How Much of a Policy Library Must Be Kept in Continual Offline RL?

Anonymous authors
Paper under double-blind review

Abstract

Continual offline reinforcement learning trains a specialist policy for each arriving task from static datasets, and a natural way to avoid forgetting is to keep every specialist in a growing library. Yet tasks share behavior, so how much of this library must be retained to keep serving every task seen so far? Task labels and parameter isolation cannot answer this question, since neither reveals whether an existing policy can actually solve a task. This paper answers this question with behavioral evidence. Paired-reset validation tests each candidate policy directly on each task identity and builds a directed compatibility graph. From this graph, we select a cover that decides both which policies to keep and which task each kept policy serves. The same behavioral evidence also bounds its own cost. Once every completion of the unobserved edges yields the same retention decision, further validation cannot change the outcome, and we can skip it. On 3D navigation benchmarks, our framework covers every task identity while retaining a minority of the specialists, improves service success over per-identity retention, and removes a substantial fraction of candidate validation episodes without altering any retention decision. It thus provides a simple, verifiable answer to what must be kept, and to when further validation is no longer necessary.

1 Introduction

A continual learner must preserve useful behavior as new tasks arrive. In offline Reinforcement Learning (RL), each task provides a static dataset from which the learner can acquire a specialist policy without further exploration (Levine et al., 2020; Khetarpal et al., 2022; Pan et al., 2026). Keeping every specialist avoids overwriting earlier policies, but also accumulates redundant capacity: recurring tasks may require familiar behavior, and even distinct tasks may be served by the same policy. This raises a basic question: how much of the policy library must be kept in order to continue serving every task it has seen?

Answering this question requires more than counting tasks or protecting parameters. An environment label identifies a task, but does not establish whether a policy learned elsewhere can solve it. Parameter isolation preserves earlier solutions without revealing whether they are still needed, and deleting a specialist because another policy appears similar can sacrifice task performance. A useful retention rule must therefore connect the decision to share a policy to evidence about its behavior. This paper answers the question with behavioral evidence.

Behavioral testing supplies this evidence, but comparing every candidate on every historical task can require substantial environment interaction, potentially offsetting the benefit of a smaller library. The central challenge is to retain enough policies to cover the tasks while collecting only the validation evidence needed to choose them. We study this under stable task identities and resettable historical environments, where paired-reset validation compares a candidate and a reference policy from shared initial conditions to inform retention and routing, not parameter updates.

We represent validated reuse as a directed graph whose edges indicate that a candidate donor policy can serve a target identity. A cover of this graph defines both the retained library and the route from each identity to a donor. At each arrival, selection is local to the previously retained policies and the new specialist. The key insight is that choosing this cover need not require revealing the entire graph: if all possible completions of the unknown edges lead to the same donor and route assignments under a fixed selector, further testing cannot change the decision.

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.

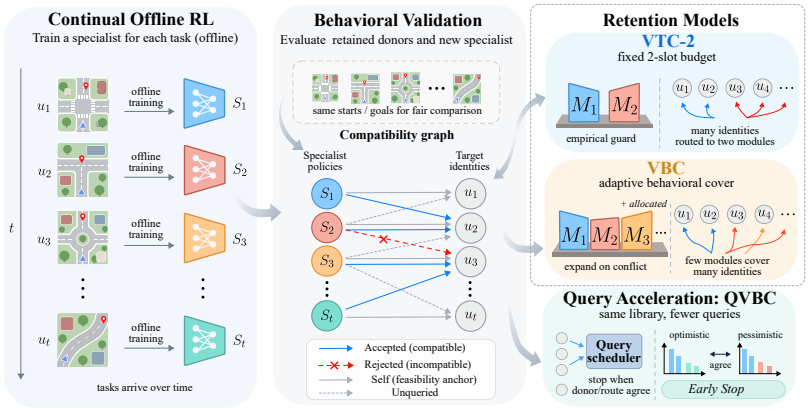


Figure 1: From offline-trained specialists to a compact retained library. Paired-reset validation supplies directed compatibility edges, with self edges anchoring feasibility. The retention rule chooses donors and routes under either a fixed budget or adaptive capacity. Query-efficient validation compares pessimistic and optimistic graph completions and stops when their complete retention decisions agree.

Figure 1 brings these ideas together in a family of verified-reuse algorithms. The approach includes fixed-budget methods like Verify-Then-Commit with two slots (VTC-2) and adaptive-capacity methods such as Verified Behavioral Cover (VBC), which minimizes library size while covering necessary tasks. To reduce environment interactions, Query-efficient VBC (QVBC) halts testing early once pessimistic and optimistic graph completions yield identical retention decisions.

Our experiments examine when this approach yields a useful compact library and what this compression costs. We study three navigation environments from two benchmarks: SimpleTown and AmazeVille, two 3D first-person environments from Continual NavBench (Kobanda et al., 2025a), and PointMaze, a MuJoCo environment from OGBench (Park et al., 2025) that serves as an independent backend. On SimpleTown, a small library maintains high success but requires adaptive capacity when behavioral conflicts arise, and recency-first querying, which gives priority to the most recently arrived identity, removes a substantial fraction of the validation episodes. PointMaze confirms that adaptive capacity preserves coverage with fewer policies at a modest quality cost. In AmazeVille, where each identity arrives only once, adaptive retention covers every identity with a minority of the specialists and improves service success, which scores each unserved identity as zero, over per-identity retention. Across all three backends, selective validation preserves every retention decision. Our contributions center on using behavioral evidence to decide both what to retain and how much to validate:

- We formulate compact policy retention through validated reuse, developing complementary fixed-budget and adaptive-cover implementations.
- We characterize when behavioral conflicts demand additional capacity, measuring the resulting trade-offs among library size, task coverage, and policy quality.
- We establish a stopping rule that preserves exact retention decisions, demonstrating significant reductions in validation queries.

2 Continual Retention with Behavioral Validation

Tasks, specialists, and routes. A stream introduces tasks at occurrences $t = 1, \ldots, T$. We refer to the stream state after the first t arrivals as prefix t. At each occurrence, a static dataset is associated with a stable environment identity u_t and used to train a specialist S_t without interactive exploration (Levine et al., 2020). Let $\mathcal{U}_t$ be the set of identities seen so far, with $U_t = |\mathcal{U}_t|$ and a known upper bound U. We study what happens after training: the learner retains a library $\mathcal{L}_t$ of K_t modules and assigns each seen identity to a donor through a route map $\rho_t : \mathcal{U}_t \to \mathcal{L}_t$. A full cover provides a route for every identity. A fixed-budget control may instead abstain on identities it cannot cover.

Algorithm 1 Verified Behavioral Cover with QVBC

Require: Arrival specialist S_t, retained library $\mathcal{L}_{t-1}$, validation mode
Build local pool $\mathcal{C}_t = \mathcal{L}_{t-1} \cup \{S_t\}$ and evaluate self edges
if validation mode is full **then**
    Reveal all unknown non-self validation edges against active identities
else
    Call QVBC query loop to reveal required non-self edges
end if
Apply exact selector to accepted graph to determine donors and routes
Commit selected donors and deterministic routes to $\mathcal{L}_t$
Delete unretained objects from the library
return Updated library $\mathcal{L}_t$

Evidence for reuse. Validation tests whether a candidate donor can serve a target identity by comparing it with an identity-specific reference policy. Both policies run from paired initial states specified by a reset manifest, a fixed list of start and goal configurations. The resulting compatibility edge is specific to the donor, target, reference, and validation rule. Compatibility is directed: a policy that serves another identity need not admit reuse in the reverse direction. A self edge connects a specialist to its own target identity, whereas a non-self edge records reuse on another identity. Validation informs retention and routing but never changes policy parameters. Independent fresh evaluation then measures the quality of the selected routes.

Retention objective. At each occurrence, selection is restricted to the local candidate pool $\mathcal{C}_t = \mathcal{L}_{t-1} \cup \{S_t\}$. The adaptive objective is to cover all seen identities with as few modules as possible from this pool, an online covering problem (Alon et al., 2009). For hierarchical specialists, a module is the high-level component, with a shared low-level policy accounted for separately. Persistent state includes policy components, reference behavior records, cached compatibility edges, routes, and normalization statistics. We report persistent storage, training work, and validation episodes separately so that a compact library is not mistaken for a reduction in every resource cost.

3 Verified Reuse and Decision-Preserving Validation

3.1 From behavioral edges to a retained library

A specialist can be discarded when the remaining policies still cover every observed identity. Behavioral edges make this criterion explicit. Self edges anchor feasibility by retaining the option to use a specialist on its own target, while accepted non-self edges create opportunities to share policies across identities. The retention rule turns these opportunities into a library and routes for each target. The two retention variants differ in capacity allocation. VTC-2 maintains two slots and uses an empirical guard to decide which policies to keep. VBC allows capacity to adapt, selecting a minimum-cardinality cover of the accepted graph within $\mathcal{C}_t$. Among covers of equal size, a scalar score from validated edge statistics and a fixed total-order tie-break determine the donors and routes. Both are specified in advance. Unretained objects are deleted and cannot re-enter later candidate pools, making selection dependent on the preceding retention path.

Algorithm 1 summarizes the update. We use an exact selector as the reference for query equivalence, while greedy and fixed-budget controls are evaluated separately in Appendix B.

3.2 What an accepted edge guarantees

Reuse should preserve performance relative to a reference while meeting an absolute competence threshold. The retention graph can enforce these requirements empirically or through a statistical certificate. To express both rules, let $Y_{\text{can}}, Y_{\text{ref}} \in \{0, 1\}$ denote success on a shared reset, and define $d = Y_{\text{can}} - Y_{\text{ref}}$. A paired failure occurs when the reference succeeds and the candidate fails, with probability $p_- = \Pr(d = -1)$. Conversely, $p_+ = \Pr(d = +1)$ measures paired improvement. The expected success gap is $\Delta = p_+ - p_-$, and the absolute candidate success rate is $q = \Pr(Y_{\text{can}} = 1)$.

