When Learned Options Become Infrastructure: Promotion-Induced Maintenance in Control Lifecycles

Anonymous authors
Paper under double-blind review

Abstract

When a learned option becomes an action for another policy, a change to its behavior can also invalidate that policy. Which callers should be suspended, and when is collecting more evidence cheaper than revalidating them? We study these questions as **promotion-induced maintenance**. Our rule uses the option-to-policy dependency graph twice: to block potentially stale calls and to choose which option to probe next. Unresolved dependencies remain blocked. On identical action–consequence data from `96` fresh CARL PointMass worlds, removing the graph eliminates the allocation response, supplying the wrong graph reverses it, and restoring the graph restores the original decisions. Correct topology lowers a source-calibrated maintenance-cost objective by `0.9%` on average. One option-reliability failure keeps the preregistered composite endpoint unconfirmed. On one released SIL-C checkpoint, dependency-local revalidation preserves guarded execution of two policies outside the affected closure during validation and saves `4.17%` of executed validation-plus-service actions, with identical paired trajectories for all 24 policies. These results show how downstream dependencies can guide evidence collection as well as maintenance. They do not establish benefits across natural task streams.

1 Introduction

Learning an option and relying on it are different commitments. Once a later learner can call an option as an action, its policies depend on that option's behavior. A change in the environment can then invalidate both the option and its callers. Relearning the option alone does not decide which callers to suspend or which unrelated policies can keep running. This issue can arise whenever a learned capability forms part of another learner's action interface. We study it through options and call it **promotion-induced maintenance**. A service here is a downstream policy that can be invoked through the runtime.

Dependencies also affect what the learner should investigate. Two options can have equally uncertain behavior but very different costs of checking their callers. An extra probe may avoid an expensive round of downstream validation for one option and save little for the other. Our rule uses the same dependency graph to decide both which calls to block and which evidence to collect. It stops probing when the expected validation work avoided no longer pays for the additional actions; unresolved dependencies remain blocked.

Final task return need not reveal these costs. In our MountainCarLoCA experiment, four maintenance choices reach the same final reward, yet retain different numbers of usable options and differ by nearly seven orders of magnitude in planning work. A reset control scores lower, so the reward can distinguish some methods while missing this difference. We therefore measure service availability and maintenance work alongside task performance.

CHiRP (Nayyar & Srivastava, 2025) already combines option invention with planning and continual transfer. Partial Models (Alver et al., 2025) and LoFo (Rahimi-Kalahroudi et al., 2023) localize adaptation after a change, while SIL-C (Lee et al., 2025) maintains compatibility as skills evolve. We build on these problems. Our focus is the link between the dependencies created by reuse, the cost of validating those dependents, and the choice of further evidence.

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.

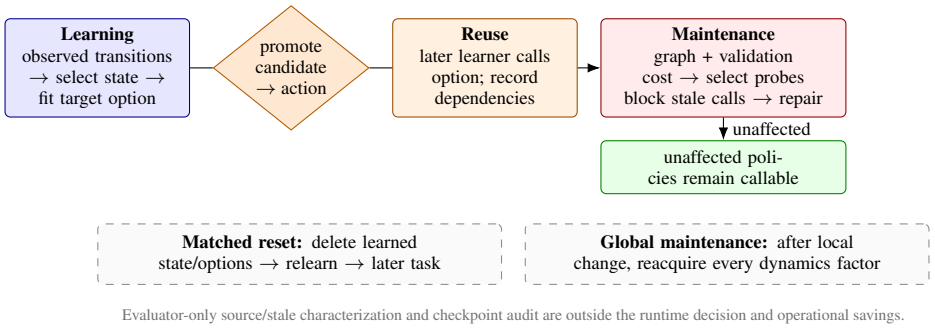


Figure 1: After promotion, other policies can call the learned option. Their dependencies determine both which calls to block after a change and which probes may reduce validation work. Dashed boxes show the reset and global-maintenance controls.

We implement the rule in a runtime that learns predictive models and feedback options, makes qualified options available to a later learner, and records dependencies from execution. After detecting a change, it suspends potentially affected options and policies before their next action, repairs the affected models, and revalidates the suspended capabilities. Unrelated policies remain available. The evaluated learning and execution paths make no LLM calls.

Figure 1 shows the lifecycle and its reset and global-maintenance controls. Our contributions are:

1. A topology-conditioned acquisition rule, tested by removing, replacing, and restoring the graph under identical data. Our central mechanism experiment tests whether downstream dependencies change which evidence is worth collecting under a fixed maintenance objective.
2. A maintenance problem formulation and executable lifecycle, with explicit tests of pre-use blocking, service availability, repair, and action cost on held-out Reacher and Acrobot trials. A separate released-SIL-C experiment tests dependency-local revalidation with a live call guard.
3. Three tested non-equivalences that motivate the measurements: a planned option may never be called, final return can hide service interruption, and small one-step prediction error can coexist with an invalid downstream service.

The contribution is this use of dependencies across acquisition and maintenance, not a new option-learning algorithm or a new sequential-testing primitive. Concrete alternatives with the same information and graph match our implementation, and preregistered tests found no non-additive promotion-by-repair interaction. These controls limit claims of superiority without removing the dependency effect tested here.

2 Promotion-induced maintenance

2.1 Setup and dependency tracking

The learner acts through a fixed set of primitive actions A_0 and receives masked observations O. Each transition produces a receipt $R_t = (o_t, a_t, o_{t+1})$: the observations before and after an executed action. From these receipts it selects a predictive state Z and fits options Ω. A later learner can then use qualified options as actions. The dependency graph G records which options and downstream policies may be affected by a local change Δ.

Before promotion, each option must pass a held-out reliability test for its declared starting region, feedback policy, termination rule, and intended effect. Callers then select it by identifier, but every primitive action still counts toward cost. Option trajectories link options to dynamics factors; executed compositions link policies to options. On confirmed change, the runtime suspends the affected closure until repair and revalidation. The closure marks possible dependence, not certain failure.

2.2 TOPOLOGY-CONDITIONED EVIDENCE ACQUISITION

Let S record which root dynamics factors each option may depend on, and let G record which downstream services call each option. The validation cost W(S;G) sums the option and service work that would follow one change at each root. A support test classifies each option–root edge as present, absent, or unresolved within a fixed number of observations. Unresolved edges stay in S.

For a set C of options, the model predicts support S'_C at the end of the sampling budget. The scheduler values further probes by $V(C) = W(S;G) - \mathrm{E}[W(S'_C;G)] - Q(C)$, where Q(C) bounds their remaining primitive-action cost. Thus a probe is useful when its expected reduction in downstream validation work exceeds its cost.

We report calibrated decision cost as selected probe actions plus W(S_final;G). The latter sums fixed source-calibrated option/service costs over each root's closure; the public repeat does not execute that revalidation separately for every arm or estimate natural change frequency. For example, if a service depends on two options with unresolved support, clearing either edge alone leaves the service quarantined and avoids no service revalidation. Clearing both can remove that shared cost, making joint evidence worthwhile when either probe alone is uneconomical.

If every V(C) is nonpositive, the scheduler stops probing and retains all unresolved edges. This can leave extra services quarantined, but stopping cannot create a false-safe release relative to the support tests. Otherwise it probes the least-observed option in the highest-value set and updates from the new receipt. We call this rule fail-closed. A topology-hidden control values only option support; a sequential-completion control samples every stream to termination. The tests and posterior calculation are standard. A generic downstream-loss implementation given the same graph exactly matches our rule.

2.3 WHAT COUNTS AS A COMPLETED LIFECYCLE?

A trial passes only if it learns reliable options, uses them to acquire a later capability with fewer actions than a matched reset, blocks affected calls before execution, keeps unrelated capabilities available, repairs affected policies, and preserves behavior after checkpoint restoration. It must also respect the declared observation and action restrictions. A higher return alone cannot pass this test. Because the tasks and changes are designed experiments, a pass establishes capability in those conditions, not frequency or benefit in a natural task population.

2.4 COST ACCOUNTING AND EXPERIMENTAL STATUS

We count primitive actions inside every option. Formation, later acquisition, maintenance, and validation costs are reported separately, as are parameter counts and checkpoint size. Evaluator-only work is disclosed and never subtracted from a method's operational cost.

We use reserved for trials whose rules and unused seeds were fixed before their outcomes, and opened for analyses of already observed data. Table 2 separates reserved lifecycle confirmation (Panel A), **opened implementation comparison** (Panel B), reserved negative certification (Panel C), opened topology replay (Panel D), and released-method confirmation (Panel E). CARL, official-CHiRP-host, MountainCarLoCA, and scaling experiments answer different questions; we do not pool their counts.

3 SYSTEM

3.1 OBSERVATIONS AND ACTIONS

The runtime uses a narrow Gymnasium API. Each step records masked observation, learner-visible action, next observation, and one action of cost. Reward, evaluator success, hidden goals, and native action semantics are excluded; learned objects, dependencies, availability, checkpoints, and costs are logged. Formation and execution have no frontier-LLM call site.

On Reacher-v5, the learner sees angle sine/cosine pairs and an opaque rotated action; velocity, target, fingertip displacement, and reward are hidden. Formation uses 48 by 12 random-control

steps plus 20 held-out episodes. Evaluation and local probes use disclosed evaluator-controlled state placement.

On Acrobot-v1, it sees four angle sine/cosine values and three opaque, secretly permuted tokens; velocities, reward, termination, and mapping are hidden. It has no state placement, cloning, unwrapped-state read, or counterfactual rollout. A disclosed partition at `-0.16` and `0.16` radians defines candidate factors; visits and invocations determine dependency.

On official CARL PointMass, the public transfer sees two positions and a seed-specific opaque rotated action. Both velocities, hidden context and `context_id`, reward, and termination are stripped. Source effects define coordinate-scoped contracts; unannounced `wind_x`, `wind_y`, or `timestep` changes are inferred only from ordinary primitive prediction residuals before later-task replay.

