prior on that pair, its unresolved excess loss is

$$\Gamma_P = \frac{\|y_A - y_B\|_2^2}{4(1+\lambda)100^2}. \tag{11}$$

Once e distinguishes the two branches, this pair's uncertainty is gone. Without P, other states can remain ambiguous.

Corollary 4 (Pair-conditioned repair does not certify interface repair)**.** *Suppose the evaluated states are partitioned into designated two-state pairs, each pair lies within one retained q_0-fiber, and P reveals the pair label but not the realized branch. If evidence e separates the two states in every designated pair, then τ is decodable from (q_0, e, P) on those evaluated states. This does not imply that τ is decodable from (q_0, e): exact interface repair still requires separation of every target-distinct collision in $\mathcal{C}$.*

The corollary is an information-contract statement, not a new theorem that side information improves decisions. Pair context is legitimate when it is part of the declared decision information; the certification point is that retaining P changes the interface being certified. Section 4 measures both information restrictions on the same archived cohort.

4 Controlled market evidence

We use ABIDES, an event-driven multi-agent market simulator (Byrd et al., 2020), to connect deployment closure to a concrete omitted distinction: order ownership. The direct witness keeps the future-response program fixed; the repair study then makes the decision information set explicit.

4.1 Equal public books, different consequences after trading

The deployment controller reads public level-two book information q but not dealer ownership, order identity, or inventory. Displayed ask depth of at least 1,600 shares selects a 400-share buy. Matched worlds have identical public books but different owner orderings; deployment consumes an owner-balanced prefix, preserving equality of the public books while moving different ownership patterns to the queue front.

For the response witness, F executes another 400-share buy, transfers only resulting holdings into a fresh inventory-sensitive successor market, and returns its public bid and ask depths. Table 1 shows one recorded pair. Before deployment the same program gives equal responses; after deployment it separates worlds that the public interface still merges. Equalizing transferred holdings or removing successor inventory sensitivity eliminates the split.

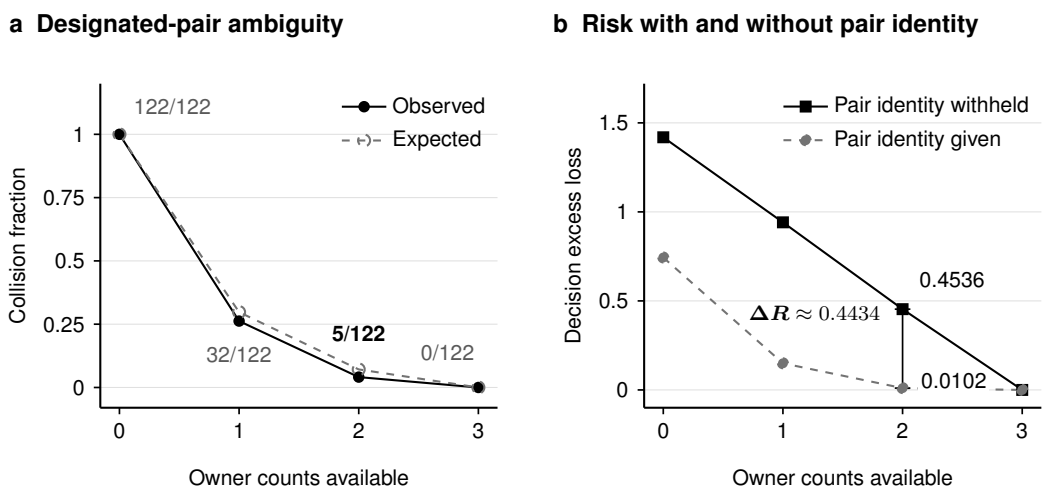

Figure 2: **Evidence value depends on the decision information set. (a)** The prospective endpoint counts unresolved target-distinct designated pairs; the dashed curve is the precomputed generator expectation. **(b)** Optimal finite-distribution risks with pair identity, $R(\sigma(I_k, P))$, and without it, $R(\sigma(I_k))$, use the same 244 arms. At two counts, the bracket marks the additional risk from withholding pair identity ($\Delta R \simeq 0.443$). The no-pair curve is a retrospective reanalysis, not a second prospective endpoint.

The same trace yields nonzero inventory decision cost: after deployment and the fixed future buy, the two inventories give $\Gamma_P = 0.25$ in Equation (11). A separate eight-pair decision bank produces excess losses $0, .25, .50, .75$ across four designed ownership contrasts; no-deployment and score-input-equalization controls are zero. Two additional finite operator banks show increasing response-law separation with ownership contrast while no-deployment responses remain equal (Appendix B). These are controlled mechanism checks, not market-prevalence estimates.

4.2 Additional observations: with and without pair identity

We next sample 128 pairs from a predeclared symmetric owner-sequence generator, without target-class, collision, or loss balancing. The 256 native outcomes contain 31 distinct inventory targets; 122 designated pairs are target-distinct and six target-equal. Let $c_{10}, \ldots, c_{13}$ count the first eight packets by dealer and evaluate

$$I_0 = q, \quad I_1 = (q, c_{10}), \quad I_2 = (q, c_{10}, c_{11}), \quad I_3 = (q, c_{10}, c_{11}, c_{12}).$$

The counts are privileged owner-attributed history, not public level-two signals.

Known-pair endpoint. When the decision maker knows the designated pair P but not its branch, remaining target-distinct pairs number $122, 32, 5, 0$ from zero through three counts; mean excess loss falls $0.74385, 0.14959, 0.01025, 0$. Separating a designated pair identifies its branch under this information contract but need not identify the target among all interface-compatible states.

Interface risk without pair identity. Regrouping the same 244 positive-pair arms by $(q, c_{10}, \ldots)$ without pair identifiers gives optimal risks $1.41902, 0.94130, 0.45363, 0$. With two counts only five designated pairs remain unresolved, yet 1,953 target-distinct state pairs still collide within global interface fibers. The difference is informational, not an estimation discrepancy. This reanalysis reuses archived native outcomes (Appendix C).

Exactness at three counts follows from $c_{13} = 8 - c_{10} - c_{11} - c_{12}$ and exhaustive inspection of the realized public-q fiber: no smaller subset of the three declared count channels repairs all 31 target classes. An unrestricted auxiliary label requires at least $\lceil \log_2 31 \rceil = 5$ bits, a different cost unit. Thus the result is mechanism-, task-, and catalog-relative, not a universal three-signal rule.

Recovery from richer history. A small GRU trained on owner-attributed placement history recovers at least 99.8% of pair-conditioned public-interface excess loss across the tested positive pairs, but a retrospective audit finds familiar eight-token inputs and target profiles in the test set. We therefore use it only as a recoverability check; architecture, overlap, and masked-input controls are in Appendix D.

5 Retained information and learned transitions

The market repair adds information missing from a public interface. A different diagnosis is needed when useful information remains in recurrent state but the predictor does not exploit it. We test that distinction in PayloadLatch, a controlled synthetic dynamical system, using a compact Gaussian recurrent state-space model (RSSM; Hafner et al., 2020). This is not a robotics benchmark or a full Dreamer agent.