An empirical rule applies fixed thresholds to estimates of these quantities (e.g., $\hat{q} \geq 0.80$, $\hat{p}_- \leq 0.20$, $\hat{\Delta} \geq -0.10$ on 100 paired resets). The certified rule requires a Clopper–Pearson upper confidence

bound on p_- to lie below a tolerance $\varepsilon_{\mathrm{saf}}$ and a lower confidence bound on q to exceed a competence floor c_{flo}, using n_2 paired resets. An initial stage of $n_1 = 50$ resets rejects a candidate as soon as its paired failures exceed the largest count that could still pass the safety certificate on the full n_2 resets, or as soon as its candidate failures exceed the largest count compatible with the competence floor. A separate margin filter rejects edges with $\hat{\Delta}$ worse than -10 percentage points. Both checks are reject-only and cannot accept an edge on their own. The margin filter in particular lies outside the simultaneous guarantee of Theorem 1 below, and Appendix A shows that the stronger 10-percentage-point certificate is not attained at the sample sizes used here.

The certificate relies on three assumptions, as formally stated in Appendix A: a finite frozen stream with training-validation separation, i.i.d. paired-reset design, and a Clopper-Pearson acceptance rule with pre-allocated family-wise error. The family size follows from a counting argument. At each occurrence, the local pool contains at most U retained policies together with the arriving specialist, so the new specialist can be tested against at most $U - 1$ earlier identities, while each retained policy can be tested against at most the one arriving identity. Each occurrence therefore introduces at most $2U - 1$ potential non-self edges, over which the family-wise budget is divided uniformly.

Theorem 1 (Finite-horizon non-self commit safety)**.** *Consider a known horizon T with at most U stable identities. Let $K_{\max} = U$, $E_{\max} = T(2U - 1)$, and $\alpha_{\mathrm{loo}} = \alpha/(2E_{\max})$. Under Assumptions 1 to 3 in Appendix A, with probability at least $1 - \alpha$ over validation manifests and rollout randomness, every accepted non-self edge (i, u) satisfies $\Delta(i, u) \geq -\varepsilon_{\mathrm{saf}}$ and candidate success rate $q(i, u) \geq c_{\mathrm{flo}}$.*

Appendix A provides the proof. The resulting certificate bounds relative success loss and ensures an absolute competence floor for every accepted edge, following the principle of safe policy improvement (Laroche et al., 2019; Thomas et al., 2019). All certified experiments in this paper use α=0.10, $\varepsilon_{\mathrm{saf}}$=0.20, c_{flo}=0.80, and n_2=100, as specified in Appendix C.

3.3 Query until the retention decision is determined

Full validation spends queries on edges whose outcomes cannot change the selected library. QVBC instead maintains a pessimistic graph G^- (self edges and known accepted edges) and an optimistic graph G^+ (adding every unknown non-self edge with an accepted label and an upper-bound score), as illustrated in Appendix Figure 6. When the exact selector yields different donors or routes on G^- and G^+, the algorithm queries one unrevealed non-self edge on the current optimistic routes (a witness), updates the graphs, and repeats until both graphs agree.

Proposition 2 (Decision Equivalence)**.** *Assume $G^- \subseteq G_{\mathrm{true}} \subseteq G^+$, the optimistic score of every unrevealed edge is an item-wise upper bound on its attainable score, the selector is exact, and a fixed total order and tie-break are independent of unknown outcomes. Under these premises, the agreeing pessimistic and optimistic decisions equal the full-revelation decisions.*

Appendix B provides the proof of the proposition. Induction extends equivalence to the complete donor and route path.

3.4 State and cost implications

The dynamic capacity K_t is bounded by $U_t \leq U$. For a hierarchical library, persistent state scales with active identities, not occurrences T. A deployment-only package can omit reference modules, and selective querying reduces non-self tests at each occurrence. Table 1 summarizes the three conclusions: the accepted-edge certificate, decision equivalence, and the persistent-state bound.

4 Experiments

Our evaluation follows three questions: whether compact libraries remain useful, when behavioral conflicts require more capacity, and how much validation can be avoided without changing retention decisions. We study three navigation environments from two benchmarks: SimpleTown and AmazeVille, two 3D first-person environments from Continual NavBench (Kobanda et al., 2025a), and PointMaze, a MuJoCo environment from OGBench (Park et al., 2025) that serves as an independent backend. Throughout, VBC selects covers with the exact selector and Greedy denotes the

Table 1: Guarantees for edge acceptance, query stopping, and retained state. Implementation evidence appears in Appendix Table 5.

Object	**Guarantee**	**Conditions**	**Outside the guarantee**
VBC accepted non-self edge	Theorem 1: family-wise safety and competence	Assumptions 1 to 3. Pre-allocated family error α	Self edges. 10 ppfilter. Uncertified paths
QVBC terminal donor/route decision	Theorem 2: same donors and routes as full verification	$G^- \subseteq G_{\text{true}} \subseteq G^+$. Exact selector. Fixed tie-break	Query optimality. Greedy path equivalence
VBC retained-state bound	$K \leq U$. With bounded U, persistent bytes have no T term	Self feasibility. Prefix-local selection. Eviction	Sublinear scaling in U. Training cost

Table 2: Comparison on the SimpleTown benchmark stream (maze topology 1). Success is the mean over 48 test conditions (16 per seed, 100 episodes each). State includes all persistent storage. Bold marks the best descriptive value and underline marks the second-best in each numeric column (higher success, lower state). PI-first and PI-best-validated coincide on displayed fields. Baseline definitions appear in Appendix C. Per-seed results appear in Appendix Table 10.

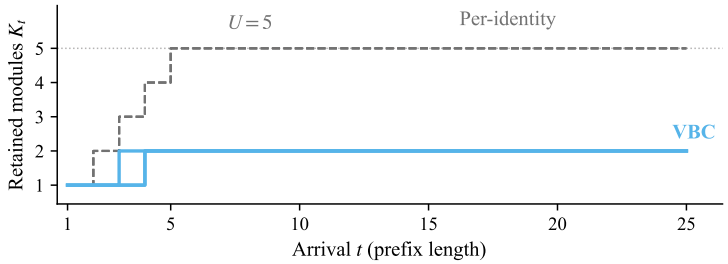
Category	**Method**	**Success** (%)	**State** (MiB)	**Retention approach**
Verified reuse	VTC-2	97.27	**2.03**	Empirical two-slot guard
	VBC, certified	95.77	<u>2.23</u>	Certified adaptive cover
Per-identity specialists	PI-first / PI-best-validated	97.52	3.96	Identity-indexed specialist
	PI-latest	97.46	3.96	Identity-indexed specialist
Structural CL	PNN (dropout-off)	**98.69**	56.32	Parameter expansion
	HiSPO	<u>98.08</u>	10.69	Shared+private subspaces
	TSN-Core	85.75	11.68	Affinity routing
Single net & replay	Finetune	97.79	13.35	Single network
	Scratch-N	97.81	13.38	Retrain from scratch
	Replay-full	94.98	4.02	Full replay buffer
	Replay-fixed128	94.23	4.02	Fixed 128-episode buffer

deterministic greedy heuristic. We qualify each instance by its edge-acceptance rule, writing "VBC, certified" on SimpleTown and "VBC, empirical" on PointMaze and AmazeVille. A controlled MDP isolates compatibility conflicts. Evaluation protocols are in Appendices C, G, and I.

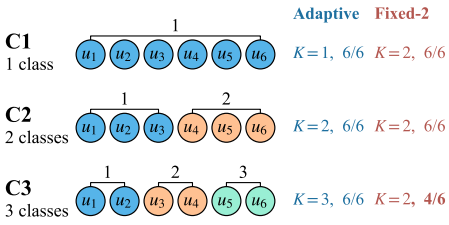
4.1 Do compact libraries remain useful under a common evaluation?

A two-slot library is sufficient for high success on the common SimpleTown benchmark. The two structural baselines with the largest persistent state attain the highest success in Table 2: PNN (dropout-off) reaches 98.69% with 56.32 MiB, and HiSPO 98.08% with 10.69 MiB. Among methods whose retained library occupies about 2 MiB, VTC-2 provides the strongest operating point, reaching 97.27% mean success with 2.03 MiB of persistent state, while per-identity (PI) baselines keep one specialist per identity at 3.96 MiB. The certified adaptive variant (VBC, certified) attains 95.77% with 2.23 MiB. The next subsection decomposes the gap to VTC-2 into certification filtering and online retention. The comparison illustrates the value of matching capacity to the workload, rather than expanding the library by default.

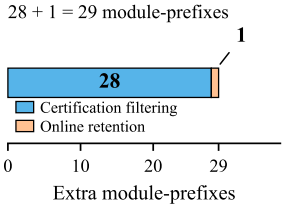
The compact library also persists as tasks recur. Fig. 2a shows that on the balanced stream the library stabilizes at two modules early, rather than becoming compact only at the endpoint. As demonstrated in Appendix Figures 7 and 8, these findings extend to heterogeneous support and persistent state, establishing that compact retention is maintained throughout these streams. Separately, Appendix Table 7 records that an eight-identity stream ends at three modules on every training seed. Only endpoint capacity was recorded for that stream, so it does not enter the trajectory-level claim.



(a) Retained modules on the balanced SimpleTown stream.



(b) Controlled compatibility classes.



(c) Decomposition.

Figure 2: Why a small library suffices, and when it must grow. Per-identity retains one specialist per identity, whereas VBC stabilizes at two. In (b), numbered brackets mark the imposed class structure, alongside measured final capacity and coverage. A fixed two-slot control covers only four of six identities. In (c), extra module-prefixes relative to empirical covers are decomposed into certification filtering and online retention.

4.2 When does adaptive capacity help?

Additional capacity can arise because certification removes empirical reuse edges or because earlier deletions restrict the current candidate pool. We separate these effects using the accounting identity

$$K_{\text{onl}} - K^*_{\text{emp}} = \left(K^*_{\text{cert}} - K^*_{\text{emp}}\right) + \left(K_{\text{onl}} - K^*_{\text{cert}}\right),$$

where K^*_{emp} and K^*_{cert} are minimum cover sizes over the available history under the empirical and certified rules, and differences are summed over prefixes in units of module-prefixes. Fig. 2c shows certification filtering accounts for nearly all additional capacity in the evaluated SimpleTown paths: 28 module-prefixes, compared with one from online retention.