3.2 Learning predictive state and feedback options

Acrobot uses 72 training and 24 validation episodes to choose between current-only and one-step-history ridge models of wrapped angle displacement (`10^-6` regularization). Four feedback candidates aim at offsets `+0.34`, `-0.34`, `+0.17`, and `-0.17`; promotion requires settling within `0.20` and a held-out success rate of `0.80`. Reacher uses the same selection procedure with six candidates. We design the candidate features and targets; the learner selects its predictive state and fits the controllers. The appendix gives the features, horizons, and thresholds.

3.3 Reusing options and recording dependencies

Each admitted controller becomes an action identifier for the later learner. Acrobot searches all 16 two-option programs on a fixed 4/8/16 schedule and tests the winner on 20 episodes at a `0.80` success threshold. Reacher learns two-option programs for six tasks at `0.85`. We link an option to regions visited in at least half its qualification episodes, and a policy to the options it calls. Checkpoints store these links and their transitive closure.

3.4 Detecting changes and repairing dependents

Acrobot receives an unannounced positive-region remap `0,1,2 -> 2,2,0`. From 480 training and 160 held-out local rows, replacement requires `50%` RMSE reduction while a clean-dynamics refit must not trigger; no downstream rollout or stale label informs the decision. Dependents leave the active set before another world action, then undergo local repair, while outside-support tasks are paired over 30 seeds. Checkpoints exclude raw transitions. Reacher uses 80 disclosed state-placed probes and `80%` error reduction.

4 Experimental design

4.1 Preregistered trials and development history

Reacher v3 and final Acrobot each reserve 12 previously unused trials, evaluated once. Acrobot balances two trials per hidden action permutation. Its primary criterion requires `10/12` passes, coverage of every permutation, no crashes or false change detections, decisions before stale use, and at most two complete shortcut matches. The stronger criterion requires `12/12`. We freeze the thresholds, 27 core gates, accounting, and 56-file source closure before evaluation. Acrobot passes the primary criterion at `11/12` but misses the stronger one because of post-restore policy reliability. Earlier Acrobot `7/12`, Reacher `8/12`, and CARL Acrobot/Pendulum `0/1` results remain separate development results; Pendulum includes one false-safe decision.

4.2 Controls and model-family intervention

Controls include current-only prediction, charged fixed-token search, matched reset, a same-rights two-observation cache, and global reacquisition. The cache tests representation form, not independent learning.

The Acrobot model intervention replaces all dynamics models with random-Fourier-feature kernel ridge regression: 96 features, validation-selected length scale, ridge `10^-4`. It receives the same masked receipts and shares lifecycle code but no fitted focal coefficient or checkpoint.

A standalone 48-dimensional reservoir consumes full observation–action history under matched Gymnasium, mask/action, protocol, budget, and accounting rights. Independent code implements option qualification, staged 4/8/16 later search, support, quarantine, local refit, restoration, and checkpointing. It imports no focal lifecycle code; refits retain its basis and replace only readouts. This is a conventional comparator, not a published-method reproduction.

We also run released CHiRP code on two FourRooms tasks, changing only stream length and output location. Trial 0 tests an action swap and door relocation; the next run supplies an independently trained library for the fixed relocation. Our maintenance layer reads saved options and transitions. It identifies potentially affected policies and repairs tabular options using state–action–next-state data and option termination contracts, without maps or downstream outcomes. We test maintenance on CHiRP-produced options, not a modification of its learning algorithm or a ranking against CHiRP.

A separately coded partial model also matches all eight shared-support worlds. Both preregistered `2 x 2` promotion-by-repair interactions are zero; these tests do not support representation exclusivity or non-additive synergy.

4.3 Outcome measures and certification tests

Trials must pass all criteria in Section 2.3. We report paired actions, reliability, NRMSE, parameters, checkpoint size, and transitions; counts such as `12/12` describe the tested cohort rather than a natural-population rate. A separate certification experiment fixes option-local hooking, task-scoped projection, and an independently coded full-history projection before evaluating `24` reserved worlds. It requires exact service closure and no false-safe decisions, with equal information and no option or service rollout before the decision. The conclusions concern these point-model certifiers, not all uncertainty-set, conformal, reachable-set, or behavioral certificates.

4.4 Testing the role of topology

The initial experiment fixes `256` iid support schedules and two role-swapped service graphs before observing outcomes. Each statistical unit is one schedule, averaged across the two graphs. We compare our rule with sequential completion, optimistic coalition, single-option marginal allocation, and topology-hidden support valuation. All receive the same receipts, probe budgets, support tests, and costs, and all retain unresolved dependencies. A preregistered repeat then uses `96` new public CARL PointMass worlds and support schedules, each once. Its endpoints are paired decision cost, the difference in observations between high- and low-cost roles, fail-closed support, identical paired receipts, option reliability, and no access to hidden context, reward, or termination.

After this cohort finishes, an opened causal replay changes only the graph supplied to the scheduler: correct topology, graph lesion, paired wrong topology, and restored correct topology. Receipts, option/service costs, thresholds, and action rights are fixed, and costs are evaluated on the true graph. No new environment actions are executed. This tests the cause of the allocation response; it does not change the outcome of the preregistered composite test.

5 Results and boundaries

5.1 RQ1: Can the runtime learn, reuse, and maintain its options?

The runtime completes `12/12` reserved Reacher and `11/12` final exact-source Acrobot trials. Every Acrobot action-permutation stratum passes at least once. Across all worlds, decisions precede downstream actions in the changed environment; no run crashes, falsely detects a change in the unchanged environment, or retries. Current-only and fixed-token shortcuts complete `0/12`; the same-rights history cache matches our runtime's recorded outputs in `12/12`.

The failed Acrobot world passes 26 of 27 core gates, but task 0 succeeds in only 23/30 rollouts after restoration. A post-hoc diagnostic reproduces the failure without finding lost checkpoint state: the same policy is insufficiently reliable. The reported result remains 11/12.

On CARL PointMass, history is selected under every hidden action map; learned probes and closures track wind_x, wind_y, and timestep. After disclosed 9/9 development, an exact-source cohort passes 18/18 reserved cells with zero false-safe decisions and full repair. This is public-environment capability, not published-method or population evidence; Reacher alone uses disclosed state placement.

5.2 RQ2: Does service topology causally change evidence acquisition?

This section tests the central mechanism: whether changing only downstream dependencies changes evidence allocation and its cost under the same maintenance objective. We distinguish the preregistered public repeat from the subsequent opened graph interventions.

In 256 prespecified iid support schedules, topology-conditioned acquisition lowers mean decision cost by 269.4 actions relative to the best tested topology-blind control (95% paired-bootstrap interval [−375.7, −168.2]). It gives the higher-loss option 2.61 more observations ([1.89, 3.43]) and meets the noninferiority criterion against every fixed control, including sequential completion, optimistic coalition, and single-option marginal allocation. The generic implementation with the same graph matches these decisions, so the benefit is not specific to our representation.

The public repeat completes all 96 CARL PointMass worlds. Mean calibrated decision cost is 146.2 actions lower than the topology-hidden control (paired difference −146.2; 95% interval [−271.2, −39.5]), with 2.93 more observations toward the higher-loss role ([1.39, 4.72]). The small average saving of 0.9% reflects where topology changes the decision: 78 worlds tie, and the other 18 save 779.5 cost units on average (4.8%). Of these, nine reduce the validation closure and save 1,830 (11.2%); the other nine spend 271 extra probes (1.7%) without reducing it. This outcome-defined split explains the result but cannot select useful probes in advance. Paired receipts match and all allocation decisions remain fail-closed. The **composite confirmation nevertheless fails** because only 95/96 worlds pass option reliability. The failed world is not replaced or retried.

In the opened causal replay, graph lesion reduces the high-minus-low allocation difference to zero. The paired wrong topology reverses it to −2.93 ([−4.71, −1.42]) and costs 200.6 more than correct topology when both are evaluated under the true graph ([89.2, 334.4]). Correct topology is never worse in the nine worlds with a cost difference; the other 87 tie. Restoring the graph reproduces every allocation byte-for-byte. All four conditions retain unresolved dependencies. The graph therefore causes the observed allocation and cost differences in this replay; this is not another independent confirmation.

5.3 RQ3: Does dependency-local maintenance preserve live services?

Released SIL-C provides a skill update without an author-injected fault. Of its 24 policies, 22/24 invoke the updated decoder, and intervention on that decoder changes service behavior in 20/22 pairs. The preregistered directional-stability gate fails: the experiment does not establish stable benefit or harm directions.

For a separate maintenance confirmation, we fix this dependency split and use a new environment/model seed. Global revalidation checks all policies; dependency-local revalidation checks dependents and places a live pre-use guard after decoder selection but before primitive action. Both arms produce exact paired trajectories and matching post-validation decisions for all 24 policies. Here callable means the prescribed execution window completes without guard abort, not passing a task-success threshold. Local maintenance executes requests for two policies outside the affected closure during validation. Each completes 0/20 full four-subtask tasks; the preserved behavior is partial-task execution, not successful completion of the full task. Local maintenance uses 257,212 rather than 268,412 total validation-plus-service actions, saving 11,200 (4.17%). Every changed-decoder request is checked before dispatch. This is a single-checkpoint, single-adjacency confirmation of behavior preservation and scheduling. Validation and service reuse the

same seed as detailed in the appendix; it is not a second independent reliability test or a repair of the failed directional-stability gate.