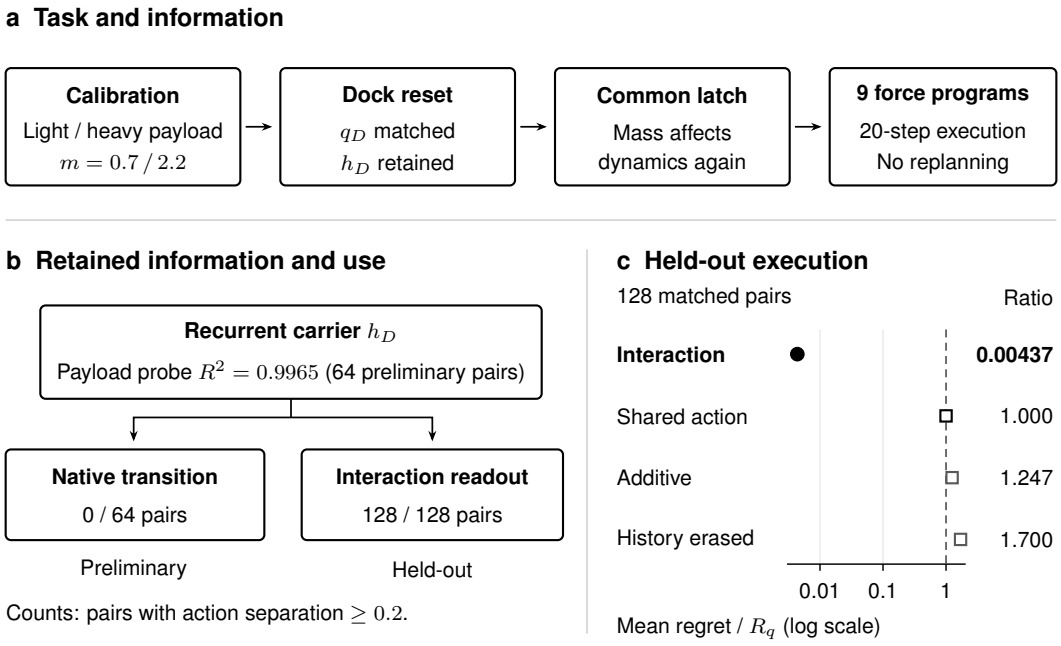

Figure 3: **Retained history can fail at the transition interface.** (a) A dock reset matches visible states while preserving recurrent history; a common latch makes mass relevant again. (b) Arrows track history use; current-state and action inputs are omitted. The probe and native result use 64 preliminary pairs; the interaction readout uses 128 held-out pairs. Counts denote selected-force gaps of at least 0.2. (c) Mean simulator regret is normalized by R_q, the mean branch regret of the pairwise minimax shared program. History erasure reverses the gain. All 256 directed carrier swaps follow the donor action and increase recipient cost. Recurrent weights remain fixed.

Task and information. Hidden payload mass affects a short calibration interaction. A dock reset removes the physical trace while recurrent memory is retained. Before latching, dynamics no longer depend on mass; a common latch action makes it relevant again without separating the paired visible observations. The controller then chooses one of nine 20-step force programs and executes it without feedback replanning. Regret is measured in the simulator relative to the best program in the same bank for the realized state and mass. A shared-action benchmark knows the pair's two possible dynamics, not the realized mass, and chooses a program minimizing worst-case branch regret. We report its mean branch regret R_q; it is not the minimum mean-risk benchmark or a lower bound for feedback controllers.

Preliminary diagnosis. On 64 preliminary pairs, a probe decodes mass from the frozen recurrent state ($R^2 \simeq 0.9965$), but the native transition model does not separate light/heavy continuation amplitudes by the specified 0.2 threshold. Its mean regret is $1.571R_q$. Adding a 20-step prior-rollout training loss also fails this endpoint ($1.688R_q$). These results motivate testing the transition predictor rather than assuming the carrier has lost the information.

Held-out transition repair. Holding the recurrent model's weights fixed, we fit a ridge predictor for $q_{t+1} - q_t$ using features $[q_t, a_t, h_D, h_D u_t, h_D v_t]$, where h_D is the deterministic recurrent vector at deployment, u_t is force, and v_t is velocity. A comparison uses only the additive features $[q_t, a_t, h_D]$. Both fits use the original 1,600 training trajectories, without mass labels. The checkpoint, feature map, fitting rule, action bank, and evaluation criteria are fixed before testing 128 disjoint paired histories. The predicted observation is fed back through each imagined program while h_D remains fixed; the selected program is then scored in the simulator.

Figure 3c summarizes the held-out decision results. The history-interaction readout separates the selected light/heavy actions in all 128 pairs and reduces mean execution regret from 0.17897 for the pairwise minimax shared action to 0.000782 (ratio 0.00437). The additive readout gives mean regret 0.22320. One-step transition prediction has $R^2 = 0.9975$ and recovers 0.9959 of the true response-contrast norm (Appendix E).

Erasing pre-deployment history without refitting removes separation and raises regret above the shared-action benchmark. In all 256 directed within-pair carrier swaps, replacing only the recipient h_D with its matched donor's carrier moves the decision to the donor-native action and increases recipient execution cost. Donor-following alone is an input-consistency check for a deterministic planner; the additional result is the adverse cost in the recipient dynamics.

The repair uses retained history rather than demonstrating lost decodability. The result is limited to one fixed model and familiar masses; it does not establish minimality of the feature map, general failure of Dreamer-style agents, or exact sufficiency of a continuous recurrent state. A separate native Cartpole audit is reported only as a boundary result in Appendix F because its pre-specified strong-effect criteria are not met.

6 Related work

State abstraction and sufficiency for control. Predictive state representations, causal states, Markov abstractions, and action-sufficient representations preserve transition or future structure needed for control (Littman et al., 2001; Shalizi & Crutchfield, 2001; Allen et al., 2021; Rakelly et al., 2021; Huang et al., 2022; Gu et al., 2025). Deep SPI connects transition and reward prediction errors, representation quality, and restricted policy updates to improvement guarantees (Delgrange et al., 2026). Reusable consequence state and predictive closure likewise make future-test scope and lawful updates explicit (Takahashi, 2026; Murray, 2026). Such guarantees exclude our failure when they actually cover the declared task and reached support. Our starting certificate is weaker: exact recovery of one fixed task on S_0. Theorem 1 identifies what current-support behavior alone leaves undecided even at equal minimum capacity and under the same deployment.

Deployment-induced instability. Performative prediction and reinforcement learning study how deployment changes the data or environment (Perdomo et al., 2020; Brown et al., 2022; Mandal et al., 2023; Rank et al., 2024). Observable properties such as fairness can likewise hold before deployment and fail afterward (Mishler & Dalmasso, 2022). Thus instability of a certified property is not itself new, and fixing T is not a separation from stateful performativity. Our direct witness concerns a representation partition: initial q- and F-agreement, the same representation-selected deployment, continued q-agreement, and disagreement under the unchanged F. The equal-budget construction further separates a change in required distinctions from an increase in minimum state capacity.

Information available to a decision. The value of more informative experiments is classical (Blackwell, 1953). Bayes-sufficient representations make dependence on a fixed distribution and loss explicit (Sevetlidis, 2026a); decision-sufficient linear-optimization datasets distinguish pointwise from global sufficiency (Ye et al., 2026). Our pair-context comparison is not the claim that more information helps. It measures how a repair conclusion changes when designated-pair identity,

available in the evaluation, is absent from the deployed interface. The catalog characterization uses standard test-collection machinery (Halldórsson et al., 2001); neither it nor the partition-refinement identity is a new optimization algorithm.