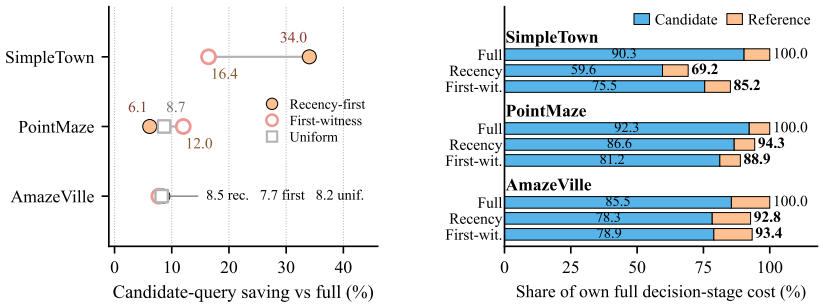
This explains when the two-slot budget becomes restrictive. As decomposed in Appendix Figure 9, it is feasible at all 123 SimpleTown prefixes under empirical validation, but leaves two prefixes infeasible under certification. To test whether behavioral conflicts themselves can require expansion, we turn to a controlled setting.

The controlled MDP assigns six identities to $c \in \{1, 2, 3\}$ compatibility classes, requiring different modules across classes. Fig. 2b shows the imposed structure beside measured capacity and coverage. Both adaptive selectors track the imposed capacity requirement and maintain full coverage, while a fixed two-slot control covers only four of six identities when three classes are present, with the shortfall manifesting as lost coverage rather than incorrect reuse.

4.3 Can we reduce validation without changing retention decisions?

Once the retention rule is fixed, we ask how much evidence it needs. We compare QVBC with full verification using the same source pool, validation outcomes, and exact selector, isolating the effect of query scheduling.

Figure 3 compares this SimpleTown workload with PointMaze and AmazeVille. On the nine SimpleTown pools, recency-first preserves every donor and route decision across all 123 prefixes while reducing candidate episodes by 34.05%. As shown in Appendix Figure 11, the longer balanced stream contributes the majority of the savings, whereas shorter streams yield smaller reductions. Two pre-designed replications, reported in Appendix Table 9, preserve every donor and route decision under all tested orderings, with recency-first remaining the cheapest. Their per-replication savings vary around the primary estimate.



(a) Candidate-query savings. (b) Decision-stage cost.

Figure 3: Validation cost on three backends, normalized by each environment's full-verification workload. In (a), filled markers are queries made while outcomes are still unknown, and open markers replay a completed outcome table. The SimpleTown first-witness marker is the replay matched to the nine-pool workload. The corresponding execution with unknown outcomes has a full candidate workload of 41,900 episodes and appears in Appendix Table 8. In (b), reference episodes are stacked on top of candidate episodes.

Table 3: Capacity–quality trade-off on PointMaze over 60 target–path pairs. Covered-target success evaluates deployed targets. Paired gaps compare against PI-latest on matching resets. State includes checkpoints, reference records, normalizers, caches, and routes. Bold indicates the best value and underline indicates the second-best among full-coverage configurations. For the paired gap, the comparison is among compression configurations, since the PI-latest gap is zero by construction.

	Capacity	**Coverage and quality**			**Storage**
Selector	K	**Coverage (%)**	**Covered success (%)**	**Paired gap (pp)**	**State (MiB)**
PI-latest	5	**100.0**	**74.57**	+0.00	182.61
VBC, empirical	**4**	**100.0**	72.06	-2.12	**146.10**
Greedy	**4**	**100.0**	71.97	**-1.95**	**146.10**
Fixed-1	1	40.0	77.60	-5.79	36.54
Fixed-2	2	46.7	67.93	-10.61	73.06

4.4 Does compact retention preserve quality on an independent backend?

PointMaze tests whether full coverage remains a reliable indicator of quality when the source policies and environment change. We train goal-conditioned implicit Q-learning specialists for five identities of PointMaze-Medium in OGBench (Park et al., 2025), a MuJoCo navigation backend independent of Continual NavBench, with the empirical edge rule applied to 100 paired resets, yielding 12 paths of 15 prefixes across training seeds, arrival orders, and validation pools, as detailed in Appendix G.

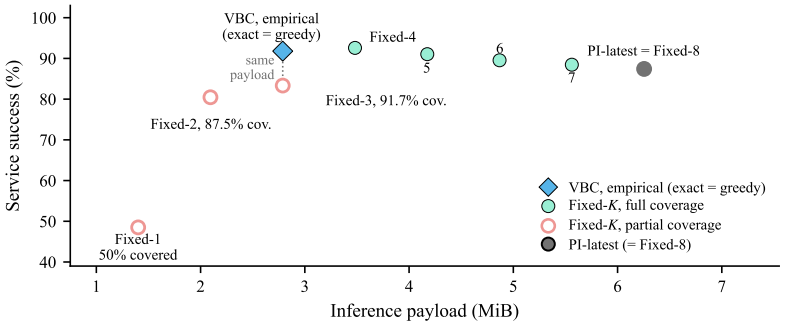
Table 3 shows that adaptive retention maintains complete coverage with a smaller library, at a measurable quality cost. Exact and greedy selection retain four policies instead of five (PI-latest), reducing persistent state by approximately 20%. Their paired fresh success is about two percentage points lower than PI-latest. By contrast, a two-slot control covers fewer than half of the target–path pairs. Adaptive capacity thus avoids the coverage loss of a restrictive budget, while independent evaluation exposes the remaining cost of sharing policies.

Decision preservation carries over to PointMaze, but the scheduling advantage changes. As illustrated in Figure 3 and detailed in Appendix Table 12, all three selective query schedules match full verification at all 180 prefixes. First-witness replay saves 12.01% of candidate episodes, reversing the ordering from SimpleTown where recency-first was cheaper. The stopping criterion remains useful for ensuring correctness even when the most effective schedule changes with the workload.

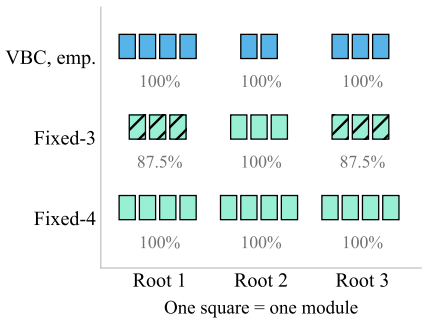
Source quality and recipe selection. Source library quality is critical for effective reuse. Appendix G describes the source construction pipeline, while Appendix Table 11 details the data-interface comparison behind the selected recipe.

Table 4: Capacity–coverage–quality summary of the AmazeVille retention study on the common fresh manifest, averaged over three fixed training roots. Service success scores abstained targets as zero. Paired gaps subtract the per-identity reference of the same root and target. Bold marks the best descriptive value and underline marks the second-best in each column. For capacity, the comparison is among full-coverage configurations. Greedy coincides with VBC, empirical and is omitted. Full eleven-configuration frontier is in Appendix Table 21.

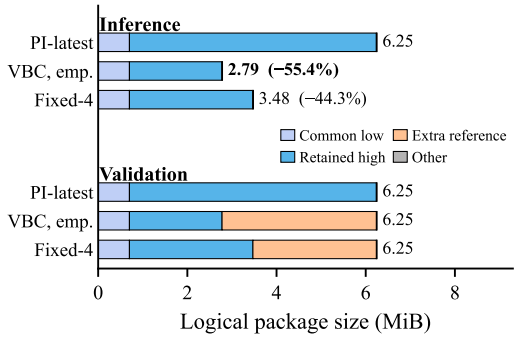
	Capacity	Coverage and quality		
Selector	K	**Coverage (%)**	**Service (%)**	**Gap (pp)**
PI-latest	8.0	**100.00**	87.42	0.00
VBC, empirical	**3.0**	**100.00**	<u>91.79</u>	<u>+4.38</u>
Fixed-3	3.0	91.67	83.33	-4.08
Fixed-4	<u>4.0</u>	**100.00**	**92.58**	**+5.17**



(a) Inference payload and eight-target service success.



(b) Retained modules per root.



(c) Logical package size.

Figure 4: AmazeVille retention on three training roots. In (a), open markers abstain on some targets (counting as zero), while filled markers cover all targets. Adaptive selection coincides with greedy, and PI-latest with Fixed-8. In (b), hatching marks partial coverage across roots. In (c), logical package sizes are separated into inference components, validation references, and bookkeeping.

4.5 Does adaptive retention transfer to AmazeVille?

The SimpleTown and PointMaze studies examine recurrent identities. AmazeVille, a larger 3D first-person environment from the same Continual NavBench (Kobanda et al., 2025a), provides a complementary test: eight distinct identities, each arriving once in a single fixed order, so sharing must rest on cross-identity compatibility. The study uses three training roots, independent end-to-end training runs that vary data collection, environment topology, and shared-policy initialization, together with the shared-low 200k configuration introduced in Appendix H, with empirical validation using 100 paired resets against the first-visit specialist and independent fresh evaluation of 100 episodes per target as specified in Appendix I.

Table 4 summarizes the retention study on a common fresh evaluation, Figure 4 locates every fixed budget on the capacity–quality plane, and Appendix Table 21 reports the complete eleven-configuration frontier. Adaptive retention covers all eight identities while retaining three of the eight

modules on average and improves deployed quality: the exact selector retains three donors on average (four, two, and three across roots), and its eight-target service success of 91.79% exceeds the per-identity reference by 4.38 percentage points. Greedy selection reproduces the exact selector exactly. Fixed-4 also covers every target and attains 92.58% service success, showing a simple fixed budget remains competitive. The value of adaptation lies in finding required capacity without per-stream tuning. As shown in Appendix Table 21, below the required capacity, loss manifests primarily as missing coverage rather than degraded routes.

Retaining three donors instead of eight reduces the inference payload from 6.25 MiB to 2.79 MiB (55.4%), while the full validation state remains at the PI-latest level (Fig. 4c). Decision-preserving querying also transfers: as shown in Figure 3, recency-first reduces candidate episodes by 8.45% while reproducing every retention decision, and first-witness replay saves 7.75%. As reported in Appendix Table 16 and Table 21, this reverses the scheduling ordering observed in PointMaze. Within this configuration, compact adaptive retention, full coverage, improved fresh quality, and reduced payloads hold together, with five of the eight targets served by non-self donors on average.