5.4 RQ4: Does persistence change complete acquisition and repair cost?

Final Acrobot retained acquisition used 92,090 actions versus reset's 147,258, saving 55,168; all 12/12 paired differences were positive (median 4,595, range 4,582–4,615). Selective repair used 30,138 versus global's 81,275, saving 51,137; all differences were positive (median 4,263.5, range 4,185–4,362). Reacher's selective/global repair ratio was below one in 12/12 (median 0.336, range 0.318–0.715). These are descriptive frozen-cohort distributions, with primitive option actions included.

Formation used 55,168 actions; checkpoint audit used 18,199; evaluator-only source and stale characterization used 17,236 and 37,896. None is hidden in savings.

A constructed grid spans N=4–16 options and sparse-to-dense closure. All 180/180 cells complete maintenance; the 28 sparse cells have median selective/global cost ratio 0.286, while fully affected cells equal 1.0. A same-rights full-observation implementation matches throughout, making this an accounting boundary rather than a representation or natural-scaling claim.

5.5 RQ5: What carries over to other implementations and CHiRP?

On twelve **opened** paired worlds, our runtime, the shared-scaffold RFF model, and the standalone reservoir complete 12/12, 10/12, and 11/12 trials. Both alternatives recover the same dependency closure in 12/12. These comparisons test particular implementations rather than rank model families in general.

Released CHiRP supplies promoted options and downstream consumers. Door-relocation confirmation repairs dependents, preserves unaffected pairs, and replicates on an independently trained checkpoint with 19/19 gates. After a sham audit excludes two direct-object estimates, one corrected source replay reaches 0.95 versus 0.99 and saves 20.0%; full lineage is in the appendix.

Across four checkpoints, outcome-blind plans reuse an option in 8/72 task positions, but fixed closure [6,7] only 1/72. A single-use nominal-only rule selects option 5 and task 7, yet realized traces call it zero times; valid repair has no continuation effect and adds 66,476 actions. Opportunity therefore need not become use, and the source-replay benefit remains conditional.

Table 1: Public-host evidence separates three quantities that endpoint return conflates. Counts within a checkpoint are correlated.

Quantity	Observation	Supported scope
Nominal opportunity	8/72 task plans; 5/20 options; fixed [6,7] query 1/72	Four-checkpoint descriptive census
Realized use	Nominal-only trial-2 selection; task 7 calls option 5 zero times and portable patch has zero effect	Single-use four-arm confirmation
Conditional benefit	0.95/0.99 success and 20.0% fewer actions	One audited source replay

5.6 RQ6: Which proxies fail at the maintenance boundary?

In a post-promotion MountainCarLoCA transfer, four retained methods tie at endpoint reward 2.0 across 12/12 starts, yet keep 2/2, 2/2, 1/2, and 0/2 valid options available; global maintenance uses 16,419,591 backups while typed selective uses two. Reset obtains reward 1.0, so the score is not globally saturated. The change is reward-only and formation is fixed, limiting this to an endpoint-versus-continuity witness.

In a frozen two-carrier assay, task-scoped and independently coded full-history point-prediction certifiers agree in 24/24 reserved worlds. Both are exact in MountainCar but share four CartPole false-safe decisions: stale-service success is 0.000–0.075 despite one-step NRMSE

Table 2: Claim-aligned lifecycle evidence. A: reserved focal confirmation; Acrobot passes its frozen primary endpoint, not the stronger 12/12 endpoint. B: separately opened, seed-paired implementation comparison; ESN is standalone. C: reserved negative endpoint. D: opened causal replay on the preserved public cohort; the source composite confirmation remains failed. E: single-use confirmation on released SIL-C. Evidence scopes are not pooled. O/L/R/C denotes option, later-policy, repair, and checkpoint reliability.

A. Reserved capability confirmation

System	Carrier	Complete	Pre-use	Cache	Current/fixed	Cost result
Focal	Reacher	12/12	–	12/12	– / –	max repair ratio 0.715
Focal	Acrobot	11/12	12/12	12/12	0/12 / 0/12	retain +55,168; repair +51,137

B. Opened seed-paired Acrobot implementation comparison

Model	Complete	O/L/R/C	NRMSE ↓	Params	Ckpt. B ↓	Save: retain/repair ↑
Focal compact	12/12	12/12 / 12/12 / 12/12 / 12/12	0.1676	56	2,917	55,180 / 51,127
Independent RFF	10/12	12/12 / 10/12 / 11/12 / 10/12	0.1689	212	11,045	55,176 / 46,728
Standalone ESN	11/12	12/12 / 12/12 / 12/12 / 12/12	0.0141	404	133,500	79,695 / 53,068

C. Reserved one-step service-certifier boundary

System	Exact / false-safe worlds	Focal/full-history match	Result
Task-scoped one-step certifiers	20/24 / 4	24/24	FAIL

D. Opened topology mechanism replay

Topology	Role effect	Calibrated cost	Boundary
C / lesion / wrong / restore	+2.93 / 0.00 / -2.93 / +2.93	-146.2 (C-L); -200.6 (C-W)	safe; source FAIL

E. Released-method maintenance confirmation

Arms	Exact decisions / trajectories	Complete actions	Available policies
Global / local + guard	24/24 / 24/24	268,412 / 257,212 (−4.17%)	0 / 2

E: Available means guarded execution during validation, not full-task success (both policies: 0/20).

`4.23e-4-4.89e-4`. Opened analysis removes them with an appropriate transition model or service-aligned risk threshold (appendix). This localizes missing model/risk information rather than proving an impossibility or option-specific representation advantage.

6 Related work

Options formalize closed-loop temporal actions, and lifelong option discovery studies later-task value (Sutton et al., 1999; Brunskill & Li, 2014). Predictive state representations support control from action-conditioned predictions (Littman & Sutton, 2001; Hefny et al., 2018); DADS and SkiMo learn reusable skills and skill-space dynamics (Sharma et al., 2020; Shi et al., 2023). CHiRP builds on CAT (Dadvar et al., 2023) to refine abstractions, invent options, and plan across tasks. These methods supply the learning ingredients; our contribution concerns the dependencies and maintenance decisions that follow reuse.

LoCA (Wan et al., 2022) evaluates adaptation to local change. Partial Models, LoFo, and factored nonstationarity (Feng et al., 2022) localize adaptation; SIL-C addresses compatibility between evolving skills and existing policies. These approaches could be combined to build a maintenance system. We instead ask how downstream validation costs should influence evidence collection before a maintenance decision, and measure availability and executed cost as well as return. Our MountainCarLoCA comparison illustrates what return alone can miss, and recent continual-RL evaluation (Ahn et al., 2025) likewise shows hidden failure modes.

Policy certificates and conformal risk control (Dann et al., 2019; Angelopoulos et al., 2024) provide complementary guarantees; Ramalingam et al. (2024) and Asiaee et al. (2026) address composition or risk under selection and intervention. Our one-step certifier failures concern the tested point models, not these methods. Similarly, the cache control that matches our evidence reuse rules out a claim of new statistical machinery.

Locatelli et al. (2016) classify means against a boundary, and Chen et al. (2014) allocates samples over structured classes. Our support tests use established statistical tools. What the maintenance graph adds is a downstream cost for unresolved support: the same uncertain edge can imply different validation work depending on its callers. The role-swap, lesion, wrong-graph, and restore comparisons test whether that cost changes sampling under identical receipts.

7 LIMITATIONS

We design the tasks, observation masks, changes, interfaces, thresholds, and schedules. CARL supplies public dynamics, but the option/task and topology protocols are ours. Topology changes allocation in `18/96` worlds, with a mean calibrated saving of `0.9%`. These experiments do not establish natural maintenance prevalence or production economics. The causal audit reuses observed data, and the original composite test remains failed because one world misses option reliability.

Exact dependency closure is limited to factor- or coordinate-scoped contracts; the coupled Pendulum experiment produces a false-safe decision. The RFF comparison shares lifecycle code, and we implement the independent reservoir ourselves. MountainCarLoCA starts with existing options. CHiRP supplies learned artifacts, but our code performs maintenance; its measured benefit is a corrected source-task replay, while the census uses plans clustered within checkpoints. SIL-C maintenance is confirmed on only one checkpoint and one update. We have not confirmed benefits on natural future-task streams or established a cross-method law.

Possible dependence does not guarantee actual harm. The CartPole failures also show that accurate one-step fits alone are insufficient for the tested service decisions; the subsequent structured-model and risk-threshold repairs are exploratory. Historical Acrobot `12/12` is compatibility evidence, not a replacement for the exact-source `11/12`. Our savings are local paired comparisons, with formation, evaluator work, storage, and compute reported separately. They do not estimate lifetime economic benefit. Matched alternatives support architecture-neutral behavior rather than a uniquely necessary representation.

8 CONCLUSION

Once other policies rely on a learned option, maintaining it involves both its behavior and its callers. Our rule uses these dependencies to block potentially stale calls and to choose further probes. Removing, replacing, and restoring the graph changes allocation under otherwise identical data; the SIL-C experiment separately preserves guarded execution of two policies outside the affected closure during validation, with identical paired behavior and `4.17%` fewer executed validation-plus-service actions. These are results for the tested graphs, costs, and interfaces, not universal necessity. The remaining question is how often such decisions pay off in naturally occurring task and change streams.

AI USE STATEMENT

Generative AI tools were used to propose and refine research hypotheses, design and critique experiments, implement and debug software, interpret results, search for and summarize literature, generate and edit figures, and draft and revise manuscript text. The authors manually checked cited claims against primary sources, reviewed generated code, exercised it with automated tests, and traced reported numerical claims to frozen result artifacts. No generative model was used as a runtime component in the evaluated formation or execution path, and no model-generated output served as a ground-truth label. The authors take responsibility for the final content, including text, claims, code, and artifacts produced with AI assistance.