Retained state versus observed behavior. Predictor behavior need not identify properties of its representation-head factorization (Sevetlidis, 2026b). Future-learning probes can likewise distinguish execution states with matching current behavior (Wang, 2026). We study a fixed representation under environment-directed deployment, not hidden optimizer-state evolution. The payload experiment adds a specific diagnostic: after training, useful history can remain decodable yet fail to affect imagined control. Its intervention-supported repair changes the transition predictor while leaving the recurrent carrier fixed.

7 Implications and limitations

What a deployment audit must establish. Declare the task, support, representation, policy, and transition before testing the reached interface $q \circ D_q$ against $F \circ D_q$. A direct matched-pair witness also needs initial q- and F-agreement. Repair evaluation must name every information source, including pair context or privileged records. On a finite known support, the catalog criterion gives an exact test; on an unknown or continuous support, these results do not supply a statistically complete certification algorithm. Testing decodability and testing a fitted predictor answer different questions.

Scope and a native-control boundary. The theory is exact, deterministic, and task-relative. The market demonstrates a controlled ownership mechanism, not its prevalence in natural markets; hedging is offline, and the no-pair risk is an empirical-distribution optimum. The three-count threshold is catalog-specific. The payload result concerns one fixed model and an open-loop program bank, not general Dreamer performance or minimal repair. A separate native Cartpole audit shows increased finite-interface regret and held-out benefit from an added observation channel, but none of three frozen models reaches the pre-specified strong-effect thresholds. We retain it as a boundary result (Appendix F), not a successful confirmation or evidence of worse absolute task performance. No experiment tests the K^2 preservation bound.

Conclusion. A current task certificate need not cover the states reached by using the representation. The required distinctions can change without increasing the minimum state budget. Repair may require additional evidence, or a predictor that better uses retained information; its effectiveness must be evaluated under the information available at deployment rather than the richer context of a selected test pair.

AI use statement

Generative AI tools assisted throughout the project, including ideation, literature search, experimental design, code drafting and debugging, figure preparation, and manuscript editing. Mathematical claims, experimental results, and numerical analyses were checked against explicit derivations, executed code, archived records, and source materials. Retrospective analyses are identified separately from prospectively specified evaluations. The authors take responsibility for the final text, proofs, code, figures, and reported results.

Reproducibility statement

Proofs appear in Appendix A; market protocols in Appendices B–C; learned recovery is in Appendix D; learned transition-interface and native-control methods are in Appendices E–F. Appendix G documents evidence status. The supplement provides archived data, weights, analysis code, vector figures, and manifests. The learned-state evaluations keep recurrent model weights fixed. The Cartpole result is reported with its originally specified effect-size criterion unchanged. Market reanalysis uses archived simulator outputs rather than rerunning the simulator.

REFERENCES

Cameron Allen, Neev Parikh, Omer Gottesman, and George Konidaris. Learning Markov state abstractions for deep reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 34, pp. 8229–8241, 2021.

David Blackwell. Equivalent comparisons of experiments. *The Annals of Mathematical Statistics*, 24 (2):265–272, 1953. doi: 10.1214/aoms/1177729032.

Gavin Brown, Shlomi Hod, and Iden Kalemaj. Performative prediction in a stateful world. In *Proceedings of the 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pp. 6045–6061, 2022.

David Byrd, Maria Hybinette, and Tucker Hybinette Balch. ABIDES: Towards high-fidelity multi-agent market simulation. In *Proceedings of the SIGSIM Principles of Advanced Discrete Simulation*, pp. 11–22, 2020. doi: 10.1145/3384441.3395986.

Florent Delgrange, Raphaël Avalos, and Willem Röpke. Deep SPI: Safe policy improvement via world models. In *International Conference on Learning Representations*, 2026.

Yuliang Gu, Hongpeng Cao, Marco Caccamo, and Naira Hovakimyan. Observations meet actions: Learning control-sufficient representations for robust policy generalization. *arXiv preprint arXiv:2507.19437*, 2025.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=S1lOTC4tDS.

Bjarni V. Halldórsson, Magnús M. Halldórsson, and Ramamoorthi Ravi. On the approximability of the minimum test collection problem. In *Algorithms – ESA 2001*, volume 2161 of *Lecture Notes in Computer Science*, pp. 158–169, 2001. doi: 10.1007/3-540-44676-1_13.

Biwei Huang, Chaochao Lu, Liu Leqi, José Miguel Hernández-Lobato, Clark Glymour, Bernhard Schölkopf, and Kun Zhang. Action-sufficient state representation learning for control with structural constraints. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 9260–9279, 2022.

Michael L. Littman, Richard S. Sutton, and Satinder Singh. Predictive representations of state. In *Advances in Neural Information Processing Systems 14*, 2001.

Debmalya Mandal, Stelios Triantafyllou, and Goran Radanovic. Performative reinforcement learning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 23642–23680, 2023.

Alan Mishler and Niccolò Dalmasso. Fair when trained, unfair when deployed: Observable fairness measures are unstable in performative prediction settings. *arXiv preprint arXiv:2202.05049*, 2022.

Daniel John Murray. Predictive closure: State, action, and the experimental compression of history, 2026. SSRN 7427098.

Juan Perdomo, Tijana Zrnic, Celestine Mendler-Dünner, and Moritz Hardt. Performative prediction. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 7599–7609, 2020.

Kate Rakelly, Abhishek Gupta, Carlos Florensa, and Sergey Levine. Which mutual-information representation learning objectives are sufficient for control? In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Ben Rank, Stelios Triantafyllou, Debmalya Mandal, and Goran Radanovic. Performative reinforcement learning in gradually shifting environments. In *Proceedings of the Fortieth Conference on Uncertainty in Artificial Intelligence*, volume 244 of *Proceedings of Machine Learning Research*, pp. 3041–3075, 2024.

Vasileios Sevetlidis. Bayes-sufficient representations in supervised learning. *arXiv preprint arXiv:2606.04045*, 2026a.

Vasileios Sevetlidis. A fiber criterion for representation identifiability in supervised learning. *arXiv preprint arXiv:2606.01092*, 2026b.

Cosma Rohilla Shalizi and James P. Crutchfield. Computational mechanics: Pattern and prediction, structure and simplicity. *Journal of Statistical Physics*, 104:819–881, 2001. doi: 10.1023/A:1010388907793.

K. Takahashi. Reusable consequence states under partial support and model uncertainty, 2026. Zenodo, doi:10.5281/zenodo.22170023; SSRN 7372918.

Qinyou Wang. Fiber fingerprints of hidden learning-state dynamics. *arXiv preprint arXiv:2608.15976*, 2026.

Yuhan Ye, Saurabh Amin, and Asuman Özdağlar. Learning decision-sufficient representations for linear optimization. In *Proceedings of the Thirty Ninth Conference on Learning Theory*, volume 336 of *Proceedings of Machine Learning Research*, pp. 6924–6975, 2026.

A Proofs and distinctions between certificates

A.1 Factorization and the support boundary

A map F factors through q on a set S if and only if F is constant on every nonempty fiber of $q|_S$. Necessity follows from $q(s) = q(s') \Rightarrow g(q(s)) = g(q(s'))$. For sufficiency, define $g(z)$ to be the common F-value on the fiber $q^{-1}(z) \cap S$. Applying this observation to $S_1 = D_q(S_0)$ yields Equation (3). Pulling back $s = D_q(x)$ and $s' = D_q(x')$ yields the equivalent kernel inclusion on S_0.

If $S_1 \subseteq S_0$, every pair in S_1 already satisfies the sufficiency implication on S_0. Hence forward invariance preserves exact sufficiency. Conversely, $F = g \circ q$ on all of X makes closure hold for any reached support, including supports outside S_0. Support exit is necessary for exact failure under the stated assumptions, but is not sufficient.