5 Related Work

Continual reinforcement learning studies how to acquire new behavior without losing earlier capabilities (Khetarpal et al., 2022), typically by protecting capabilities during training through experience replay (Rolnick et al., 2019; Gai et al., 2023; Hu et al., 2025b), parameter regularization (Kirkpatrick et al., 2017), architectural expansion (Rusu et al., 2016), pruning (Mallya & Lazebnik, 2018), or hierarchical policy subspaces (Kobanda et al., 2025b; Gaya et al., 2023). A related line of work connects new tasks to existing components through expert gating (Aljundi et al., 2017), policy composition (Hu et al., 2024), affinity-based transfer (Żurek et al., 2026), multi-game sequence modeling (Lee et al., 2022), and selective weight activation (Hu et al., 2025a), sharing capacity through learned representations or task relationships. We address a complementary stage: after specialists have been trained, behavioral validation determines which policies must remain available. Unlike reuse through learned representations, we evaluate compatibility via observed task success in resettable environments. The resulting storage bound depends on the number of distinct identities rather than the number of arrivals, and whether matching lower bounds (Chen et al., 2022) hold in this setting remains open.

Our validation procedure connects to two further bodies of work. High-confidence policy evaluation (Thomas et al., 2015), uncertainty-aware control (Kahn et al., 2017), safe policy improvement (Laroche et al., 2019), Seldonian learning (Thomas et al., 2019), and SafeAdapt (Anisimov et al., 2026) condition deployment or updates on statistical constraints. Our certified retention variant applies this principle to directed reuse edges, using paired-reset rollouts and Clopper–Pearson intervals (Clopper & Pearson, 1934) to bound uncertainty across a finite family of potential edges. Collecting only the evidence needed for a decision relates to adaptive submodular cover (Golovin & Krause, 2011), interactive set covering (Guillory & Bilmes, 2010), active policy evaluation (Zhang et al., 2020; Anwar et al., 2025), and query algorithms under explorable uncertainty (Erlebach et al., 2016). QVBC applies optimistic and pessimistic bounds to continual policy retention, stopping only when donor and route assignments are guaranteed, which preserves the exact retention path while enabling schedule comparisons.

6 Conclusion

This paper asked how much of a policy library must be kept in continual offline reinforcement learning, and answered the question with behavioral evidence. Paired-reset validation builds a directed compatibility graph, a cover of this graph selects compact libraries under fixed budgets or adaptive capacity, and validation halts once every completion of the unobserved edges yields the same retention decision. Experiments on three navigation environments from two benchmarks show that compact adaptive retention covers every identity while improving or matching the quality of per-identity retention, and that selective querying removes a substantial fraction of validation episodes without altering any decision. Our framework thus answers both what must be kept and when further validation can no longer change that decision.

REFERENCES

Rahaf Aljundi, Punarjay Chakravarty, and Tinne Tuytelaars. Expert gate: Lifelong learning with a network of experts. In *CVPR*, pp. 7120–7129, 2017.

Noga Alon, Baruch Awerbuch, Yossi Azar, Niv Buchbinder, and Joseph Naor. The online set cover problem. *SIAM Journal on Computing*, 39(2):361–370, 2009.

Maksim Anisimov, Francesco Belardinelli, and Matthew Wicker. SafeAdapt: Provably safe policy updates in deep reinforcement learning. *arXiv preprint arXiv:2604.09452*, 2026.

Abrar Anwar, Rohan Gupta, Zain Merchant, Sayan Ghosh, Willie Neiswanger, and Jesse Thomason. Efficient evaluation of multi-task robot policies with active experiment selection. In *CoRL*, pp. 1636–1653, 2025.

Xi Chen, Christos Papadimitriou, and Binghui Peng. Memory bounds for continual learning. In *FOCS*, pp. 519–530, 2022.

Charles J. Clopper and Egon S. Pearson. The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika*, 26(4):404–413, 1934.

Thomas Erlebach, Michael Hoffmann, and Frank Kammer. Query-competitive algorithms for cheapest set problems under uncertainty. *Theoretical Computer Science*, 613:51–64, 2016.

Sibo Gai, Donglin Wang, and Li He. OER: Offline experience replay for continual offline reinforcement learning. In *ECAI*, pp. 772–779, 2023.

Jean-Baptiste Gaya, Thang Doan, Lucas Caccia, Laure Soulier, Ludovic Denoyer, and Roberta Raileanu. Building a subspace of policies for scalable continual learning. In *ICLR*, 2023.

Daniel Golovin and Andreas Krause. Adaptive submodularity: Theory and applications in active learning and stochastic optimization. *Journal of Artificial Intelligence Research*, 42:427–486, 2011.

Andrew Guillory and Jeff Bilmes. Interactive submodular set cover. In *ICML*, pp. 415–422, 2010.

Jifeng Hu, Sili Huang, Li Shen, Zhejian Yang, Shengchao Hu, Shisong Tang, Hechang Chen, Lichao Sun, Yi Chang, and Dacheng Tao. Tackling continual offline RL through selective weights activation on aligned spaces. In *NeurIPS*, 2025a.

Jifeng Hu, Li Shen, Sili Huang, Zhejian Yang, Hechang Chen, Lichao Sun, Yi Chang, and Dacheng Tao. Continual diffuser (COD): Mastering continual offline RL with experience rehearsal. *IEEE Transactions on Neural Networks and Learning Systems*, 36(12):20429–20443, 2025b.

Shengchao Hu, Yuhang Zhou, Ziqing Fan, Jifeng Hu, Li Shen, Ya Zhang, and Dacheng Tao. Continual task learning through adaptive policy self-composition, 2024.

Gregory Kahn, Adam Villaflor, Vitchyr Pong, Pieter Abbeel, and Sergey Levine. Uncertainty-aware reinforcement learning for collision avoidance. *arXiv preprint arXiv:1702.01182*, 2017.

Khimya Khetarpal, Matthew Riemer, Irina Rish, and Doina Precup. Towards continual reinforcement learning: A review and perspectives. *Journal of Artificial Intelligence Research*, 75:1401–1476, 2022.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017. doi: 10.1073/pnas.1611835114.

Anthony Kobanda, Odalric-Ambrym Maillard, and Rémy Portelas. A continual offline reinforcement learning benchmark for navigation tasks. In *CoG*, pp. 1–8, 2025a.

Anthony Kobanda, Rémy Portelas, Odalric-Ambrym Maillard, and Ludovic Denoyer. Hierarchical subspaces of policies for continual offline reinforcement learning. In *ICLR MCDC Workshop*, 2025b.

Romain Laroche, Paul Trichelair, and Rémi Tachet des Combes. Safe policy improvement with baseline bootstrapping. In *ICML*, pp. 3652–3661, 2019.

Kuang-Huei Lee, Ofir Nachum, Mengjiao Yang, Lisa Lee, Daniel Freeman, Winnie Xu, Sergio Guadarrama, Ian Fischer, Eric Jang, Richard Zhang, Yevgen Chebotar, Alexander Herzog, Llion Jones, Dmitry Kalashnikov, Mohammad Norouzi, Peter R. Wurman, and Vikas Sindhwani. Multi-game decision transformers. In *NeurIPS*, volume 35, pp. 27921–27936, 2022.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

Arun Mallya and Svetlana Lazebnik. PackNet: Adding multiple tasks to a single network by iterative pruning. In *CVPR*, pp. 7765–7773, 2018.

Chaofan Pan, Xin Yang, Yanhua Li, Wei Wei, Tianrui Li, Bo An, and Jiye Liang. A survey of continual reinforcement learning. *arXiv preprint*, abs/2506.21872, 2026.

Seohong Park, Kevin Frans, Benjamin Eysenbach, and Sergey Levine. OGBench: Benchmarking offline goal-conditioned RL. In *International Conference on Learning Representations*, 2025.

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy P. Lillicrap, and Gregory Wayne. Experience replay for continual learning. In *NeurIPS*, volume 32, pp. 348–358, 2019.

Andrei A. Rusu, Neil C. Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

Philip S. Thomas, Georgios Theocharous, and Mohammad Ghavamzadeh. High confidence policy improvement. In *ICML*, pp. 2380–2388, 2015.

Philip S. Thomas, Bruno Castro da Silva, Andrew G. Barto, Stephen Giguere, Yuriy Brun, and Emma Brunskill. Preventing undesirable behavior of intelligent machines. *Science*, 366(6468): 999–1004, 2019.

Shun Zhang, Edmund H. Durfee, and Satinder Singh. Querying to find a safe policy under uncertain safety constraints in Markov decision processes. In *AAAI*, volume 34, pp. 2552–2559, 2020.

Dominik Żurek, Kamil Faber, Marcin Pietroń, Paweł Gajewski, and Roberto Corizzo. TSN-Affinity: Similarity-driven parameter reuse for continual offline reinforcement learning. *arXiv preprint arXiv:2604.25898*, 2026.

Appendix

The appendices provide the proofs, protocols, and extended results behind the claims of the main text, and are organized as follows. Appendix A states the statistical assumptions formally, proves the family-wise safety theorem, and examines how far the accepted edges remain from a stronger certificate at the sample sizes used in this work. Appendix B proves the decision-equivalence proposition behind query-efficient validation and compares the exact and greedy selectors on all tested instances. Appendix C specifies the environments, specialist training, and evaluation protocol. Appendix D presents extended retention results, including a decomposition of where additional capacity comes from, and Appendix E accounts for validation cost and replicates the SimpleTown query study on two further pre-designed evaluation runs. Appendix F reports the full comparator results and robustness checks. The remaining appendices extend the study beyond SimpleTown: Appendix G reports the PointMaze retention study, Appendix H describes how the AmazeVille source configuration was developed and selected, and Appendix I gives the complete AmazeVille protocol, the full retention frontier, and the deployment package accounting. Appendix J documents environment access, computational resources, and scope.

A Statistical Guarantees

This section makes the safety guarantee of the verified-reuse family precise. Theorem 1 states that, with probability at least $1-\alpha$, every non-self edge accepted over the stream simultaneously satisfies the paired-loss bound and the success floor. Establishing this statement requires three ingredients: a finite and pre-declared stream that separates training from validation, independently drawn paired resets, and an acceptance rule that spends a fixed error budget over a bounded family of intervals. We state each ingredient as a formal assumption, prove the theorem by a union bound over this family, and then quantify how much stronger the certificate could become at larger sample sizes.