REPRODUCIBILITY STATEMENT

The main text specifies learner-visible inputs, action rights, qualification thresholds, frozen evidence scopes, and native action accounting. The supplementary code includes the complete experiment implementations, frozen artifacts, claim-evidence ledger, figure and table generators, and focused tests. The appendix documents public-transfer, scaling, freeze-lineage, and non-confirmatory diagnostic details. Reserved confirmation, opened discovery, and the reserved negative endpoint are labeled separately throughout the paper.

REFERENCES

Hongjoon Ahn, Jinu Hyeon, Youngmin Oh, Bosun Hwang, and Taesup Moon. Prevalence of negative transfer in continual reinforcement learning: Analyses and a simple baseline. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/ba9e3d60610f3525717665966d86e0cd-Abstract-Conference.html.

Safa Alver, Ali Rahimi-Kalahroudi, and Doina Precup. Partial models for building adaptive model-based reinforcement learning agents. In *Proceedings of the 3rd Conference on Lifelong Learning Agents*, volume 274 of *Proceedings of Machine Learning Research*, pp. 959–977. PMLR, 2025. URL https://proceedings.mlr.press/v274/alver25a.html.

Anastasios N. Angelopoulos, Stephen Bates, Adam Fisch, Lihua Lei, and Tal Schuster. Conformal risk control. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/f3549ef9b5ff520a7e41ff3cc306ab2b-Abstract-Conference.html.

Amir Asiaee, Kaveh Aryan, and James P. Long. Partial causal structure learning for valid selective conformal inference under interventions. In *Proceedings of the 42nd Conference on Uncertainty in Artificial Intelligence*, volume 337 of *Proceedings of Machine Learning Research*, pp. 161–184. PMLR, 2026. URL https://proceedings.mlr.press/v337/asiaee26b.html.

Emma Brunskill and Lihong Li. PAC-inspired option discovery in lifelong reinforcement learning. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pp. 316–324. PMLR, 2014. URL https://proceedings.mlr.press/v32/brunskill14.html.

Shouyuan Chen, Tian Lin, Irwin King, Michael R. Lyu, and Wei Chen. Combinatorial pure exploration of multi-armed bandits. In *Advances in Neural Information Processing Systems 27*, 2014. URL https://proceedings.neurips.cc/paper_files/paper/2014/hash/d3ea0f3316d2da934d79b8b344eafee4-Abstract.html.

Mehdi Dadvar, Rashmeet Kaur Nayyar, and Siddharth Srivastava. Conditional abstraction trees for sample-efficient reinforcement learning. In *Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence*, volume 216 of *Proceedings of Machine Learning Research*, pp. 485–495. PMLR, 2023. URL https://proceedings.mlr.press/v216/dadvar23a.html.

Christoph Dann, Lihong Li, Wei Wei, and Emma Brunskill. Policy certificates: Towards accountable reinforcement learning. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 1507–1516. PMLR, 2019. URL https://proceedings.mlr.press/v97/dann19a.html.

Fan Feng, Biwei Huang, Kun Zhang, and Sara Magliacane. Factored adaptation for non-stationary reinforcement learning. In *Advances in Neural Information Processing Systems 35*, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/cf4356f994917177213c55ff438ddf71-Abstract-Conference.html.

Ahmed Hefny, Zita Marinho, Wen Sun, Siddhartha Srinivasa, and Geoffrey Gordon. Recurrent predictive state policy networks. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1949–1958. PMLR, 2018. URL https://proceedings.mlr.press/v80/hefny18a.html.

Daehee Lee, Dongsu Lee, TaeYoon Kwack, Wonje Choi, and Honguk Woo. Policy compatible skill incremental learning via lazy learning interface. In *Advances in Neural Information Processing Systems 38*, 2025. doi: 10.52202/085713-4296. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/bb97e9a7c811904c9b01f51fde66edcf-Abstract-Conference.html.

Michael L. Littman and Richard S. Sutton. Predictive representations of state. In *Advances in Neural Information Processing Systems 14*. MIT Press, 2001. URL https://papers.nips.cc/paper_files/paper/2001/hash/1e4d36177d71bbb3558e43af9577d70e-Abstract.html.

Andrea Locatelli, Maurilio Gutzeit, and Alexandra Carpentier. An optimal algorithm for the thresholding bandit problem. In *Proceedings of the 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 1690–1698. PMLR, 2016. URL https://proceedings.mlr.press/v48/locatelli16.html.

Rashmeet Kaur Nayyar and Siddharth Srivastava. Autonomous option invention for continual hierarchical reinforcement learning and planning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 19642–19650, 2025. doi: 10.1609/aaai.v39i18.34163. URL https://ojs.aaai.org/index.php/AAAI/article/view/34163.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems 20*, 2007. URL https://papers.nips.cc/paper/2007/hash/013a006f03dbc5392effeb8f18fda755-Abstract.html.

Ali Rahimi-Kalahroudi, Janarthanan Rajendran, Ida Momennejad, Harm van Seijen, and Sarath Chandar. Replay buffer with local forgetting for adapting to local environment changes in deep model-based reinforcement learning. In *Proceedings of the 2nd Conference on Lifelong Learning Agents*, volume 232 of *Proceedings of Machine Learning Research*, pp. 21–42. PMLR, 2023. URL https://proceedings.mlr.press/v232/rahimi-kalahroudi23a.html.

Ramya Ramalingam, Sangdon Park, and Osbert Bastani. Uncertainty quantification for neurosymbolic programs via compositional conformal prediction. *arXiv preprint arXiv:2405.15912*, 2024. URL https://arxiv.org/abs/2405.15912.

Archit Sharma, Shixiang Gu, Sergey Levine, Vikash Kumar, and Karol Hausman. Dynamics-aware unsupervised discovery of skills. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=HJgLZR4KvH.

Lucy Xiaoyang Shi, Joseph J. Lim, and Youngwoon Lee. Skill-based model-based reinforcement learning. In *Proceedings of the 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 2262–2272. PMLR, 2023. URL https://proceedings.mlr.press/v205/shi23a.html.

Richard S. Sutton, Doina Precup, and Satinder Singh. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112(1–2):181–211, 1999. doi: 10.1016/S0004-3702(99)00052-1. URL https://doi.org/10.1016/S0004-3702(99)00052-1.

Yi Wan, Ali Rahimi-Kalahroudi, Janarthanan Rajendran, Ida Momennejad, Sarath Chandar, and Harm H. Van Seijen. Towards evaluating adaptivity of model-based reinforcement learning methods. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 22536–22561. PMLR, 2022. URL https://proceedings.mlr.press/v162/wan22d.html.

A CARRIER INTERFACES AND RECEIPT SCHEMA

Every low-level receipt contains the masked observation before an action, the learner-visible action and its parameters, the masked observation afterward, and a unit environment-action charge. Reward, evaluator success, hidden goals, and native action semantics are excluded from formation and execution. The artifact records random choices, fitted models, promoted identifiers, learned dependencies, availability transitions, checkpoint contents, and environment-action counts.

Reacher-v5 exposes two joint angles as sine/cosine pairs and accepts a bounded opaque two-dimensional action, densely rotated before the simulator. Angular velocity, target coordinates, fingertip displacement, and reward are hidden. Formation uses 48 random-control episodes with 12 recorded steps and 20 held-out episodes. Broad initial-velocity evaluation and local causal probes use evaluator-controlled state placement; this matched diagnostic right is never attributed to the learner.

Acrobot-v1 exposes four angle sine/cosine values and three opaque tokens under a hidden token-to-native-action permutation. Both angular velocities, reward, termination, and the mapping are hidden. Standard reset and executed tokens are the only interventions: there is no state placement, simulator clone, unwrapped-state read, or counterfactual rollout. The authored second-joint partition has boundaries at `-0.16` and `0.16` radians. Empirical option visits and learned program invocation determine dependency; the partition itself is not claimed as discovered.

The public transfer uses the official CARL 1.1.1 `CARLDmcPointMassEnv` at commit `2adf54c...`. It exposes only the first two native observation coordinates (position) and accepts a two-dimensional learner action under a seed-specific hidden orthogonal mixing. Both native velocity coordinates, reward, termination, every context value, and raw `info` keys including `context_id` are stripped at the boundary. The learner has no state placement, clone, counterfactual rollout, or simulator parameter read.

B PREDICTIVE MODELS AND OPTION QUALIFICATION

Acrobot formation uses 72 training and 24 validation episodes, each with one random warm-up followed by 40 random actions. A row stores four current observations, previous wrapped two-angle displacement, token, next displacement, region, and provenance. Current-only ridge features are observation, token one-hot, their outer product, and intercept. The history design adds previous displacement and its token interactions. Both use ridge `10^-6`; held-out normalized RMSE selects the model with deterministic ties.

Acrobot option targets are relative second-joint offsets `+0.34`, `-0.34`, `+0.17`, and `-0.17`. At each step, the controller evaluates all tokens and minimizes predicted squared target error plus `0.08` predicted-motion energy. Success requires three consecutive steps within `0.20` radians inside a 45-step horizon. Qualification uses 20 held-out episodes and threshold `0.80`. Reacher uses six relative-angle options, current/history selection, a 60-step horizon, `0.12` tolerance, and six consecutive settled steps; its controller inverts the fitted action-effect map with fixed feedback gains.

C PROMOTION, HIGHER ACQUISITION, AND DEPENDENCY

Promotion serializes a qualified controller under an opaque identifier that becomes a later learner action. Primitive execution remains inside the runtime and every primitive step is charged. The later task supplies an opaque task ID, two decision stages, and scalar waypoint error after each selected option; waypoint coordinates and expected option identities are hidden.

Acrobot searches all 16 two-option programs. Four common-random repeats score every program, the four lowest-loss programs advance to eight, and the best two advance to sixteen. Summed exposed waypoint error, deterministic tie-breaking, and a paired runner-up margin select the candidate. Final admission uses 20 new episodes at success `0.80`. Reacher learns two-option programs for six opaque tasks and requires `0.85` reliability.