The direct witness in Equation (5) asks for a pair in both $\ker q$ and $\ker(q \circ D_q)$ but not $\ker(F \circ D_q)$. Current sufficiency supplies the initial F-equality. General closure failure omits the additional requirement of initial q-equality.

A.2 Proof of Theorem 1

For the construction in Equation (4), $S_0 = \{(i, j, i) : i, j \in [K]\}$. On this support, $q_{\text{bad}} = q_{\text{closed}} = F = i$. Both encoders therefore have K values, are exact, and select action i under the same policy. Since F takes K distinct values, no exact current representation can have fewer than K values.

The deployed transition maps (i, j, i) to (i, j, j), so both encoders induce the same reached support

$$S_1 = \{(i, j, j) : i, j \in [K]\}.$$

On S_1, $q_{\text{closed}} = y = F$ and is exact. For any fixed i and distinct j, j', $q_{\text{bad}}(i, j, j) = q_{\text{bad}}(i, j', j') = i$, whereas the respective targets are j and j'. Thus q_{bad} fails closure and supplies a direct witness. The transition, encoders, policy, and target are fixed; the change is which states are reached. Again F takes K values on S_1, and the representation $q_1(i, j, j) = j$ is exact with K values. This also proves equality of the minimum fresh state budgets.

Any procedure receiving only the encoders' current-support input-output behavior receives identical information in these two cases. It cannot infer different closure outcomes from that information alone. The argument does not restrict a procedure that can inspect off-support encoder behavior or exploit additional valid structural assumptions.

A.3 Decodability, a common readout, and transition preservation

Deployment closure allows a new readout. Let $S_0 = \{u\}$ and $S_1 = \{v\}$, with $q(u) = q(v) = 0$, $F(u) = 0$, and $F(v) = 1$. Each support is separately decodable, but their union is not. Therefore a current readout need not remain correct even when closure holds. A common readout exists precisely when $\ker(q|_{S_0 \cup S_1}) \subseteq \ker(F|_{S_0 \cup S_1})$. For an already fixed prediction head, its actual values must additionally be checked on the reached support.

A transition-preserving abstraction is stronger than task-relative closure. For example, take F constant, let $q(a) = q(b) = 0$ and $q(c) = 1$, and use a single action with $T(a) = a$ and $T(b) = c$. The target is globally decodable, but no abstract transition can map the common code of a and b to both 0 and 1.

One-step closure also need not imply closure at a longer horizon. Let $a \mapsto c \mapsto e$ and $b \mapsto d \mapsto f$ under one fixed action, with absorbing e, f. Set $q = 0$ on all six states, and $F = 0$ except $F(f) = 1$. From $S_0 = \{a, b\}$, the target is decodable both currently and after one deployment, but not after two. The maps are time-homogeneous and F is unchanged. A certificate for longer reuse must explicitly cover the corresponding reached supports.

A.4 Proof of Proposition 2

Fix a retained code z and its fiber $B_z = q_0^{-1}(z) \cap S_1$. If τ takes r_z values on B_z, any repairing evidence map must separate those r_z target classes within that fiber. Otherwise two states would have identical (q_0, e) and different targets. Thus $|E| \geq r_z$ for every z, and $|E| \geq m^*$.

For achievability, enumerate the target classes separately inside each fiber and assign labels from $\{1, \dots, m^*\}$. Labels can be reused across distinct retained-code fibers because q_0 identifies the fiber. The joint code (q_0, e) identifies the local target class, and a decoder therefore exists. A fixed-length binary label needs exactly $\lceil \log_2 m^* \rceil$ bits.

For the restricted catalog, a selected family e_J permits exact decoding if and only if every target-distinct pair sharing q_0 differs on at least one selected observation. Equivalently, $\bigcup_{j \in J} \mathcal{C}_j = \mathcal{C}$. Minimizing the additive cost gives Equation (7). If some collision is separated by no catalog observation, exact repair is impossible under that catalog. This is the standard test-collection formulation, applied to deployment-generated target collisions.

A.5 Proof of Corollary 4

The positive statement is immediate: once P identifies a two-state designated pair, an e-value that separates its two branches identifies the target within that pair.

The converse implication fails in general. Let $S_1 = \{a, b, c, d\}$, let q_0 be constant, and define two designated pairs $\{a, b\}$ and $\{c, d\}$. Set

$$\tau(a) = \tau(c) = 0, \qquad \tau(b) = \tau(d) = 1,$$

and choose evidence

$$e(a) = e(d) = 0, \qquad e(b) = e(c) = 1.$$

Within each designated pair, e separates the two branches, so τ factors through (q_0, e, P). Without P, however, a and d share the same (q_0, e) value but have different targets, as do b and c. Hence τ does not factor through (q_0, e). Proposition 2 gives the general criterion: interface repair requires separation of all pairs in $\mathcal{C}$, not only the designated pairs.

A.6 Repair while preserving old distinctions

Proposition 5 (Preservation-constrained repair)**.** *Let $\mathcal{P}_0$ be the retained partition and $\mathcal{P}_1$ the coarsest exact target partition. Their coarsest common refinement $\mathcal{P}_0 \wedge \mathcal{P}_1$ is the coarsest exact repair preserving every old distinction. If $|\mathcal{P}_0| = |\mathcal{P}_1| = K$, the refinement has at most K^2 cells. The construction in Theorem 1 attains this bound although a fresh exact representation uses K states on either support.*

A repair preserving every old distinction must refine $\mathcal{P}_0$. A repair exact for the target must refine $\mathcal{P}_1$. Consequently every admissible repair refines their coarsest common refinement, and that common refinement is itself admissible. Its cells are the nonempty intersections of one old cell and one target cell, so there are at most K^2 when each partition has K cells.

On S_1 in Theorem 1, the old code groups states by i and the exact target groups them by j. All K^2 intersections are nonempty singletons, attaining the bound. A fresh target code uses only K states; preserving both the old row and new column distinctions uses K^2. The new code cannot be a deterministic function of the old one, because each fixed old row contains all K required target values.

A.7 Proof of Lemma 3

Completing the square gives

$$L(Y, h) = \frac{1}{100^2} \left[\frac{\lambda}{1+\lambda} \|Y\|_2^2 + (1+\lambda) \left\| h + \frac{Y}{1+\lambda} \right\|_2^2 \right].$$

Conditioning on $\mathcal{I}$, the minimizing measurable hedge is $h^*_{\mathcal{I}} = -\mathbb{E}[Y \mid \mathcal{I}]/(1+\lambda)$. Relative to $h^*(Y)$, its expected excess loss is the residual conditional variance in Equation (9).

For $\mathcal{I} \subseteq \mathcal{J}$, write

$$Y - \mathbb{E}[Y \mid \mathcal{I}] = Y - \mathbb{E}[Y \mid \mathcal{J}] + \mathbb{E}[Y \mid \mathcal{J}] - \mathbb{E}[Y \mid \mathcal{I}].$$

The two terms on the right are orthogonal in L^2 by the conditional-expectation property. Squaring, taking expectations, and dividing by $(1+\lambda)100^2$ proves Equation (10).