Assumptions and error allocation. The statistical assumptions governing the finite stream, paired-reset design, and acceptance rule are as follows:

Assumption 1 (Finite frozen stream and training-validation separation)**.** *The task stream $(u_1, \ldots, u_T)$ has known finite length T and contains at most U distinct identities. Conditioned on all offline training data, training randomness, and policy parameters, the validation reset variables are independent of training. Validation outcomes never modify policy parameters or the stream sequence.*

Assumption 2 (i.i.d. paired-reset design)**.** *For each target identity u, the paired-reset variables $X_{u,1}, \ldots, X_{u,n_2} \overset{\text{i.i.d.}}{\sim} P_u$ are drawn independently. Each $X_{u,j}$ encapsulates start and goal configurations together with all exogenous randomness needed for candidate and reference rollouts. Candidate and reference outcomes are measurable functions of the same variable $X_{u,j}$. All candidate edges sharing a target identity may use the same random manifest.*

Assumption 3 (Implemented acceptance rule and error allocation)**.** *Given a pre-allocated family-wise error budget $\alpha \in (0,1)$, the allocation is $\alpha_{\text{loo}} = \alpha/(2E_{\max})$ per one-sided Clopper–Pearson interval, with $K_{\max} = U$ and $E_{\max} = T(2U-1)$. A non-self edge e is accepted only if $\text{UCB}_{\text{CP}}(N_e^-, n_2, \alpha_{\text{loo}}) \le \varepsilon_{\text{saf}}$ and $\text{LCB}_{\text{CP}}(S_e, n_2, \alpha_{\text{loo}}) \ge c_{\text{flo}}$, where N_e^- counts paired failures and S_e counts candidate successes over n_2 paired resets.*

With these assumptions in place, simultaneous validity follows from a union bound over the pre-declared family of intervals.

The family-wise error budget partitions α uniformly across $2E_{\max} = 2T(2U-1)$ one-sided Clopper–Pearson intervals. Each accepted non-self edge consumes at most two looks (one for p_-, one for q), so the total number of looks never exceeds the pre-declared bound. Formally, the early stage rejects an edge once its paired failures exceed the largest count whose Clopper–Pearson upper bound on the full n_2 resets would still satisfy the safety tolerance, or once its candidate failures exceed the largest count whose lower bound would still satisfy the competence floor. The n_1=50 reject-only stage therefore does not consume additional α because it never accepts an edge or enlarges the set accepted by the full certified criterion.

Table 5: Implementation evidence and validity conditions for the guarantees in Table 1.

Object	Implementation evidence	Sampling / validity
VBC accepted non-self edge, Theorem 1	Family error α, a reject-only n_1=50 stage, and Clopper–Pearson intervals. A 10 ppempirical filter can only reject. On the edges accepted by the deployed rule, Figure 5 gives $\mathrm{LCB}_\Delta \geq -18.28$ ppat n=100.	Conditional on Assumption 2
QVBC terminal decision, Theorem 2	Exact donor and route agreement on all 123 prefixes of the nine-pool SimpleTown workload	Finite exact-selector instances
VBC retained-state bound	$K \leq U$. Figure 8 shows the balanced-stream storage plateau	Cached identity-pair edges contribute at most a U^2 term, and training remains $\mathcal{O}(T)$

Proof of Theorem 1. Fix any one of the $E_{\max} = T(2U-1)$ potential non-self edges e in the pre-declared family. Under Assumption 2, the paired-failure count N_e^- is binomial with parameters n_2 and $p_-(e)$, and the success count S_e is binomial with parameters n_2 and $q(e)$. The Clopper–Pearson upper bound is an exact one-sided interval, so $\Pr(p_-(e) > \mathrm{UCB}_{\mathrm{CP}}(N_e^-, n_2, \alpha_{\mathrm{loo}})) \leq \alpha_{\mathrm{loo}}$ and $\Pr(q(e) < \mathrm{LCB}_{\mathrm{CP}}(S_e, n_2, \alpha_{\mathrm{loo}})) \leq \alpha_{\mathrm{loo}}$ marginally for each fixed edge. Let G be the event that every one of the $2E_{\max}$ one-sided intervals simultaneously covers its parameter. A union bound over the pre-fixed family gives $\Pr(G) \geq 1 - 2E_{\max} \cdot \alpha_{\mathrm{loo}} = 1 - \alpha$. The bound does not require cross-edge independence, because each interval is marginally exact for its own edge.

Now fix any non-self edge e accepted by VBC. By Assumption 3, this edge passed the full-n_2 acceptance tests, so $\mathrm{UCB}_{\mathrm{CP}}(N_e^-, n_2, \alpha_{\mathrm{loo}}) \leq \varepsilon_{\mathrm{saf}}$ and $\mathrm{LCB}_{\mathrm{CP}}(S_e, n_2, \alpha_{\mathrm{loo}}) \geq c_{\mathrm{flo}}$. On the event G, $p_-(e) \leq \mathrm{UCB}_{\mathrm{CP}}(N_e^-, n_2, \alpha_{\mathrm{loo}}) \leq \varepsilon_{\mathrm{saf}}$ and $q(e) \geq \mathrm{LCB}_{\mathrm{CP}}(S_e, n_2, \alpha_{\mathrm{loo}}) \geq c_{\mathrm{flo}}$. Since $\Delta(e) = p_+(e) - p_-(e) \geq -p_-(e)$, the accepted edge satisfies $\Delta(e) \geq -\varepsilon_{\mathrm{saf}}$ and $q(e) \geq c_{\mathrm{flo}}$ on G. The reject-only n_1 stage never accepts an edge, and the empirical margin filter can only remove candidates, so neither enlarges the accepted set. Under Assumption 1, the subsequent cover and route selection uses only the accepted-edge set. On G, the bounds hold simultaneously for this entire set and hence for any selected subset. □

Table 5 relates the sampling conditions and implementation checks to the conclusions in Table 1. The empirical margin filter at 10 percentage points provides additional rejection, while the certificate remains the one established above.

Certificate gap at current sample sizes. Figure 5 reports conservative two-margin Bonferroni calibration on the 11 non-self edges accepted by the deployed rule. One of these edges is unaccepted under this calibration at n=100 and enters the accepted set from n=200. It is the worst-case curve in the figure. The calibration quantifies how far these edges are from a stronger 10-percentage-point guarantee: even at n=400, the worst-case lower bound is -17.19 percentage points, below the required threshold of -10 percentage points. Six of the eleven edges reach that threshold. Thus, larger samples strengthen some individual certificates without establishing the tighter guarantee for the full family in the studied range.

B Retention and Query Equivalence

This section proves Theorem 2, the decision-equivalence guarantee behind query-efficient validation. The stopping rule compares exact retention decisions on two completions of the partially observed compatibility graph: a pessimistic graph G^- that treats every unresolved edge as absent and an optimistic graph G^+ that treats it as present, giving the sandwich condition $G^- \subseteq G_{\mathrm{true}} \subseteq G^+$ illustrated in Figure 6. Intuitively, if both completions yield the same donors and routes, no further query can change the decision. The proof formalizes this intuition and shows that the query loop always has a witness to resolve and must terminate.

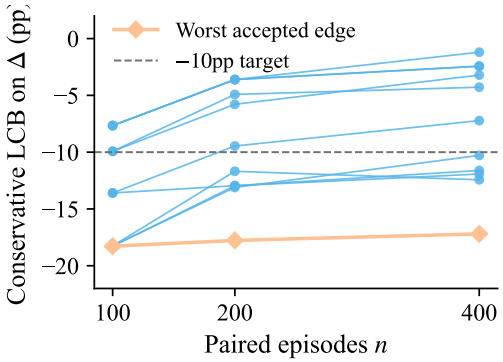
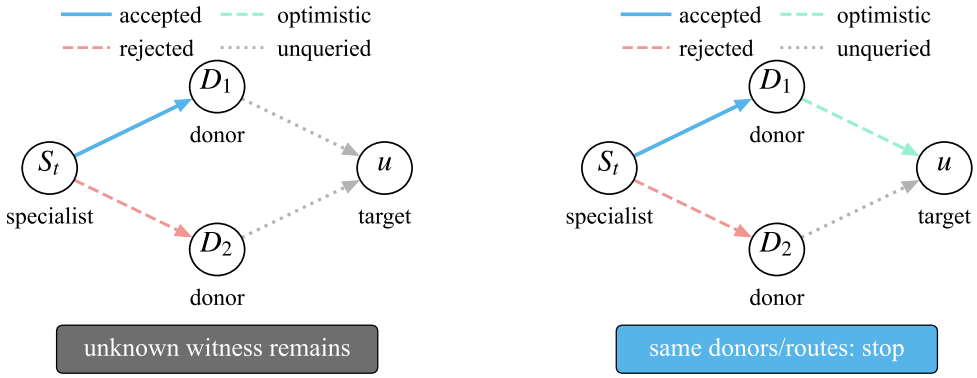


Figure 5: Conservative two-margin lower confidence bounds on the paired success gap Δ for 11 non-self edges at n=100, 200, 400. The target line marks a 10 pp loss bound. One edge is unaccepted under this calibration at n=100 and enters from n=200 as the worst-case curve.



(a) Pessimistic G^-

(b) Optimistic G^+

Figure 6: Schematic of pessimistic G^- and optimistic G^+ graphs used by decision-preserving querying under the sandwich condition $G^- \subseteq G_{\text{true}} \subseteq G^+$. Nodes are the arriving specialist S_t, candidate donors D_1, D_2, and target identity u. Querying terminates when exact donor and route decisions agree on both graphs.

Proof of the decision-equivalence proposition. The argument proceeds in three steps: the loop always has a witness to query, it terminates, and the terminal decision equals the full-revelation decision.

Witness existence. Suppose the loop has not stopped, so the optimistic decision (D^+, ρ^+) differs from the pessimistic decision or the pessimistic graph is infeasible. If every route edge of ρ^+ were already revealed, those edges would be present in G^- with the same accepted labels, so (D^+, ρ^+) would also be the unique exact solution on G^- under the fixed total order and tie-break, contradicting that the loop has not stopped. Hence at least one unrevealed non-self edge lies on the current optimistic routes, and querying it strictly enlarges the revealed set.