An option depends on each authored region visited in at least half of its qualification episodes; a learned policy depends on the options in its selected program. The checkpoint stores the transitive region-to-option-to-policy closure. Expected programs remain hidden until evaluation.

D CHANGE, QUARANTINE, REPAIR, AND RESTORATION

The Acrobot positive-region remap changes native consequences `0,1,2` to `2,2,0`, so every old consequence changes while both nonzero torque directions remain available. The runtime obtains 480 training and 160 held-out positive-region rows through standard resets and ordinary actions. With the previously selected history choice fixed, it confirms a change only if replacement held-out RMSE improves by at least `50%` over the stale local model; the identical clean refit must not trigger. The same rows perform diagnosis and repair, with no downstream rollout or sealed stale-success label available.

On confirmation, the learned support closure is removed from the active task map, and invoking a quarantined task raises before an environment action. Only after this mutation may the evaluator characterize stale behavior. Outside-support tasks use the same 30 reset seeds as their clean reference and must preserve both their reliability threshold and full paired success vector. The repaired local model is installed, affected policies are re-evaluated, and the active set is restored. Global reacquisition refits global and local models under changed dynamics and validates the complete policy set.

The checkpoint stores fitted coefficients, option/policy structure, learned support, and active/quarantined state, but zero raw transitions and zero LLM calls. Restoration is tested prospectively on the full active policy set. Reacher uses a sequential residual only as a warning and starts maintenance after 80 controlled local probes, split 60/20, demonstrate at least `80%` held-out error reduction; those probes use the disclosed state-placement right.

E FROZEN COHORTS, CONTROLS, AND ACCOUNTING

Reacher v3 has 12 unused single-attempt trials. The final Acrobot exact-source cohort has 12 exact-unseen trials, two in each hidden token-permutation stratum. Its primary endpoint requires `10/12`, at least one pass per stratum, zero exceptions and clean false confirmations, strict pre-use provenance in all worlds, and at most two complete shortcut matches; a strong endpoint requires `12/12`. The candidate, thresholds, 27 core gates, complete 56-file local import closure, and accounting were frozen before opening. The primary endpoint passes `11/12`; the strong endpoint fails because one restored checkpoint policy set has minimum success `23/30`, below `0.80`. Earlier frozen `8/12` Reacher and `7/12` Acrobot failures remain development lineage, not extra confirmation.

The current-only control receives the same formation rows without previous displacement. Fixed-token search tests every constant token per option and pays all search/evaluation actions. Matched reset deletes retained predictive/option state, reacquires it through world actions, and must recover the same higher program. A same-rights two-observation cache delegates prediction to the fitted map without an explicit promoted state object. Selective maintenance is paired with global reacquisition.

The independent Acrobot intervention replaces every focal fitted dynamics model, including local repair and global reacquisition, with random-Fourier-feature kernel ridge regression (Rahimi & Recht, 2007). Inputs are current masked observation, optional previous displacement, and token one-hot. A fixed seed produces 96 cosine features; held-out NRMSE selects length scale from `{0.75,1.5,3.0}` with ridge `10^-4`. It shares option qualification, higher search, dependency bookkeeping, quarantine, and evaluation, but no focal coefficient or checkpoint.

The standalone reservoir baseline imports no focal Acrobot lifecycle module. It independently wraps Acrobot-v1 with the same first-four-coordinate observation mask and hidden token permutation, and independently implements collection, fixed-reservoir state updates, ridge readout fitting, one-step feedback control, staged `4/8/16` two-option search, visit-based support, active-task quarantine, local readout refit, restoration, and JSON checkpointing. The 48-dimensional fixed reservoir consumes current masked observation, previous wrapped displacement, and previous token; candidate-token readout features include reservoir-by-token interactions. Global and positive-region readouts use ridge `10^-4`. Refits reuse the frozen reservoir basis and replace only readouts, preventing



Figure 2: Operational costs over the 12-trial exact-source Acrobot cohort (primary endpoint 11/12). Retention saves 55,168 actions and selective repair saves 51,137. Source formation (55,168), checkpoint audit (18,199), and evaluator-only source/stale characterization (17,236/37,896) are disclosed outside the paired bars.

random-feature resampling from masquerading as change evidence. The checkpoint retains zero raw transitions but serializes the fixed basis in this implementation. Shared items are limited to the public environment, mask/action rights, authored option/task/fault protocol, budgets, and accounting definitions. This is an independent implementation control, not an official reproduction of CHiRP, Partial Models, LoFo, or another published system.

Operational totals include primitive execution inside options. Retained/reset counts later acquisition and whatever reset must reacquire; selective/global counts change confirmation/refit and policy validation. Source formation, source-policy characterization, checkpoint audit, and post-decision stale-policy characterization remain separate. Evaluator-only actions are never subtracted from a baseline. The RFF checkpoint excludes its regenerable random basis and includes learned normalization and coefficients.

The machine-readable paired-table artifact retains every frozen per-world difference. Exact-source Acrobot retention saving has minimum/median/maximum `4,582/4,595/4,615` actions and selective-repair saving has `4,185/4,263.5/4,362`; both have the expected sign in `12/12` worlds. Reacher selective-to-global core-repair ratios have minimum/median/maximum `0.318/0.336/0.715` and are below one in `12/12` worlds. These summaries describe the deliberately stratified constructed cohorts. No iid population interval or natural fault-frequency claim is attached to them.

F CARL DYNAMICS TRANSFER AND CONSTRUCTED SCALING BOUNDARY

Two earlier opened CARL carriers remain in the development lineage. Acrobot and Pendulum each completed `0/1` full-gate worlds; Pendulum made one false-safe decision. PointMass was developed afterward, and none of these counts are pooled.

The exact selective-closure result is scoped to factor- or coordinate-specific executable contracts. PointMass makes that condition explicit by assigning options and residual tests to learned x/y response coordinates. Pendulum is a coupled counterexample for the current contract: its selective prediction leaves one invalid option active and an allegedly unaffected program drops below threshold. This establishes a false-safe boundary, not that dynamical coupling is its unique cause or that every coupled system must fail.

CARL PointMass formation collects 18 by 32 source training transitions and 8 by 32 held-out transitions. Candidate models use either action alone or previous two-position displacement plus action; both include an intercept and use ridge `10^-7`. Across three hidden action maps, current-only NRMSE is `1.020-1.059` and history NRMSE is `0.0160-0.0204`. Four constructed coordinate-displacement candidates (`+/-0.04` on x or y) become options only after three consecutive steps within `0.010` position and `0.00055` per-step displacement on their learned effect coordinate. Their controller solves the minimum-norm learner action for that coordinate from the fitted action-effect row. All four options and four hidden two-option programs qualify at `1.0` in every world.

An opened ordered preflight selects the first passing values after testing 5 `wind_x`, 5 `wind_y`, and 4 `timestep` severities: `wind_x=1.0`, `wind_y=1.0`, and `timestep=0.02` from source `0.004`. Later global families are not evaluated after timestep passes. Twelve by 28 primitive training and six by 28 held-out transitions supply detection and repair. Detector error improvement uses every receipt; the repair fit retains before/after positions within the promoted contract support `[-0.23,0.23]^2`. No context value, reward, termination, or changed-world downstream task action enters either decision.

The first three-world repeat passed `8/9`: under timestep in one world, the repair fit restored three options but the positive-x option stayed at `0/5`, propagating to two `0/6` programs. On that identical world and evaluator seeds, the original fit fails again, while either more receipts or the same-action contract-support fit restores `4/4`; v3 adopts the latter and reruns the same nine cells and gates. It passes `9/9`. Wind-x closes options/tasks `[0,1]`, wind-y closes `[2,3]`, and in every local cell at least one dependent option and both dependent programs are stale while both unrelated programs remain reliable. Timestep selects both coordinates and makes all four options/programs stale, so selective work correctly equals global work. Every repair returns all services to `1.0` with zero false-safe. Local validation saves `11,667` executed actions in total; the six local cells save strictly and three global cells save zero. These failure-driven opened results retain the `8/9` attempt and are not pooled with confirmation.

A single-use confirmation then freezes the same mechanism, severities, thresholds, gates, and contract-support fit, six repository-unseen world seeds, 52 local source/configuration files, and 86 CARL source files. The first invocation is terminated by the macOS GLFW sandbox before any world or receipt completes (`world_count=0`, `change_cell_count=0`); its immutable empty result is recorded as an infrastructure-invalid attempt. A disjoint recovery freeze changes only the execution right to one unsandboxed process and hashes that abort lineage before opening its seeds. It passes all `6/6` worlds and `18/18` cells: all family-specific closures are exact, all twelve local cells save strictly (`23,483` actions total), all six global cells equal global work, no cell is false-safe or executes a downstream task before its decision, and every repair restores all services. No retry or seed replacement occurs within either freeze. This is fresh confirmation of the tested public-dynamics capability, not an external-method ranking or population claim.

The constructed scaling artifact separately executes 180 lifecycle cells over option-library sizes `N=4,8,12,16`, `T=2N` tasks, four dense shared worlds, and sparse-to-dense changed-root/task closures. It recovers all `11,808` hidden edges, localizes every remap, preserves every unaffected task, and repairs every affected task. Across the `28` sparse cells, the selective/global median native-action ratio is `0.2857`; fully affected cells equal `1.0`, and a same-rights full-observation implementation matches all `180/180`. The exact additive regression follows fixed arity and fixed sampling budgets. We use it only to demonstrate an accounting/closure boundary in the designed family, not a natural scaling law or representation advantage.