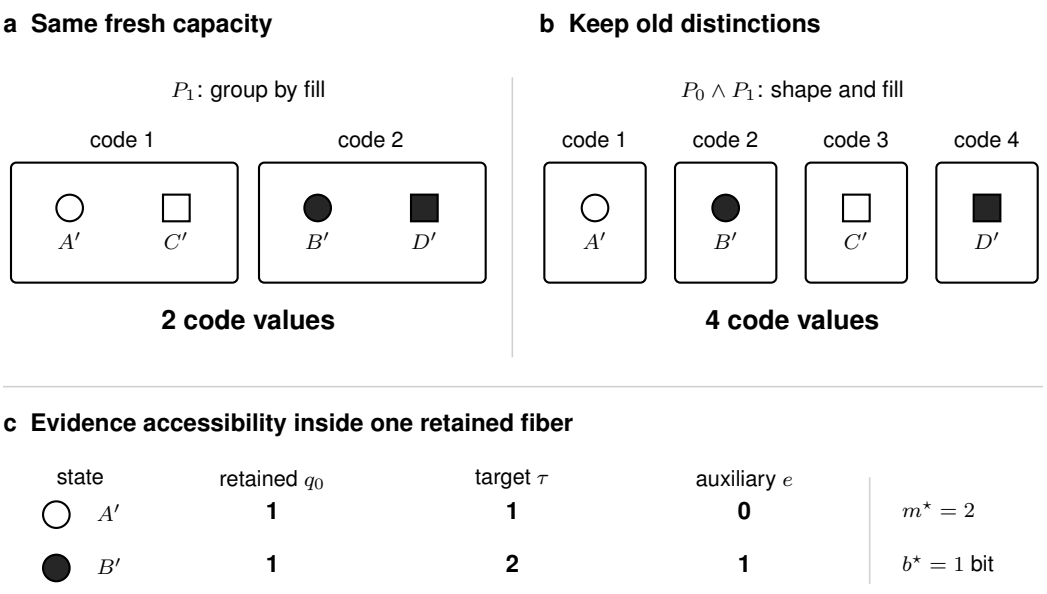

Figure 4: **Three repair quantities in the exact construction.** Regrouping by the new target preserves the fresh K-state budget; retaining old and new distinctions can require the common refinement; auxiliary evidence asks what must become observable inside a retained fiber.

For a known pair with equal branch probabilities, the conditional mean is $(y_A + y_B)/2$. Its mean squared residual is $\|y_A - y_B\|_2^2/4$, giving Equation (11). If an observation identifies the branch, this residual is zero. On a finite distribution assigning positive mass to every state, zero residual conditional variance is equivalent to the target being constant on each information fiber, which is precisely exact repair.

B Market interfaces, perturbations, and decision protocol

B.1 Observation and response timing

The public observation is the saved level-two book dictionary: bid and ask price/depth pairs and the last-trade field. The deployment policy uses that dictionary only. In the recorded witness, the public bid and ask prices are 99,990 and 100,000 integer cents; the saved last-trade field is null in both worlds. Full public dictionaries agree between the arms at each comparison. They need not remain numerically unchanged over time: depth changes from $(1600, 1600)$ before deployment to $(1600, 1200)$ afterward.

The fixed future-response program executes an independent 400-share buy, transfers only holdings into a fresh kernel containing four inventory-sensitive successor agents, and queries the resulting public depth pair. In dealer-role order $(10, 11, 12, 13)$, inventories after deployment and the subsequent buy are $(-200, -200, -200, -200)$ in world A and $(-100, -300, -200, -200)$ in world B. Their successor responses are $(700, 100)$ and $(685, 130)$. Without deployment, both inventories are $(-100, -100, -100, -100)$ and both responses are $(580, 220)$.

Equalizing the transferred holdings produces $(700, 100)$ in both worlds. Disabling successor inventory sensitivity produces $(800, 800)$ in both. These are interventions on the successor mechanism. They differ from the decision benchmark's scoring-input equalization, which only replaces one offline inventory vector with the other and does not execute a repaired market.

The direct response witness and the later inventory audit use different readouts. The former includes the successor kernel and reports public liquidity; the latter reports the inventory after the fixed probe. The before/after comparison keeps its own declared readout unchanged. We do not treat equality of these two different readout definitions as an experimental control.

B.2 Finite perturbation families

For a class of matched ownership contrasts, let $c_A(r)$ and $c_B(r)$ count the successor depth responses r in N equally weighted trials. The reported marginal total variation is

$$\mathrm{TV}(P_A, P_B) = \frac{1}{2N} \sum_r |c_A(r) - c_B(r)|.$$

Paired disagreement is instead $N^{-1} \sum_{k=1}^{N} \mathbf{1}\{R_A(k) \neq R_B(k)\}$. Pairing the nuisance inputs does not make these estimands identical.

The finite-shock study enumerates five common shocks (sell 200, sell 100, no order, buy 100, buy 200 shares) crossed with three successor per-side activity levels (300, 400, 500). Every one of the 15 conditions has weight $1/15$, so total variation is exact for this finite future-response operator. A common public audit after deployment precedes the nuisance event. The no-deployment control retains the same future operator.

The native-noise study uses 24 preselected paths, each with ten native noise agents of order size 20 shares, and identical 200-nanosecond deterministic message latency in the two arms. Each path has weight $1/24$. Its total variation is an empirical marginal-response statistic for that path collection, not a population-wide exact law or a test of random latency. The ten agents are not independent denominator units.

Table 2: **Response differences under the two perturbation families.** Contrast levels are designed queue-ownership differences. All no-deployment total variations are zero.

	Finite shocks ($N = 15$)		Native noise ($N = 24$)	
Contrast	TV	Paired difference	TV	Paired difference
None	0	$0/15$	0	$0/24$
Small	$3/5$	$9/15$	$7/12$	$14/24$
Medium	$4/5$	$12/15$	$3/4$	$20/24$
Large	1	$15/15$	1	$24/24$

For the medium native-noise contrast, the absolute marginal count differences sum to 36, so TV is $36/(2 \cdot 24) = 3/4$; paired disagreement is $20/24 = 5/6$. The stochastic comparisons extend the deterministic witness by holding a response family and its weights fixed, rather than substituting an unannounced future distribution.

B.3 Offline pair-conditioned decision benchmark

The decision study uses eight new queue pairs, two at each designed contrast level, checked under two deterministic kernel seeds. For each branch, the evaluator runs the market, retrieves the post-probe inventory after kernel termination, and then scores a hypothetical hedge. No hedge order is submitted to the market.

The benchmark decision knows the fixed matched pair, its two possible inventories, their equal prior, and the public observation. It does not know which branch is realized or observe the realized terminal inventory at decision time. Since public observations agree within the pair, its hedge must be shared. This gives

$$h_P = -\frac{y_A + y_B}{2(1+\lambda)}, \qquad \Gamma_P = \frac{1}{2} \sum_{b \in \{A,B\}} [L(y_b, h_P) - L(y_b, h^*(y_b))].$$

Equation (11) follows. The four designed contrast levels have excess losses 0, 0.25, 0.50, and 0.75, reproduced across the two variants and integrity seeds. Without deployment, both inventories are balanced and excess loss is zero. Replacing both scoring vectors by the A inventory also gives zero. The use of both potential outcomes to define this benchmark does not grant an online decision access to the realized future inventory, nor does it produce a single public-only rule shared across all distinct pairs.

C Evidence frontiers and the information-set reanalysis

C.1 Prospective randomized queue design

The randomized queue cohort uses seed 104729 and a sample size of 128 matched pairs. The generator, sample size, observation catalog, execution schedule, and original collision and pair-conditioned loss endpoints were specified before cohort generation and native labeling. Each arm has 16 owner tokens and four total packets per dealer. The first four tokens are a random permutation of the dealers. The next four are independent uniform draws conditioned only on not all being the same dealer. The last eight are a shuffled complement completing the dealer totals. Arms A and B are generated independently.