Termination. The number of potential non-self edges is bounded by $E_{\max} = T(2U - 1)$, as counted in Section 3, and each query reveals one more edge, so the loop terminates in finitely many steps. The termination condition compares only the final decision objects, so any ordering over the currently unrevealed optimistic route edges preserves the same certificate invariant. The ordering changes only how many edges must be revealed.

Correctness at termination. When the loop stops, the exact selector returns the same ordered donors and the same complete routes on G^- and G^+. Throughout the loop, $G^- \subseteq G_{\text{true}} \subseteq G^+$: the pessimistic graph contains exactly the edges validated as accepted together with all self edges, and the optimistic graph additionally contains every unrevealed edge with its most-optimistic label and an optimistic score that upper-bounds any attainable score of that edge. Because the selector is exact

Table 6: Exact selector versus the deterministic greedy heuristic on the evaluated instances. Cardinality agreement is reported as matched prefixes over evaluated prefixes.

Evaluation scope	**Cardinality K**	**Donors / routes**
Capacity decomposition (123 prefixes)	$123/123$ equal	Donor set differs on one of three seeds of the $T{=}25$ stream
All graph-prefix completions	Equal on every graph-prefix	Routes differ on $17/25$ prefixes of that same seed. Covered-route fresh success was not evaluated
Controlled MDP (18 configurations)	$18/18$ equal, with final $K{=}1, 2, 3$	Final routes identical on $18/18$, with covered-route fresh success $= 1.0$

and the total order and tie-break are fixed independently of the unknown outcomes, its output on G_{true} is sandwiched between the pessimistic and optimistic outputs. Since these two outputs coincide, the full-revelation decision equals the common agreeing decision. Committing this decision leaves the same retained library and route map as full verification, so the subsequent prefix state is identical. □

Exact versus greedy on tested instances. Table 6 compares the exact selector with the deterministic greedy heuristic. The two selectors yield identical retained-module counts on all 123 SimpleTown prefixes and all 18 controlled MDP configurations. Their donor sets differ on one training seed of the balanced stream ($T{=}25$), and completing every graph prefix of that seed records route differences on 17 of 25 prefixes. Equal capacity therefore does not imply equal retention decisions. The main scheduling results use the exact selector as their reference, with greedy selection evaluated as a separate comparator.

C Experimental Setting

This section specifies the environments, specialist training, and evaluation protocol behind the SimpleTown experiments. Table 7 summarizes the evidence scopes and indicates which part of the paper each row supports.

SimpleTown environment and streams. SimpleTown is a Godot-based 3D navigation benchmark with configurable maze topologies. Agents receive first-person observations and must navigate to goal positions. We use three training seeds (100, 200, 300) with pre-generated offline datasets. The benchmark stream follows a fixed sequence of 8 occurrences drawn from $U{=}5$ maze identities on one topology. The balanced stream extends the same identity pool to $T{=}25$ occurrences, while the eight-identity stream enlarges it to $U{=}8$ over $T{=}24$ occurrences. The heterogeneous-support stream uses $T{=}15$ occurrences with fixed per-identity data partitions. Rows (a)–(d) of Table 7 summarize these settings and their evaluation scopes.

Policy backbone and training. SimpleTown specialists share the hierarchical behavior-cloning backbone of the underlying benchmark. A residual multilayer perceptron with input dimension 142, two hidden layers of width 256, GELU activations, layer normalization, and dropout 0.1 produces a seven-dimensional high-level output (the mean and log standard deviation of a three-dimensional relative waypoint and one waypoint-selection logit) and a 16-dimensional low-level output parameterizing eight action coordinates. Each occurrence is trained for $100{,}000$ updates with mini-batches of 64 using Adam (learning rate 3×10^{-4}, cosine annealing, gradient-norm clipping at 1), optimizing the sum of the Bernoulli and diagonal-Gaussian negative log-likelihood losses of the two hierarchy levels. The retained high-level modules referenced throughout the paper are the high-level components of these specialists. All library methods share the same low-level component, which is why persistent state scales with the number of retained modules rather than with the total parameter count. Structural baselines retain their native parameterizations: PNN allocates one column per occurrence, HiSPO maintains shared and private subspaces, TSN-Core (Żurek et al., 2026) routes through affinity scores over frozen modules, and the replay baselines mix historical transitions into the training buffer. The per-identity (PI) baselines keep one specialist per identity and differ only in routing: PI-first routes to the first occurrence of an identity, PI-latest to the most recent, and PI-best-validated to the occurrence with the highest observed validation success. Finetune and Scratch-N

Table 7: Overview of streams and evaluation scopes. Column Scale records the size of each scope (T and U, identity count, or prefix count). Access is written as source / validation / evaluation. — marks an unused stage.

Scope	**Scale**	**Replication**	**Access** (source / val. / eval.)	**Protocol notes**
SimpleTown retention streams				
(a) Benchmark stream (topology 1)	T=8, U=5	3 training seeds	specialists / — / shared manifests	12 methods, with 16 conditions per seed and 48 in total
(b) Balanced stream	T=25, U=5	3 training seeds	specialists / matched / matched	Final $K = [2, 2, 2]$
(c) Eight-identity stream	T=24, U=8	3 training seeds	specialists / paired / matched	Final $K = [3, 3, 3]$, with endpoint capacity only
(d) Heterogeneous-support stream	T=15, U=5	3 training seeds	fixed partitions / — / matched	Final $K = [3, 4, 3]$
Validation matrices and controlled MDP				
(e) Behavioral validation matrices	T=8 or 25, U=5	9 pools	recorded policies / recorded outcomes / —	Offline graph evaluation
(f) Controlled synthetic MDP	6 identities, 3 classes	18 configurations	synthetic / fresh / fresh	Order sensitivity
Validation scheduling				
(g) First-witness scheduling	123 prefixes	9 source pools	specialists / recorded outcomes / —	Executed full validation
(h) Recency-first scheduling	123 prefixes	9 source pools	specialists / fresh / —	Episode-weighted workload
Independent domains and backends				
(i) PointMaze retention	T=15, U=5	3 training seeds	specialists / paired n=100 / paired n=200	2 orders $\times$ 2 pools per seed, giving 180 prefixes
(j) AmazeVille development	8 identities, 4 configurations	5 training roots	specialists / self n=400 / —	Source selection only
(k) AmazeVille retention	8 identities, 11 configurations	3 fixed roots	specialists / paired n=100 / paired n=100	1 arrival order, shared fresh evaluation, and 66 packages

are single-network controls that either continue training one network or retrain from scratch. The dropout-off flag for PNN denotes the deterministic evaluation adapter of that baseline, not a separate method.

Evaluation protocol. Fresh evaluation on the benchmark stream compares each method on 16 test conditions per training seed (8 target occurrences paired with 2 fresh rollout manifests), giving 48 test conditions per method across three seeds, with 100 episodes per condition. Validation and fresh evaluation use distinct random manifests with non-overlapping seeds. Training, validation, and evaluation data are strictly separated. Certified acceptance of a non-self edge uses n_2=100 paired resets per edge with candidate and reference rollouts on shared initial conditions, n_1=50 as a reject-only early stage, α=0.10, ε_{saf}=0.20, and c_{flo}=0.80, matching Assumption 3. Candidate and reference episodes of a paired reset share the same initial configuration. A paired failure is counted only when the reference succeeds and the candidate fails, matching $d = -1$ in Section 3.

D Retention Breadth and Mechanism

A plateau in module count is useful only if the other retained state also remains bounded. Figure 8 shows the corresponding storage plateau for the balanced stream. Figure 7 gives the heterogeneous-support trajectory over all 15 arrivals. To explain departures from a smaller cover, Figure 9 compares empirical, certified, and online capacity across nine behavioral matrices. Figure 10 then specifies the controlled conflicts used to test capacity adaptation.

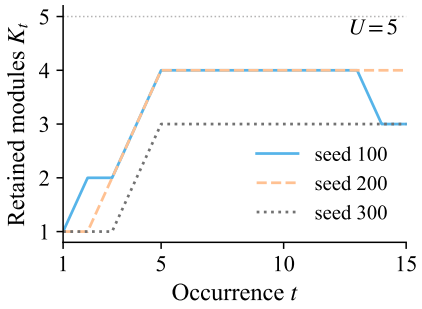


Figure 7: Online retained modules on the heterogeneous-support SimpleTown stream (T=15, U=5), one curve per training seed. The dotted line marks the identity count.

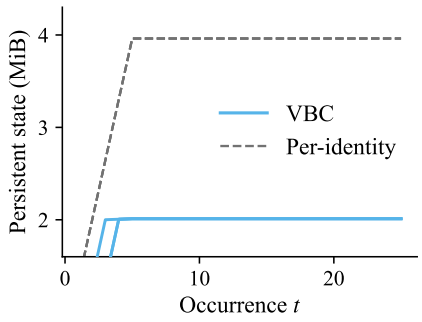
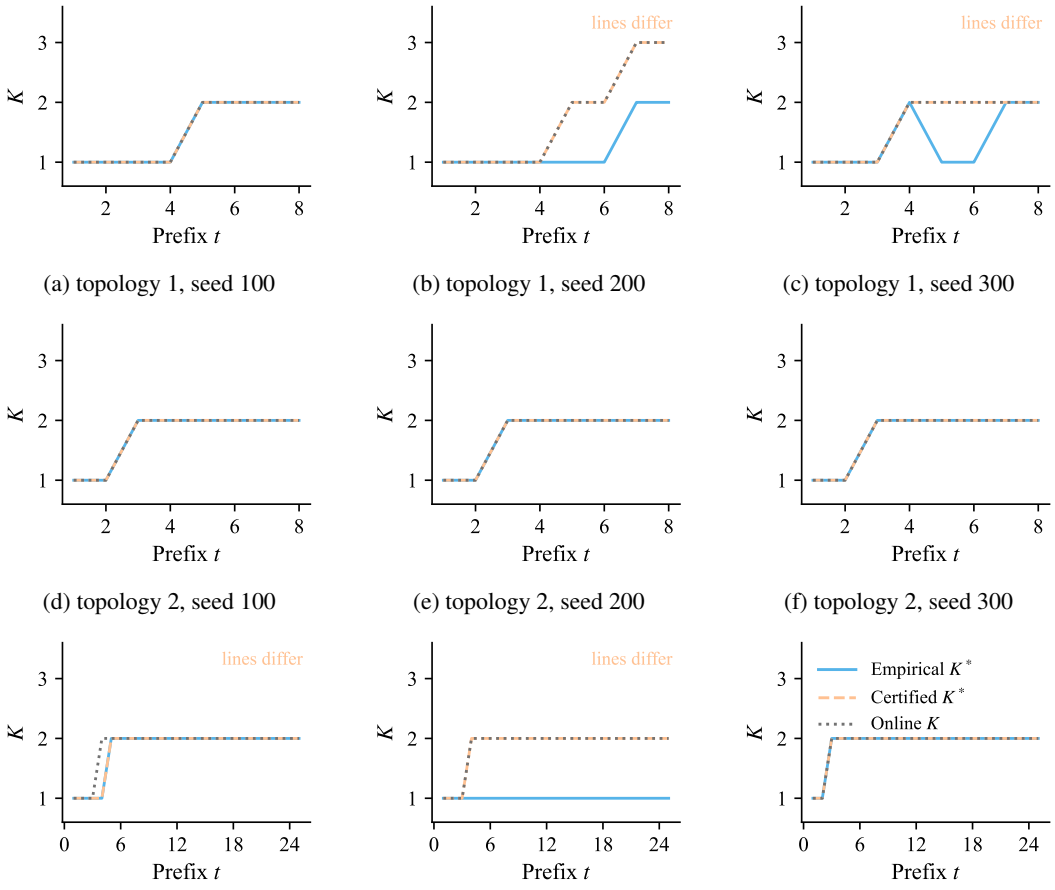