G TOPOLOGY-CONDITIONED SUPPORT ACQUISITION AND CAUSAL AUDIT

The public repeat first executes complete option receipt streams, then each paired scheduler consumes its selected prefixes. Selected probe costs are the primitive actions recorded in those receipts. The maintenance term uses option/service costs calibrated on the source before the repeat, summed over the final closure for one declared event per root. It is a calibrated decision objective, not a fresh measurement of each arm's subsequent changed-world revalidation. The full receipt-bank collection cost is reported separately below; unused receipts are not charged as online probes.

The support learner observes a binary material-effect event for every option–root pair and applies bounded-look familywise tests at threshold 0.50, maximum 60 looks, and a four-look balanced floor. Present and absent edges terminate; an unresolved edge remains included at the horizon. The topology-conditioned rule receives the learned option-to-service graph and measured option/service validation costs. For every coalition of unresolved options, it computes current fail-closed validation work minus posterior-predictive work after bounded qualification minus the remaining empirical primitive-action bound. A positive highest-value coalition is sampled; an option in no positive coalition is retired from further probing but its unresolved edges remain included. Thus stopping may over-quarantine but cannot release an unresolved edge relative to the bounded tests. The Beta(1,1) posterior prediction and combinatorial search are implementation choices, not claimed statistical novelties.

The expected-risk discovery freezes 256 hash-declared iid Bernoulli support schedules before outcomes and averages two role-swapped service topologies within each schedule. Correct topology has mean decision cost 15,930.2, versus 16,199.6 for the strongest topology-hidden control; the paired difference is −269.4 with 95% bootstrap interval [−375.7,−168.2]. Relative to sequential completion it is −1,548.5 ([−1,903.0,−1,206.1]). The high-minus-low-role observation effect is 2.61 ([1.89,3.43]), with 57 changed schedules and no reverse. Optimistic coalition and single-option marginal controls are also worse in paired mean. A generic implementation that directly minimizes downstream loss on the same graph matches focal exactly.

The explicitly authorized public repeat freezes 96 disjoint CARL PointMass world, support-schedule, and execution seeds; the rule, primary topology-hidden control, thresholds, reliability floor, bootstrap seeds, and no-retry policy are fixed. It executes 34,560 option episodes comprising 1,889,293 support actions plus 79,872 formation actions. All 96 trials complete, all online boundaries are fail-closed, paired probabilistic options have identical receipts, and the learner receives zero context, reward, or termination values. The mean correct-minus-topology-hidden calibrated decision cost is −146.2 actions (95% paired-bootstrap interval [−271.2,−39.5]), and the role effect is 2.93 ([1.39,4.72]). Allocation changes in 18 worlds. However, all-option reliability is satisfied in only 95/96: slot 27 has one nominal option at 0/60 and two mixed-context options at 44/60, below 0.80. The prespecified full composite therefore remains failed even though cost, role, safety, rights, and paired-receipt endpoints pass.

An opened no-replay analysis partitions all 96 worlds by realized operation rather than selecting favorable outcomes. The accounting identity holds in every world: calibrated cost saving equals source-calibrated closure validation work displaced minus selected extra probe actions. The unconditional 146.2-action (0.9%) saving is therefore an exact mixture of the following regimes:

Table 3: Observed applicability regimes on the fixed 96-world public cohort. The partition is descriptive and exhaustive.

Regime	Worlds	Extra probes	Validation saved	Net saving
Topology inactive	78	0	0	0
Closure reduced	9	1,427	3,257	+1,830 (11.2%)
Probe only, no closure resolution	9	271	0	−271 (−1.7%)

Across all 18 topology-active worlds, mean saving is 779.5 actions (4.8%). Thus topology has economic value in the tested cohort exactly when the additional receipts shrink realized downstream validation by more than their probe cost. Because the partition uses realized closure outcomes, it is an explanatory applicability boundary, not a prospective classifier, natural prevalence estimate, or new confirmation endpoint.

The post-confirmation causal audit reconstructs every stream from the preserved receipt bank and reruns four scheduler inputs. Correct and lesion replay match the original focal and topology-hidden arms in every topology/world. Correct topology has role effect +2.93; lesion is exactly zero; supplying the paired wrong topology yields −2.93 ([−4.71,−1.42]) and costs 200.6 more than correct topology on the true graph ([89.2,334.4]). Correct restore is byte-exact. Every condition stays conservative. This localizes a behavioral causal role for downstream topology while remaining opened replay evidence rather than a second public confirmation.

Exact slot-27 replay and a model-by-start factorial localize the admission miss to the captured formation model, not world geometry, starts, environment order, or reuse. One admission-triggered extra ordinary receipt batch and refit restores both directional options on captured and eight fresh start batches, but an extension-only reset control also restores them. A disjoint 96-world discovery triggers the repair in 0/96, so this is a caught rare-tail engineering repair, not retained-memory exclusivity, public prevalence, or a paper headline.

H MountainCarLoCA post-promotion transfer

The public-protocol assay independently implements the MountainCarLoCA continuous transition equations and phase interface at the official deterministic setting: position [−1.2,0.5], velocity [−0.07,0.07], force 0.001, gravity 0.0025, terminal-2 radius 0.07, terminal rewards (4,2) in phase 1 and (1,2) in phases 2–3, local phase-2 starts in position [0.4,0.5] and velocity [0,0.07], and evaluation starts in position [−0.2,−0.1] and velocity [−0.01,0.01]. At position at least 0.4 with nonnegative velocity, the environment forces action 2, matching the public implementation.

This is a post-promotion maintenance assay rather than an official LoCA agent reproduction. A deterministic tabular planner with 171 x 141 = 24,111 states separately solves terminal-1 and terminal-2 feedback controllers, which are then executed in the continuous dynamics. Every controller achieves 24/24 qualification for every one of twelve seeded start cohorts. Phase 2 changes only the terminal-1 reward contract. Typed selective and a same-rights direct full-history implementation update that reward and the downstream terminal-choice program while leaving both transition controllers executable. A support-only closure additionally quarantines terminal 1 and performs eight permitted local revalidation rollouts (13–43 actions across cohorts). A blocking global baseline actually resolves both tabular policies, executing 16,419,591 state-action backups; the resulting policies exactly equal the retained policies. All four retained methods obtain terminal-2 reward 2.0, while their minimum valid-option availability is 2/2, 2/2, 1/2, and 0/2. Deleting phase-1 transition information leaves the phase-2 local-reset control unable to reacquire terminal 2 and yields reward 1.0.

The artifact preserves every rollout summary, method-native coordinate, and gate. Costs in actions, semantic contract writes, and model backups remain separate. The result supports endpoint-versus-availability ranking divergence on one published reward-change protocol. It does not support an official published-agent ranking, dynamics-change external validity, or population inference.

I Official CHiRP host and policy-support confirmation

For trial 0, we ran the released CHiRP repository at commit 0d95f202b1a9 with its official requirements, algorithm, and hyperparameters on a consecutive two-task prefix of native continuous FourRooms. The only execution changes were stream truncation and isolated output paths. Problem 0 consumed 270,550 native actions and promoted option IDs 4–6; problem 1 retained all three, added 7–8, consumed 63,311 training actions, and reached final evaluation success 1.0. The machine-readable artifact retains the full commit and SHA-256 values; the option- and abstract-transition-library hashes begin a6768eb808e8 and a1ca33dd84ff.

The frozen intervention swaps native action consequences 2 and 3 only in the authored upper-right region y >= 24.75, x >= 16.5. Both operational arms diagnose the swap from 20 ordinary probes per action at the same state and spend 40 actions. Quarantine precedes every downstream option action. Repair changes only q-table best-action entries at the affected abstract state; validation uses official CHiRP option execution with step_max=800.

The first confirmation used each option's single stored nominal trajectory as dependency evidence and selected only option 4. It fails unchanged: pre-change option 4 is 0.87, repaired success is 0.81, and nominally unaffected option 5 changes from 0.99 to 0.83. A same-seed diagnostic then finds exact baseline/repaired outcomes and step counts for option 4 in 100/100, while stale option 5 differs in 19/100 outcomes and 23/100 step counts. Thus q-table repair works, but one nominal trace under-covers the stochastic policy's possible support; unpaired thresholding also mixed mechanism error with sampling variation.

The revised closure uses the persisted abstract transition relation plus q-table best-action reachability. It takes zero additional environment actions, selects [4,5,6], and leaves [7,8] outside support. After an opened development pass, single-use confirmation freezes disjoint evaluation, operational, and probe seeds while retaining the same checkpoint and intervention. All eleven gates pass. Dependent repaired outcomes and step counts equal their paired baselines in 300/300; unaffected stale outcomes and step counts equal baseline in 200/200. Option 4 changes 0.93 -> 0.04 -> 0.93, option 5 changes 1.0 -> 0.89 -> 1.0, and conservatively included option 6 remains 1.0.

The separately executed selective arm spends 40 + 15,821 = 15,861 native actions; full-library maintenance spends 40 + 23,857 = 23,897. The actual saving is 8,036 actions (33.6%), and options 7–8 remain available only under selective maintenance. Shared validation records match exactly. The 149,054 paired evaluator actions that characterize baseline, stale, and repaired behavior are reported separately and never subtracted from either operational arm. This first revision is fresh in random seeds only, not in checkpoint or fault family.

The second intervention changes the family rather than resampling the swap. Development first rejects two door edits because they would place an existing initiation state inside a wall. It then reports all three boundary-valid candidates: narrowing the old opening changes duration but not success; moving it three cells gives stale success 0/40 for option 6 and 13/40 for option 7; moving it six cells gives 0/40 for both. We freeze the smallest relocation crossing the severity gate. Closing cells (15,6–8) and opening (15–17,9–11) changes free-space topology. On an unaffected interior state and an old-door boundary state, enumerating all 24 primitive-action permutations leaves maximum mean-displacement error 0.491 for the best identity match.