There are $4^4 - 4 = 252$ allowed ordered middle blocks. Their count vectors produce $\binom{7}{3} - 4 = 31$ distinct first-eight target profiles. The realized 256-arm cohort covers all 31. The generator does not stratify on contrast magnitude, target identity, or which observation first separates a pair. The source protocol also excludes exact overlap with its earlier queue cohorts; this does not amount to cross-mechanism generalization.

Every archived native inventory equals -100 times its first-eight owner-count vector. The public dictionaries are identical across all arms before deployment and across all arms after deployment. The two times have different depths. The six target-equal pairs contribute zero pair-conditioned excess loss and are omitted from the original positive-pair mean; they are included in the all-arm sensitivity analysis below.

C.2 Original endpoint: designated pairs and known-pair decisions

Let P identify one of the 122 target-distinct designated pairs, selected uniformly, and choose its arm uniformly. At observation level k, the original residual-loss rule is

$$\Gamma_{P,k} = \begin{cases} \Gamma_P, & I_k(x_A) = I_k(x_B), \\ 0, & I_k(x_A) \neq I_k(x_B). \end{cases}$$

Because the pair is known, separating its two possible observations identifies the target. Averaging this rule is exactly $R(\sigma(I_k, P))$ for the declared empirical distribution. The associated collision count only ranges over the 122 designated pairs, not over every target-distinct pair in the retained public fiber.

Table 3: **Prospectively specified pair-conditioned results.** Observations come from the archived native market labels. Generator expectations enumerate the specified queue distribution; they are not fitted to these outcomes.

Counts added	Designated collisions	Collision fraction (expected)	Pair-conditioned risk (observed)	Expected risk
0	$122/122$	1.00000	0.74385	0.74615
1	$32/122$	0.29767	0.14959	0.17134
2	$5/122$	0.07133	0.01025	0.03040
3	$0/122$	0	0	0

C.3 Retrospective analysis without designated-pair information

This retrospective analysis withholds P from the decision rule. The population for the primary comparison remains the same 244 arms from target-distinct pairs, with uniform arm weights. For each observation group g defined by I_k, compute its mean native inventory $\bar{y}_g$. Equation (9) becomes

$$\widehat{R}(I_k) = \frac{1}{2n\,100^2} \sum_g \sum_{i \in g} \|y_i - \bar{y}_g\|_2^2, \qquad n = 244. \tag{12}$$

This quantity is an exact oracle optimum for the empirical distribution: the conditional means use all its labeled states. It is not a train/test estimate for a learned predictor. Defining this finite distribution

requires the outcome labels, but the admissible decision rule itself uses only I_k, not pair identity or the realized target. This is a retrospective sensitivity analysis, and was not an endpoint in the original pre-specified protocol.

Table 4: **The same positive-pair cohort under two information restrictions.** "All-fiber collisions" counts every unordered pair of the 244 arm states with the same I_k but a different native target. These are not independent observations.

Counts added	Interface groups	With pair identity	Without pair identity	All-fiber collisions
0	1	0.74385	1.41902	28,420
1	4	0.14959	0.94130	8,569
2	13	0.01025	0.45363	1,953
3	31	0	0	0

All-arm sensitivity. With uniform weights on all 256 arms, risks without pair identity are 1.41490, 0.95383, 0.46185, and 0. The corresponding all-fiber collision counts are 31,262; 9,693; 2,227; and 0. Thus inclusion of the six target-equal designated pairs does not remove the information-set distinction. Such a pair can have zero *within-pair* uncertainty while its states still belong to a globally ambiguous interface group.

Exact generator comparison. An additional retrospective enumeration makes the same comparison under the generator distribution, conditional on the two designated targets differing. Let w_a be the number of allowed middle blocks yielding target count vector a, so $\sum_a w_a = 252$. There are

$$Z = \sum_{a \neq b} w_a w_b = 60{,}792$$

ordered target-distinct block pairs. The marginal target weight under this conditioning is $w_a(252 - w_a)/Z$. Grouping this weighted distribution by the first k counts gives risks without pair identity 1.44730, 0.96486, 0.48243, and 0. The original pair-conditioned expectations are those in Table 3. This enumeration concerns the declared generator, not unseen market mechanisms.

C.4 Exact repair of the full observed fiber

For every arm, $c_{10} + c_{11} + c_{12} + c_{13} = 8$. Three recorded counts determine the fourth and therefore the entire inventory vector. This establishes full-fiber exactness independently of the designated-pair analysis. Enumerating all eight subsets of the catalog $\{c_{10}, c_{11}, c_{12}\}$ on the archived labels leaves target-distinct collisions for every subset of size at most two; only the full three-count set has zero risk. Hence its minimum exact unit-cost catalog repair is three on this realized support.

All 31 targets share the same public code, so Proposition 2 requires an unrestricted label alphabet of size 31, or five fixed-length bits. A count channel is not one bit. Moreover, if pair identity is retained, each positive pair's fiber has only two targets and needs only a binary auxiliary label in the unrestricted problem. These statements use different retained information and must not be compared as if they were the same bound.

C.5 Designed calibration cohort

An earlier prospective calibration used 16 deliberately chosen matched pairs, four at each contrast level, to exercise the nested observation catalog. It contained 32 unique arms, all matching the exact inventory/count relation. Among its 12 target-distinct designated pairs, unresolved pair counts were 12, 3, 1, and 0; mean pair-conditioned excess losses were 0.5000, 0.1042, 0.0208, and 0. The public fiber contained 13 targets, requiring four unrestricted label bits. The observation ladder was designed before native outcomes, rather than discovered as a natural-market regularity. The randomized study removes this deliberate contrast and rotation stratification while retaining the same controlled mechanism.

D RECURRENT RECOVERY: METHODS, OVERLAP, AND NUMERICAL CHECKS

D.1 TASK, INFORMATION, AND TARGET UNITS

The learner predicts the four-dealer inventory following a 400-share deployment and a separate 400-share probe. Its input is the first eight owner-attributed placement events in the initial queue. These events precede both trades; the target is not provided as input. Dealer roles $(10, 11, 12, 13)$ map to tokens $(0, 1, 2, 3)$. The model does not receive the public level-two observation. Its access to recorded owner information is richer than that of the deployment controller.

The predicted inventory is expressed in 100-share units: the analytic target for a sequence is the negative first-eight owner-count vector. The native test inventory is divided by 100 before scoring the learned predictor. With this normalized target y, the hedge is $h = -\hat{y}/2$ and its excess loss is $\|\hat{y} - y\|_2^2/2$, equivalent to Equation (8) in shares.

D.2 ARCHITECTURE AND TRAINING RECIPE

An embedding layer maps the four token types to eight-dimensional vectors. A one-layer, 16-hidden-unit GRU processes the eight-token sequence, and a linear head maps its final hidden state to four inventory predictions. The training set contains 6,000 unique full 16-token queues generated with seed 66001. Each starts with a shuffled balanced four-dealer prefix; the remaining multiset contains three packets per dealer and is shuffled. Full sequences in the fixed test cohort are excluded. Training labels are analytic count targets, not inventories generated by native training simulations.