Figure 8: Balanced-stream prefix trajectory of persistent state for VBC (solid) and per-identity (dashed), matching the K_t path in Figure 2. The three training seeds coincide after t=4. The persistent plateau is 2.0112 MiB after prefix 10. This uses a separate matched-reset evaluation, distinct from the shared benchmark comparison in Table 2.

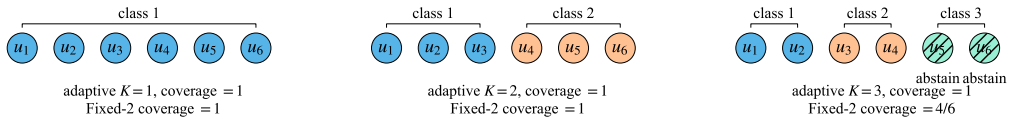


(a) topology 1, seed 100 (b) topology 1, seed 200 (c) topology 1, seed 300

(d) topology 2, seed 100 (e) topology 2, seed 200 (f) topology 2, seed 300

(g) balanced stream, seed 100 (h) balanced stream, seed 200 (i) balanced stream, seed 300

Figure 9: Cover cardinalities on nine behavioral matrices with $U{=}5$. Empirical K^* (solid), certified K^* (dashed), and online K (dotted) are all-history or prefix-local covers as labeled in the bottom-right panel. The three series coincide on most prefixes. Panels marked "lines differ" are the matrices in which certification or the online path increases K. Integer prefix ticks are shown on every panel.



(a) One class (C1) (b) Two classes (C2) (c) Three classes (C3)

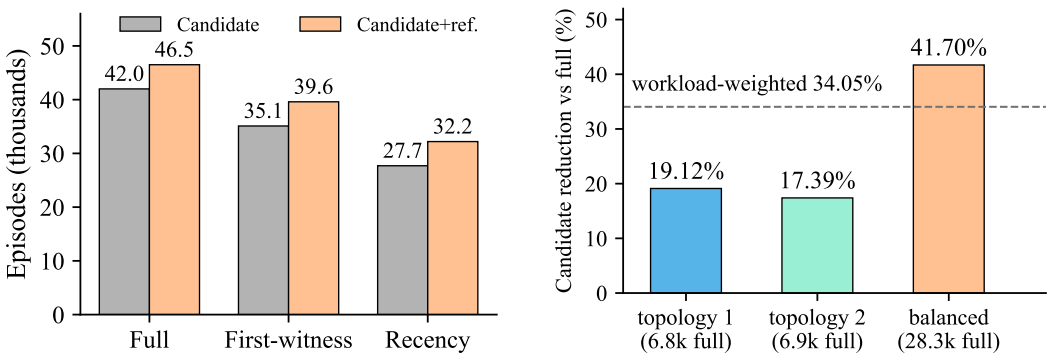
Figure 10: Compatibility structure of the controlled finite MDP. Circles denote identities $u_1, \ldots, u_6$, grouped by action-decoding class using color and braces. Identities in different classes cannot share a module, so full coverage requires at least one module per class. Hatched identities in the three-class panel are the abstentions of the fixed two-slot control on the primary arrival order. The secondary order abstains on a different pair, and coverage remains four of six in every run.

E VALIDATION WORKLOAD

This section details the query workloads behind Figure 3. Figure 11 separates the SimpleTown candidate counts by schedule and by stream, and Figure 12 reports per-pool candidate episode counts, showing that the balanced stream contributes the largest share of the workload.

Table 8 collects the executed nine-pool schedules and the supplementary offline accounts. Candidate counts are reported as queried over total potential episodes. The execution column distinguishes queries made while outcomes are still unknown from replays of a completed outcome table and from queries that reuse stored certificates.

Replication of the SimpleTown query study. The nine-pool query comparison was replicated twice under fresh evaluation randomness, with the replication design fixed in advance. Table 9 reports the two replications separately and combined. Alongside recency-first and first-witness, the replication evaluates a third ordering, most-unknown, which first queries witnesses of the identity with the largest number of unrevealed non-self edges. Both replications preserve the complete donor and route decisions of full verification under all three orderings and both verifiers. Recency-first remains the cheapest ordering throughout, and the certified verifier shows wider variation in savings across replications than the empirical verifier. The two replications are nested repetitions over the same nine trained policy pools, not new training replicates, so they quantify the stability of the savings rather than extending them to a larger population of pools.



(a) Episodes by schedule

(b) Reduction vs full by stream

Figure 11: Validation workload on nine SimpleTown policy pools spanning 123 prefixes. (a) Candidate and candidate-plus-reference episode counts by schedule. Recency-first queries initially unknown outcomes, while first-witness replays the same completed outcome table. (b) Candidate-query reductions by stream. Bars have equal width. The full candidate workload of each stream is printed under its label, and the dashed line is the workload-weighted aggregate.

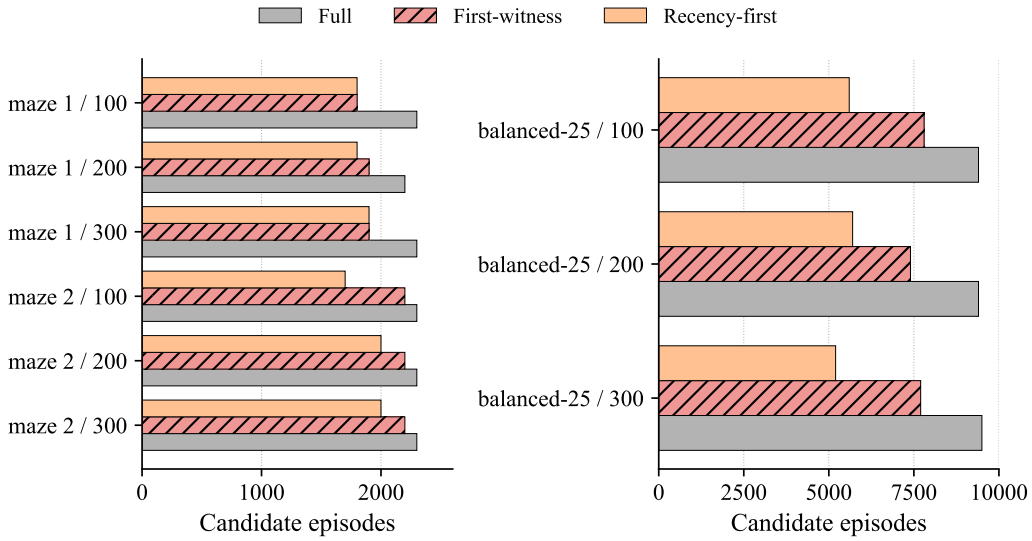


Figure 12: Per-pool candidate episode counts over 123 prefixes. The six short-stream pools and three balanced-stream pools use separate horizontal scales. First-witness replays the same completed outcome table used to assess recency-first.

Table 8: Validation workloads. Candidate and decision-stage columns report queried/total episodes with reduction percentages in parentheses. The execution column distinguishes queries made while outcomes are still unknown from replays of a completed outcome table and from queries that reuse stored certificates. Bold marks the best descriptive value and underline marks the second-best within the nine-pool workload. One first-witness execution with unknown outcomes has a full candidate workload of 41,900 episodes, 100 fewer than the matched nine-pool account.

Schedule	Execution	Candidate episodes	Decision-stage
Nine-pool workload			
Recency-first	unknown outcomes	27,700/42,000 **(34.05%)**	32,200/46,500 **(30.75%)**
	stored certificates	31,600/42,900 <u>(26.34%)</u>	36,100/47,400 <u>(23.84%)</u>
First-witness	unknown outcomes	34,100/41,900 (18.62%)	38,600/46,400 (16.81%)
	matched replay	35,100/42,000 (16.43%)	39,600/46,500 (14.84%)
Supplementary accounts (separate from the nine-pool totals)			
Empirical offline	recorded outcomes	33,600/42,200 (20.38%)	n/a
Certified offline	recorded outcomes	28,050/33,750 (16.89%)	n/a
Controlled MDP	fresh synthetic	54,450/114,150 (52.30%)	65,250/124,950 (47.78%)

Table 9: Query savings relative to full verification in two SimpleTown replications of the nine-pool study, each using 100 episodes per queried edge. Candidate savings exclude reference episodes. Decision savings include them. All schedules preserve every donor and route decision. Most-unknown is evaluated only in these replications and is not part of the primary comparison. Bold indicates the best savings and underline indicates second-best in each column.