Each operational arm independently scans ten wall-adjacent states with 24 one-step downward probes. The receipts recover all three closed nominal-door cells and three opened nominal-wall cells. Only then do retained nominal option trajectories select [6,7]; [4,5,8] never execute before this quarantine decision. Repair is not an inverse edit: separate tabular Q-learners consume continuous state, submitted primitive action, next state, and the already-promoted option termination contract. They execute 1,200 episodes per dependent option, inspect neither map contents nor downstream task success, and fall back to the retained CHiRP controller outside learned concrete states.

Single-use confirmation freezes these rules, source hashes, 100-rollout validation, and disjoint probe, training, validation, and post-decision diagnostic seeds. All 13 gates pass. Both independent arms infer [6,7] and reproduce identical repair-table hashes. Their final 100 training episodes succeed at 1.0; validation is 100/100 for both repaired options in both arms. Fresh stale success is 0/100 for option 6 and 26/100 for option 7, while options [4,5,8] match baseline in both outcome and steps for 300/300 paired rollouts. Selective maintenance spends 240 + 115,332 + 5,991 = 121,563 actions; global requalification spends 240 + 115,332 + 22,764 = 138,336, an actual saving of 16,773 (12.1%) while three options remain available. The 196,981 diagnostic actions occur after both decisions and are excluded.

For the cross-checkpoint test, we generated exactly one additional official two-task checkpoint at trial 1, without searching trials. It reaches final evaluation success 1.0 after 51,668 task-1 training steps and contains option IDs 4–8; its library hash begins 8e4c694cb70d, distinct from trial 0. We reused the already selected three-cell relocation and derived dependency IDs from trial 1's own retained trajectories. After a passed discovery, a single-use confirmation freezes 1,242 top-level seeds disjoint from discovery, the complete local source closure, the checkpoint hash, 100-rollout evaluation, and 19 exact gate keys.

All 19 gates pass. Both independently executed arms infer [6,7]; stale success is 0/100 and 37/100, and both repaired options validate at 100/100 in each arm. Options [4,5,8] again match base/stale outcome and duration in 300/300. Selective maintenance spends 240 + 114,275 + 6,067 = 120,582 actions; global requalification spends 240 + 114,275 + 22,943 = 137,458, saving 16,876 (12.3%) while all three unaffected options remain available. The 210,930 diagnostic actions are excluded. Together, the topology result replicates at two independently trained checkpoints; it does not estimate checkpoint-population prevalence, establish an official CHiRP method, formal all-support safety, arbitrary topology generalization, or head-to-head superiority.

I.1 PUBLISHED-METHOD SOURCE-TASK COMPARISON AND FRESH-CHECKPOINT CORRECTION

To obtain an actual published-method coordinate rather than another authored global-maintenance arm, we precommitted to the first subsequent native CHiRP task whose pre-change retained-option plan intersects the frozen dependent set `[6,7]`. A read-only preflight inspected tasks 2–19 in order; none uses either option, an observed overlap of `0/18` for this checkpoint, closure, and change. We stopped before training and invoked the disclosed fallback: source task 0, the first original task whose plan uses a dependent option, replayed at continuation slot 2. Its plan uses option 6 but not option 7. This selection uses only pre-change plans and does not screen changed-world outcomes; all reported savings are therefore per triggered event, not expected benefit over the published stream.

The first trial-1 integration stopped before an endpoint because a locally defined wrapper recursed while unpickling. Its replacement completed an opened source replay, but directly returned persisted runtime option objects instead of invoking released CHiRP's fresh reconstruction at the task boundary. A later equal-work sham showed that this adapter alone changes continuation behavior; we therefore preserve but exclude the opened `33.3%` estimate. This is a comparison-semantic failure, not evidence against the maintenance capability.

We then froze a single-use replication before training trial 2. Exactly one new official two-task checkpoint was generated; its option-library hash begins `b3bfdc29cfe8`, differs from trials 0 and 1, and remains byte-exact across all arms. Nominal retained trajectories again derive dependent IDs `[6,7]` and unaffected IDs `[4,5,8]` without reading changed-world endpoints. The frozen task selector again finds no task 2–19 using the affected closure and replays source task 0. Geometry checks preserve every option boundary. The primary endpoint requires all arms to complete, success at least `0.90`, portable non-inferiority within five points, every dependent validation at least `0.95`, complete additive accounting, and at least `10%` action saving.

The original v3 endpoint passes its 18 frozen gates, but inherits the same direct-object mismatch, so its `45.3%` estimate is excluded. A post-confirmation correction keeps the checkpoint, task, topology, closure, maintenance seeds, and changed-only arm fixed. It first executes and charges identical maintenance while discarding its patches; under official fresh reconstruction, this sham behaviorally reproduces changed-only continuation exactly: `410,554` actions, `0.99` endpoint, final steps, option IDs and q-table state counts, and learning-event sequence. The corrected portable arm again spends `240 + 116,364 + 6,012 = 122,616` maintenance actions, then `205,741` continuation actions. It reaches `0.95`, learns `[4,5]` rather than official `[4,5,6]`, and totals `328,357`, saving `82,197` (`20.0%`) versus changed-only. The four-point endpoint gap is within the frozen five-point boundary. Against the equal-work sham total `533,170`, persisted patch content accounts for `204,813` fewer actions. This correction audits the already opened checkpoint; it is not a second fresh confirmation.

An opened nominal scan of trials 0–2 finds one fixed-closure candidate, trial-0 task 6. We run tasks 2–6 in released order without source replay. The predeclared primary gate fails because unchanged task 6 does not actually call option 7 after the intervening tasks. Under changed dynamics, changed-only and corrected portable do call option 7; portable reaches `0.98` versus `1.0`, using `6,651,949` versus `7,657,048` complete actions and saving `1,005,099` (`13.1%`). An official-fresh equal-work sham exactly reproduces changed-only behavior on the audited coordinates. We report this as an opened changed-world future-prefix result, not confirmed pre-change natural reuse or future-task transfer.

Two preserved infrastructure stops precede the scientific attempt. The first verifier compared Python tuples with their JSON list representation and stopped before an attempt marker. The second inherited an interpreter missing released dependency `gym==0.26.2` and stopped before creating any checkpoint file or arm. A v3 recovery freezes all released requirement versions and opens trial-2 scientific data once. Because the task-boundary correction follows that opening, we classify the final result as an audited source replay on a fresh-trained checkpoint rather than a pristine second confirmation; it does not establish future-task transfer, population prevalence, arbitrary-topology generalization, or universal superiority.

A separate future-task protocol then freezes two geometry-safe door relocations, nominal-only lexicographic selection over released tasks 2–19, no source-task fallback, and a single new checkpoint. Two zero-data infrastructure stops preserve the holdout; a third attempt completes only source-task

training before a headless plotting crash, so trial 3 is retired. With the plotting backend fixed before opening, trial 4 completes both official tasks, reaches task-1 evaluation `0.99`, retains options `4–8`, and produces library hash `7dd9c6c2a34e`. Both relocations induce dependent set `[6,7]` and pass geometry checks, but no future nominal plan intersects it. Thus all `36` raw topology–task pairs are ineligible, representing one unique closure query per task, and the protocol stops before changed-world evaluation or any arm. No source task, replacement fault, or replacement checkpoint is used. A prose-only artifact label says `trial-3`; all protocol, path, and hash fields identify trial 4, and no gate reads the label.

After this stop, an outcome-blind descriptive census enumerates every retained option, future task 2–19, and nominal trace cell for the four distinct trial-0, trial-1, trial-2, and trial-4 libraries. Three of four checkpoints have at least one nominal reuse; `8/72` checkpoint–task plans use a retained option, `5/20` checkpoint-specific retained options are ever reused, and the complete option–task matrix has `8/360` hits. The fixed `[6,7]` query has only `1/72`, while outcome-blind strict local-cell candidates exist in `3/4` checkpoints. Trial 4 itself contains a different nominal opportunity–option 4 in task 19–so its fixed-query non-hit is not structural impossibility. Checkpoints are the primary clusters; task and option pairs within them are correlated descriptive coverage, not independent prevalence trials.

A single-use follow-up freezes a nominal-only cross-checkpoint rule before any changed-world endpoint: maximize future-task coverage, then shortest prefix, smallest closure, largest trace count, lowest trial, and cell. It selects trial 2, cell `(25,10)`, option 5, and task 7. Four official-fresh task-2–7 arms share the byte-exact library and runtime. From 64 receipts, portable infers the barrier and closure `[5]`, quarantines before continuation, keeps `[4,6,7,8]` available, spends `62,636` training plus `3,776` validation actions, and validates `100/100`.

The positive endpoint fails: task 7 calls option 5 zero times in unchanged, changed-only, and portable execution, using learned option 18 instead. Changed-only and sham continuation match exactly. Portable and sham total `7,561,049` actions at `0.90` success; changed-only totals `7,494,573` at the same success. The valid but unused patch adds `66,476` actions with zero continuation effect. This confirms opportunity-versus-use, not future-task benefit; with trial-0 non-use, nominal plan overlap is not an invocation guarantee.

J CORRECTED SIL-C PUBLIC-METHOD AUDIT

We separately reproduced the released SIL-C Kitchen workflow at pinned commit `eaaccb57...` and tested whether a frozen downstream policy's official sequential-completion service changes when an appended skill decoder is available. The complete enumerated adjacency contains policies `1–24`; all 24 run online. Before online outcomes, released traces classify 22 nonzero-use primary pairs and two zero-use controls, so no policy is removed from the incidence denominator. Each pair uses old, focal, lesion, and independently loaded rescue arms for 20 episodes of at most 280 steps. This is an external-method phenomenon audit, not an arm in the focal lifecycle confirmation and not a ranking of SIL-C.