All five model seeds (701–705) use the same training set, mean squared error, AdamW with learning rate 0.003 and weight decay 10^{-5}, batch size 128, and 160 fixed epochs. Batch order is shuffled using the model seed. There is no validation-based early stopping in the archived recipe. The test cohort contains 12 matched pairs, three at each of four designed contrast levels, and was fixed before training and native test labeling. The saved training receipt records that native test labels were absent during training. The native labels, training configuration, original sampling recipe, receipt, and five saved state dictionaries accompany the supplementary reanalysis.

D.3 ORIGINAL RECOVERY RESULTS

For each of the nine target-distinct test pairs, recovery is

$$1 - \frac{\text{mean learned excess loss over its two arms}}{\text{optimal shared-hedge excess loss for that pair}}.$$

The denominator is the known-pair public-interface benchmark, not a fitted global public-only policy. The minimum over the nine pairs and five seeds is 0.99804776. On the three target-equal pairs, the denominator is zero, so recovery ratios are undefined; the maximum seed-mean learned excess loss is 0.0002514922. The exact count predictor incurs zero excess loss on every native label.

At inference, the input-sensitivity control replaces all eight tokens with token zero, which denotes dealer 10. It is not a learned neutral or missing-data token. Because the training prefix always includes every dealer, this constant-token sequence lies outside the training support. The severe loss increase demonstrates sensitivity of the trained predictor to this corruption; it does not show that all owner-free predictors are impossible, or that recurrence is necessary for the count function.

D.4 RETROSPECTIVE OBSERVABLE-INPUT OVERLAP AUDIT

The original separation check uses the full 16-token queue. Reconstructing the deterministic training generator confirms zero full-sequence overlap, but the network only reads eight tokens. The 6,000 training queues contain 3,631 unique eight-token inputs; the 24 evaluated arms contain 23 unique eight-token inputs. Fourteen test-arm inputs occur in training and ten do not. Every test target-count profile occurs in training. Hence full-sequence novelty is not model-input novelty, and neither implies target-profile novelty.

We additionally evaluate the already saved weights on the archived test inputs, without training or regenerating native labels. Table 5 reports descriptive errors split by the observable-input overlap.

The unseen-input subset is small and all its target profiles are familiar; these errors are not evidence of generalization to an unseen mechanism or target class. This overlap and subgroup analysis was added retrospectively.

Table 5: **Saved recurrent-model results and retrospective input-overlap audit.** Recovery is the worst positive-pair recovery for each seed. MAE is in 100-share inventory units. The final two columns average over 14 seen-input and 10 unseen-input arms respectively, using the saved weights; they are descriptive subgroup results.

Seed	Minimum recovery (%)	Median test arm MAE	Mean MAE seen inputs	Mean MAE unseen inputs
701	99.8048	0.00992	0.00976	0.00975
702	99.8837	0.00495	0.00506	0.00666
703	99.9049	0.00733	0.00662	0.00821
704	99.9473	0.00601	0.00546	0.00768
705	99.9513	0.00621	0.00632	0.00633

D.5 Evaluation-unit correction and saved-weight verification

The original exact-count control compared packet counts such as -2 directly with native inventories such as -200 shares, causing that control to fail. Multiplying the count vector by the fixed 100-share packet size resolves the mismatch on all 24 arms. This correction changes neither the saved weights nor the native labels, test ordering, thresholds, or learned metrics. The raw result and corrected numerical interpretation are both retained.

For the retrospective check, all five model-file hashes match the archived training receipt. Safe state-dictionary loading and inference under PyTorch 2.10.0 reproduce the archived per-pair learned excess losses to a maximum absolute difference below 10^{-8}. The receipt reports training under PyTorch 2.11.0. These are numerical inference checks, not a rerun of the original training chronology or proof of broad generalization.

E LEARNED WORLD-MODEL TRANSITION-INTERFACE ASSAY

E.1 ENVIRONMENT AND RECURRENT MODEL

PayloadLatch is a controlled dynamical system with hidden payload mass $m \in \{0.7, 2.2\}$. A short calibration interaction depends on m; the system is then dock-reset so the visible observation $q = [\text{position}, \text{velocity}, \text{latched}, \text{stage}]$ is matched while the recurrent model state is retained. Before deployment the payload is decoupled. A common zero-force latch action then makes payload mass control-relevant for the fixed continuation-action bank. Thus matched light/heavy histories can share the declared current interface at deployment while requiring different downstream controls.

We use a compact Gaussian recurrent state-space model with deterministic state dimension 128, stochastic state dimension 16, and hidden width 128. Training uses 1,600 mixed-payload trajectories, batch size 32, Adam with learning rate 3×10^{-4}, a reconstruction objective, KL weight 0.05 with one free nat, and a prior-prediction weight 0.5. Payload labels never enter the model input or loss. The model checkpoint is held fixed throughout the evaluations reported below; its hash is recorded in the supplementary manifest.

E.2 PRELIMINARY DIAGNOSIS

A preliminary analysis used 64 matched pairs held out from model training. The recurrent state retained a strong linear payload trace ($R^2 = 0.99645$), but the model's native imagined dynamics selected materially different light/heavy actions in none of the pairs and incurred mean regret 1.5708 times the strongest q-only shared-action baseline. A second model trained with an additional 20-step prior-rollout objective likewise produced zero action separation and a regret ratio of 1.6878.

A retrospective analysis of these preliminary results suggested a specific interface failure: the recurrent carrier contained the relevant information, but the action-conditioned transition readout did not use it effectively. The subsequent evaluation therefore tested only a pre-specified context-gated transition readout on a disjoint 128-pair set. The 64 preliminary pairs were excluded from readout selection, threshold setting, and final evaluation.

E.3 TRANSITION READOUT AND HELD-OUT EVALUATION

The readout holds the deployment carrier h_D fixed during a continuation rollout and predicts $\Delta q_t = q_{t+1} - q_t$ from the 390-dimensional feature map

$$[q_t, a_t, h_D, h_D a_t^{\text{force}}, h_D q_t^{\text{vel}}].$$

Features are standardized with a 10^{-6} standard-deviation floor, then fit once by multi-output ridge regression with $\alpha = 1$. The intercept is unpenalized. Fitting uses exactly the original 1,600 training episodes; there is no payload label, feature search, hyperparameter sweep, or fit on evaluation pairs. An additive comparison uses $[q_t, a_t, h_D]$ with the same fitting rule.

Evaluation uses 128 new matched-pair seeds, 920000–920127. For every pair, the readout evaluates the unchanged continuation bank, executes the selected action in the exact dynamics, and scores regret against the state-specific oracle within that bank. The strongest q-only baseline chooses one shared action inside each matched current-interface class. Table 6 reports the criteria fixed before this evaluation and the resulting values.

History erasure. The erasure control resets the recurrent carrier at the dock-reset boundary while replaying the same visible post-reset observation/action sequence. The transition readout is not refit. Mean regret rises from 0.000782 to 0.30419, action separation falls from 1.000 to 0, and the erasure removes more than the full q-relative improvement because the erased controller performs worse than the strongest q-only shared action.

Carrier transplant. For every matched pair whose readout-selected actions differ by at least 0.2, both directed transplants replace only the recipient h_D with the donor h_D while holding recipient q, readout weights, and action bank fixed. All 256 directed cases follow the donor-native decision; all 256 increase recipient real execution cost relative to its own readout-selected decision. This

Table 6: **Pre-specified evaluation criteria for the transition readout.** Criteria were set before the 128-pair evaluation.

Criterion	Required value	Observed value
Action separation across matched pairs	$\geq .75$	1.000
Mean regret / strongest q-only regret	$\leq .60$	0.00437
Erased-history action separation	$\leq .25$	0.000
Fraction of q-relative improvement destroyed by erasure	$\geq .50$	1.703
Eligible directed carrier transplants	≥ 1	256
Transplant decision follows donor	$\geq .95$	1.000
Transplant increases recipient real cost	$\geq .50$	1.000
One-step transition R^2	$\geq .90$	0.9975
Predicted/true response-contrast norm	$[.75, 1.25]$	0.9959

intervention localizes the useful information to the carrier-to-transition pathway; it does not imply that h_D is a minimal or uniquely interpretable state.

Reproducibility and scope. The readout specification and model checkpoint were recorded before evaluation on the 128-pair set. Two independent executions reproduce the reported metrics. The result supports only the controlled claim that a retained historical distinction can be present in a recurrent carrier yet absent from the native action-conditioned transition interface, and that the specified context-gated readout restores prospective control on this substrate. It does not establish generality to native robotics benchmarks or uniqueness/minimality of the feature map. Exact source and checkpoint hashes, including historical program identifiers used in the original execution, are preserved in the machine-readable supplement.

F Native Cartpole finite-interface stress test

F.1 Models, root distribution, and finite interface

The environment is the unmodified `dm_control` `cartpole/balance` task. We evaluate three Gaussian RSSMs with deterministic state dimension 128 and stochastic state dimension 16; their weights are held fixed throughout this analysis. Before the deployment evaluation, a separate screening set was used to choose one root distribution: 32 random-action control steps with amplitude 1.0 and action repeat four. The selected distribution was then checked on 128 disjoint roots, which were excluded from interface fitting and final evaluation.

For each model seed, 384 new calibration roots and 512 new audit roots are generated, with the same physical roots paired across model seeds. A 32-centroid codebook is fit only on the 144-dimensional recurrent inference state at calibration roots. One deployment action per code is then selected by averaging the model's 40-step imagined reward costs over calibration states in that code. The action bank is $\{-1, -.5, 0, .5, 1\}$; no native future-task label enters the code-to-action table.

At test time, deployment reads only the 32-code interface and executes its associated action for 40 native control steps. The future task is unchanged across branches: F is the five-component vector of mean native $(1 - \text{reward})$ over 32 steps for the same five constant actions. It is evaluated before deployment, after deployment, and after an equal-duration zero-action time control from saved simulator states.

F.2 Information diagnostic

For each retained code, define empirical information risk as the best shared action's average future loss minus the average state-specific oracle loss. The shared action is reoptimized within the evaluated fiber, so this is an information diagnostic rather than held-out policy performance. Before evaluating the audit roots, four decision criteria were fixed: pre-deployment risk at most .01, post-deployment risk at least .03, an increase of at least .02, and a post-minus-time-control difference of at least .01.

Table 7: **Cartpole finite-interface information diagnostic.** All three models show the same directional increase, but none reaches both pre-specified effect-size criteria.

Seed	Pre	Post	Time control	Post−Pre
92601	.005775	.016329	.003036	.010554
92602	.004905	.019192	.003032	.014287
92603	.004839	.019211	.003474	.014371

The post-minus-time-control differences are .013293, .016161, and .015737. Thus all three models show greater information risk after deployment than under the time control, but the absolute post-deployment risks and pre/post increases fall below the two larger effect-size criteria. We therefore treat this result as suggestive boundary evidence rather than confirmation. Bootstrap intervals and the full pre-specified decision rule are provided in the supplementary audit package; no threshold is changed after observing the results.

Approximate response-splitting pairs are constructed only among roots sharing both pre- and post-deployment codes. Using a pre-response maximum component gap of .01 and a post-response gap of .05, the three seeds yield 52, 53, and 59 response-splitting pairs. These pairs share roots and are not independent samples. A retrospective audit finds that only 11, 4, and 1 of those pairs respectively also change the state-specific optimal future action, so response separation should not be interpreted as the same number of action-conflict witnesses.

F.3 Held-out repair and baselines

Repair policies are fit on calibration labels and then held fixed for evaluation on the 512 audit roots. The declared augmentation adds a four-bin pole angular-velocity channel, with bin boundaries fixed from calibration data. A within-code shuffled-channel control, a 15-neighbor full-recurrent-state baseline, and a 15-neighbor raw-observation baseline are evaluated under the same future loss.

Table 8: **Held-out Cartpole repair.** Regret is evaluated on disjoint audit roots after fitting on calibration roots only.

Seed	Code only	$q + \omega$	Shuffled	Full state	Raw obs.
92601	.018096	.012125	.017862	.003843	.002536
92602	.021639	.009532	.022345	.003231	.002951
92603	.020886	.006531	.021922	.003993	.003198

The angular-velocity augmentation reduces code-only regret by 33.0%, 55.9%, and 68.7%. Shuffled channels do not provide comparable improvement. Raw-observation performance is slightly better than the full recurrent-state baseline, so this experiment does not establish a need for recurrence. It instead shows that a finite learned output interface can omit decision-relevant information that remains available upstream. The added channel was specified before the audit but is not claimed to be minimal.

Scope and provenance. The analysis uses native simulator losses and fixed checkpoints but does not constitute an exact continuous-state deployment-closure theorem. All three model seeds share the same 512 physical roots. The earlier root-distribution screen is separate from this audit, and its evaluation roots are excluded from interface fitting and final evaluation. Source and return manifests, saved-model hashes, calibration/audit split, independent numerical recomputation, and the original unmet effect-size criterion are archived in the supplementary package.

G DATA PROVENANCE AND REPRODUCIBILITY

The supplementary manifests identify archived inputs, model weights, source files, and their hashes.

Numerical sources. The randomized information-set analysis uses the complete archived native-label cohort, saved public observations, and the associated queue bank; the source configuration pins the ABIDES implementation used for those outcomes. The learned-predictor audit uses the archived native test labels, fixed test cohort, training configuration, original sampler, saved model state dictionaries, and training receipt. The unit-alignment correction changes neither the saved weights nor the native labels or learned metrics.

Prospective and retrospective status. The randomized queue generator, sample size, observation catalog, execution schedule, designated-pair collision endpoint, and pair-conditioned loss endpoint were fixed before cohort generation and native labeling. The no-pair-identity risks, all-fiber collision counts, exhaustive observation-subset check, exact generator enumeration, observable-input overlap audit, and seen/unseen-input error split are retrospective analyses of the same archived records. They are identified as such in the main text and are not relabeled as prospective endpoints.

Executable scope. The supplied script `analysis/reanalyze.py` reproduces the information-set analysis with exact rational accumulation of the finite risks, checks the native-label/count relation, reconstructs the training input set, and optionally evaluates the saved weights with `--check-models`. `results/reanalysis.json` and `results/risk_frontier.csv` separate cohorts and information restrictions; figure generation reads those outputs. A manifest records included input-file hashes. The package reproduces the reported reanalysis and learned inference from archived outputs; it does not rerun the original ABIDES simulations.

Learned world-model study. The supplementary world-model records distinguish the preliminary 64-pair analysis from the later 128-pair evaluation. They include the recurrent checkpoint, training recipe, transition-readout source, pair-level results, and intervention outputs. The feature map and fitting rule were chosen after the preliminary analysis but before the 128-pair set was evaluated. Replaying the analysis reconstructs the training recipe, fits the two linear readouts on the original training episodes, and executes the specified controls without updating the recurrent model. Historical program identifiers are retained only inside machine-readable archives so that the hashes remain verifiable.