Two earlier sweeps remain excluded. The first paired wrapper overrode `reset`, while the released server calls `reset_model` directly. After that seam was corrected, a v3 source audit found that released LazySI falls back to process-global NumPy sampling when a sparse entry map is missing; sequential arms therefore consumed different random draws. The v3 artifact is permanently classified `PROTOCOL_INTEGRITY_FAILURE`. Before v4, we added arm-local NumPy streams, exact focal/rescue action and observation gates, official-checkpoint regression tests, and a 24-pair preflight. One launcher then stopped on a missing `python` executable before a server, confirmation seed, result, or scientific outcome existed; the disclosed resume changed no scientific input.

The single-use v4 run uses environment/model seed `20260903`, no retry or replacement, and the same frozen 22 primary references, two controls, service metrics, and effect thresholds. It completes with zero errors. All `24/24` rows use the corrected schema and arm-local streams; focal and rescue actions and observations match exactly. All `22/22` primary pairs invoke the appended decoder and pass seeded, low-level rescue, service rescue, and causal-trajectory gates; both controls make zero appended-decoder calls. Endpoint or temporal service differs in `20/22`. Mean directions are

17 benefit, three harm, and two neutral. Pairwise 5,000-replicate bootstrap labels are stable for six benefit, one harm, and two neutral pairs, with 13 unstable.

The precommitted protocol required at least 20 causal pairs, ten sensitive pairs, three raw directions per benefit/harm side, and two bootstrap-stable directions per side. Protocol integrity, coverage, sensitivity, and raw mixed effect pass; the stable harm count is `1 < 2`, so the final decision is `DESCRIPTIVE_MIXED_EFFECT_NOT_STABLY_CONFIRMED`. A deterministic post-hoc census over all 24 executed rows records updated-decoder invocation in `22/24` and causal service sensitivity in `20/24` overall (`20/22` given invocation). This supports one native published-adjacency occurrence witness, not a new confirmation or repair of the failed sign-stability gate. It does not establish frequency beyond this adjacency, maintenance-cost benefit, universal benefit, representation superiority, all skill-learning methods, or cross-method prevalence. Static follow-up audits also found no zero-training substitute: official HRM relearns dependent upper consumers after a lower reward-machine change, while SkiMo/SPiRL release no trained downstream consumer and their relevant checkpoint/interface route would require training plus an authored executor intervention.

J.1 Single-use dependency-local maintenance confirmation

Using the same released checkpoint and dependency split, a separate freeze fixes a new environment/model seed (`20260907`), all policies `1–24`, 20 episodes per policy, the validation and service windows, and two primary arms: global revalidation versus dependency-local revalidation with a live pre-use guard. No policy is retrained, replaced, or retried. The guard executes after the interface requests a decoder but before that decoder produces a primitive action or the environment receives it. Source, freeze, validation, service, and 29 checkpoint hashes are verified before analysis.

All nine frozen gates pass. The two arms make identical post-validation decisions and exact paired trajectories for all 24 policies, with no stale dispatch or guard failure. Global maintenance makes every service unavailable during validation and spends `134,206` validation actions. Dependency-local maintenance validates only the 22 dependent policies, spends `123,006`, and keeps policies 2 and 6 callable; requests for both policies execute during the validation window under `5,600` pre-use guard checks each. The common service window costs `134,206` actions per arm, so complete operational validation-plus-service work is `268,412` versus `257,212`, a saving of `11,200` (`4.17%`). A diagnostic replay of the two unaffected policies costs another `11,200` actions; it is reported separately and is not subtracted from either arm. This confirms the operational value of the frozen dependency split at one released checkpoint and adjacency. It does not establish natural prevalence, lifetime economics, universal necessity of explicit topology, representation exclusivity, or a repair of the failed directional-stability gate above.

Callable in the SIL-C overlay means that the prescribed 20-episode window completes without a guard abort; it is not an admission test against a task-success threshold. In particular, the unaffected policies 2 and 6 each complete zero of 20 full four-subtask tasks in validation, with mean rewards `1.35` and `1.55`; their partial-task behavior is preserved exactly. The server restarts the same environment/model seed between validation and service windows, and all cross-stage global trajectories match. The seed is fresh relative to discovery, but the service window is a paired replay rather than an independent held-out reliability test.

K Exact-source confirmation and historical compatibility replay

The supplement supplies the archived frozen Acrobot module and an isolated verifier for the original 56-file closure. The later shared-module extension is retained separately, and its replay is not labeled byte-exact frozen execution.

Every reserved result retains its freeze-manifest hash, frozen seed list, and pre-open source/protocol verification result. The final Acrobot run freezes the complete local import closure, not a hand-selected subset: 56 files hash to freeze `2f1597e7...`, verified immediately before opening and after completion. Twelve repository-unseen seeds cover each hidden action permutation twice and run once without retry or replacement. Its primary endpoint passes `11/12`; one world fails only prospective checkpoint reliability after otherwise completing formation, promotion, quarantine, repair, preservation, and both cost comparisons.

An opened post-hoc diagnostic preserves that failure. Restoring its JSON checkpoint and serializing it again gives exact object equality with zero raw transitions and LLM calls. On the same rollout seeds used immediately before serialization, the restored runtime reproduces every success bit and rates `26/30, 30/30, 30/30`; it also reproduces every frozen prospective bit, including task 0 at `23/30`. Across twelve new diagnostic 30-episode cohorts, task 0 pools to `313/360=0.8694` (95% Wilson interval `[0.8307,0.9004]`), with one cohort again at `23/30`; the other tasks pool above `0.986`. This localizes the miss to a prospective reliability tail of the same policy, not checkpoint state loss. The cohorts do not upgrade `11/12`, replace the failed seed, or change the frozen `0.80` threshold.

The earlier Acrobot reserved cohort achieved `12/12`, but three shared modules later evolved while the certifier was instrumented. Its original hashes remain recorded, and a separate non-confirmatory compatibility replay reruns all twelve worlds under the released implementation without reading frozen outcomes during execution. It exactly matches the old aggregate, decision, and every pre-existing per-world field after excluding timestamps. This establishes historical behavioral compatibility, not recovery of that unavailable source snapshot. The final exact-source `11/12` cohort therefore supplies the headline Acrobot count and costs; the two cohorts are not pooled.

L Long-horizon service-validity boundary and opened repair

The single-use CARL certificate confirmation freezes `12` CartPole worlds with `gravity: 9.8 -> 6.0` and `12` MountainCar worlds with `max_speed: 0.07 -> 0.010`. Its three primary arms are option-local validation and hooking, task-scoped projection, and an independently implemented full-history whole-task projection that does not call the focal projection function. All arms use ordinary primitive before–action–after rows and make zero changed-world option or service calls before deciding. Source code, CARL source, environments, thresholds, tasks, changes, and hash-derived reserved seeds are frozen; no seed is replaced or retried.

Task-scoped and independent full-history predictions agree in all `24/24` worlds. Both are exact against post-decision service evaluation in `20/24`: MountainCar passes `12/12`, while CartPole passes `8/12`. Each of the four failures is a task-1 false-safe. Predicted patrol rates are `0.833–1.000`, but actual success is `0.000–0.075`; changed-model one-step NRMSE is only `4.23e-4–4.89e-4`. Option-local validity also disagrees with service validity in both directions, losing valid coverage in four CartPole worlds and exposing a stale service in seven. These overlapping counts are descriptive and are not pooled as independent failures.

The reserved result remains failed. On six disjoint opened development seeds, we then compare equal-budget passive four-step, generic balanced, and contract-directed primitive probes. Each arm receives `2,600` training and `900` validation actions. Replacing absolute next-state fitting with a translation-invariant delta model yields zero false-safe decisions and full valid-service coverage in all `18/18` arm–world cells; all three probe policies match. A degree intervention shows degree 1 already gives exact service decisions, although degree 3 has lower multi-step primitive validation error. A diagnostic-only replay of the fixed degree-3 model on the twelve opened CartPole confirmation worlds corrects all four prior false-safe decisions. This is an opened structural-model repair and an active-probe null result, not a confirmation rescue or a new certifier claim.

M Versioned statistical evidence reuse

A separate constructed family tests fixed nonadaptive services of depths `1,2,4,8` over option libraries of size `8,12,16`, two worlds, and sparse-to-full changed closure. Each factor stores controller-counterfactual calibration scores from primitive receipts and binds them to controller and factor versions. Changed factors alone receive fresh detection, fitting, and calibration rows; no changed-world option or service is invoked before `keep / quarantine / abstain` decisions. The narrow finite-sample statement assumes iid calibration/deployment initial states, linear additive action-independent residuals, factor independence, invariant unchanged factors, and a fixed service graph.

The preserved v1 artifact uses a `0.20` service-risk budget while operational validity requires success at least `0.90`. It passes `16/18` cells and makes three focal/direct shared false-safe decisions with

true success `0.8792`, `0.8875`, and `0.8875`. Version v2 changes only the budget to `1 - 0.90 = 0.10`; worlds, seeds, changes, gains, data, models, tasks, evaluator, and costs stay fixed. It passes `18/18`, yields zero false-safe decisions, and includes `140 / 104 / 44` keep/quarantine/abstain decisions. Bypassing stale-version checks creates a false-safe in `17/17` applicable cells; wrong same-size dependency substitution does so in `11/11`; correctly rebound repair restores every service.

The cache-disabled direct end-to-end arm matches focal decisions, risk, and coverage but gives median focal/direct certificate-work ratios `0.754` overall and `0.732` for one-root changes. Version v3 adds the predeclared same-rights kill control: a generic cache keyed by learned projection, controller version, and factor version, using neither the option-certificate class nor focal compiler. It matches focal decisions, intervals, risk, coverage, false-safe behavior, and certificate work in all `18/18` cells. Hence the ratios measure versioned score reuse relative to recompute-all; they do not establish an option-boundary statistical or compute advantage. The v1–v3 lineage is opened architecture-neutral lifecycle evidence, not confirmation or a standalone certificate method.