		Candidate savings (%)			Decision savings (%)		
Ordering	**Verifier**	**Rep. A**	**Rep. B**	**Both**	**Rep. A**	**Rep. B**	**Both**
Recency-first	Empirical	**36.73**	**36.04**	**36.39**	**33.19**	**32.54**	**32.87**
	Certified	34.27	24.48	29.37	31.01	22.15	26.58
First-witness	Empirical	21.33	20.05	20.69	19.27	18.32	18.80
	Certified	19.11	9.56	14.34	17.30	8.65	12.97
Most-unknown	Empirical	<u>34.36</u>	<u>31.98</u>	<u>33.17</u>	<u>31.05</u>	<u>28.88</u>	<u>29.97</u>
	Certified	30.07	22.84	26.46	27.22	20.68	23.95

F COMPARATORS AND ROBUSTNESS

This section expands the benchmark comparison of Table 2. Table 10 reports per-seed success, exact persistent bytes, online drop, and validation episodes for every method. Figure 13 presents the paired intervals of the replay comparison under matched total-persistent storage, complementing the replay rows of Table 2.

Table 10: Per-seed persistent state (MiB for seeds 100/200/300), online drop (pp), and validation episodes on the SimpleTown benchmark stream. Success is the mean over the three training seeds. Single values in the state and drop columns indicate identical metrics across seeds. Drop is the largest fresh-success loss of an online deployed route relative to its identity reference, taken over prefixes and targets. Bold marks the best value and underline marks the second-best for success and state. PI-first and PI-best-validated coincide on displayed fields. The dash for VTC-2 marks validation episodes not recorded in this comparison. Entries marked n/a are undefined for that method.

Category	**Method**	**Success** (%)	**State** (MiB)	**Drop** (pp)	**Val. episodes**
Verified reuse	VTC-2	97.27	**2.03/2.04/2.02**	7	—
	VBC, certified	95.77	<u>2.01/2.67/2.01</u>	7/0/6	2,800/2,800/2,900
Per-identity specialists	PI-first / PI-best-validated	97.52	3.96	n/a	0
	PI-latest	97.46	3.96	3/1/1	0
Structural CL	PNN (dropout-off)	**98.69**	56.32	n/a	n/a
	HiSPO	<u>98.08</u>	10.69	n/a	n/a
	TSN-Core	85.75	11.68	n/a	n/a
Single net & replay	Finetune	97.79	13.35	n/a	n/a
	Scratch-N	97.81	13.38	n/a	n/a
	Replay-full	94.98	4.02	n/a	n/a
	Replay-fixed128	94.23	4.02	n/a	n/a

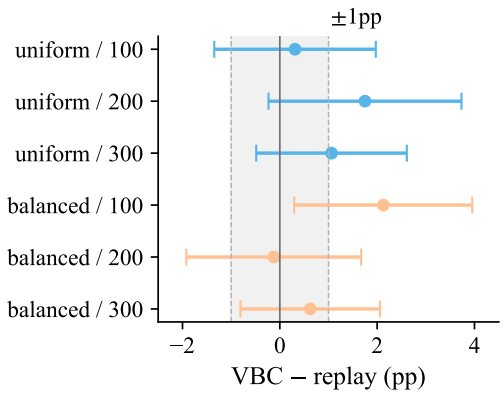


Figure 13: Simultaneous paired confidence intervals for VBC−replay under matched total-persistent storage and paired benchmark evaluation manifests. Positive values indicate higher success for VBC. Rows differ in the weighting of evaluation cells (uniform across cells, or balanced across identities) and in training seed (after the slash). The shaded band marks a ± 1 ppvisual scale. Intervals that include zero leave the comparison inconclusive.

G PointMaze: Retention Beyond SimpleTown

Sources and recurrence. We use the five task identities of PointMaze-Medium in OGBench (Park et al., 2025), a MuJoCo navigation environment. Each arrival exposes a researcher-constructed, task-specific static offline dataset, rather than a partition of the published benchmark dataset. Data are collected with the benchmark waypoint oracle, Gaussian action noise of standard deviation 0.5, and actions clipped to $[-1, 1]$. Each training shard contains at least 100 complete episodes and 10,000 valid transitions, retaining both successful and unsuccessful trajectories. Goal-conditioned implicit Q-learning trains each specialist for one million updates without online policy optimization, using a learning rate of 3×10^{-4}, batch size 1024, three hidden layers of width 512, discount 0.99, expectile 0.9, and actor regularization coefficient 0.003. Evaluation uses deterministic actions, a 1000-step horizon, and termination on reaching the goal. The source recipe is selected on separate development seeds by maximizing the minimum identity-level mean self success, followed by the overall mean. This selects the original data interface. The retention study then trains 45 specialists with three independent training seeds, five identities, and three visits per identity. Arrival orders reuse these same sources: one repeats the sequence $(1, 2, 3, 4, 5)$ three times, while the other begins

Table 11: Effect of including terminal successors in the PointMaze source-data interface. Differences are paired self-success gaps (pp) relative to the original interface over 15 seed–identity pairs (100 episodes per source). Bold indicates the superior interface for that learner.

Learner	Mean gap (pp)	Positive	Tied	Negative
GCBC	**+10.80**	**14**	1	**0**
GCIQL	-6.80	5	2	8

Table 12: PointMaze query workloads over 12 paths and 180 prefixes. Recency-first begins with unknown outcomes. First-witness and uniform witness replay the completed outcome table. All schedules preserve full-verification decisions. Bold indicates best and underline indicates second-best (lower episodes, higher savings).

	Query episodes		Savings relative to full	
Schedule	**Candidate**	**Candidate + ref.**	**Cand. saving (%)**	**Total saving (%)**
Full	71,600	77,600	0.00	0.00
Recency-first	67,200	73,200	6.15	5.67
First-witness	**63,000**	**69,000**	**12.01**	**11.08**
Uniform witness	<u>65,400</u>	<u>71,400</u>	<u>8.66</u>	<u>7.99</u>

with that sequence and then groups the remaining visits by identity. Each seed has two validation pools shared across orders. Orders and pools do not create additional independent training replicates.

Source-data intervention. The interface comparison tests whether making terminal successor states available improves source learning. It crosses the original and terminal-successor-inclusive interfaces with goal-conditioned behavior cloning (GCBC) and goal-conditioned implicit Q-learning (GCIQL), using three development seeds, all five identities, and one million updates per source. Paired self evaluation uses 100 episodes per source. Table 11 shows that the effect depends on the learner. GCBC improves in almost every seed–identity pair, whereas GCIQL has lower mean self success despite gains on some pairs. This contrast supports assessing the interface together with the learning algorithm when selecting source recipes.

Validation and retention. Non-self edges use the empirical rule in Section 3, with 100 paired resets and the first-visit specialist as the reference for each identity. Self edges retain the option to deploy the corresponding specialist on its own identity. The exact selector first minimizes the number of donors, then maximizes the worst route score, and finally uses a deterministic donor tie-break. Recency-first queries initially unknown outcomes along each path. Once the path is fixed, full-table completion supplies the outcomes needed by the other selectors. First-witness and uniform-witness schedules replay this table. The uniform schedule samples from the currently legal witness set at each step. Table 12 reports their workloads and confirms agreement with full verification on donors, complete routes, and the resulting next-prefix library and route state at all 180 prefixes.

The per-seed breakdown in Table 13 explains the modest aggregate gain: recency-first avoids queries for one training seed and uses the full candidate workload for the other two. The replay schedules quantify how another legal query order would traverse the same outcomes. Their counts measure potential episode savings rather than wall-clock speedups.

Fresh quality, abstention, and storage. All terminal routes are fixed before independent fresh evaluation. Each deployed target receives 200 paired candidate and PI-latest episodes. Reset sets are specific to the path, retention configuration, and target, so the paired gaps in Table 3 subtract PI-latest re-evaluated on that same set. This within-reset pairing isolates the success difference between deployed policies. Table 3 shows full coverage for VBC, empirical and greedy selection, while fixed-budget controls abstain on some targets. Their success rates therefore condition on the deployed subset, with coverage accounting separately for the unserved target–path pairs. Persistent bytes measure retained checkpoint files, reference records, normalizers, caches, routes, and metadata. They are distinct from retained-policy count and from an inference-only model size.

Table 13: PointMaze candidate query episodes by training seed, summed over two orders and two validation pools. Bold highlights the seed with substantial query savings.

Training seed	Full	Recency-first	Saving (%)
2027260841	24,000	24,000	0.00
2149419016	24,000	24,000	0.00
3527082596	23,600	**19,200**	**18.64**

Table 14: AmazeVille source development across five training roots and eight identities (400 self-evaluation episodes per source). Mean ± SD summarizes root-level identity averages. The weakest-identity column takes the minimum after averaging across roots. ⋆ marks the selected shared-low recipe. Task-local configurations are diagnostic controls. Bold indicates overall best and underline indicates second-best.

Low-level policy	Updates	Self success (%)	Weakest identity (%)
Shared	100k	83.92 ± 2.08	71.30
	200k⋆	85.41 ± 1.30	<u>73.70</u>
Task-local	100k	<u>86.59 ± 1.12</u>	69.20
	200k	**88.19 ± 0.79**	**76.60**

H AMAZEVILLE SOURCE DEVELOPMENT

Factorial comparison. The source-development study covers all eight AmazeVille identities using five training roots. It compares a shared low-level policy with task-local low-level policies at 100k and 200k updates per arrival. The shared configuration learns its low-level policy on the first identity and freezes it, adding a high-level module for each subsequent identity. The task-local control trains a separate high- and low-level policy for each identity. Both use the hierarchical behavior-cloning backbone. Increasing duration also changes the cosine learning-rate schedule to the corresponding horizon, so the comparison concerns a duration-and-schedule configuration rather than an isolated update-count intervention.

Each source is evaluated on 400 episodes with a 300-control-step horizon. The complete matrix contains 160 source conditions and 64,000 episodes. Within each root and low-level configuration, both durations use the same data and self-evaluation resets. We average equally over identities within each root and then over the five roots. The standard deviation in Table 14 is computed across those five root means, not across episodes. Table 15 gives the individual root means and Figure 14 shows the identity-level variation.

Selection and interpretation. The selection rule is fixed in advance and restricted to the shared-low family: maximize the weakest identity after averaging across roots, then the mean over all identities, with shorter duration breaking any remaining tie. Table 14 shows that the 200k configuration improves both the weakest-identity mean and overall self success within that family. Task-local policies provide a diagnostic comparison of low-level sharing, while the selection rule keeps the architecture fixed. The study therefore identifies a source recipe for the chosen evaluation horizon, leaving cross-identity reuse to a separate retention evaluation.

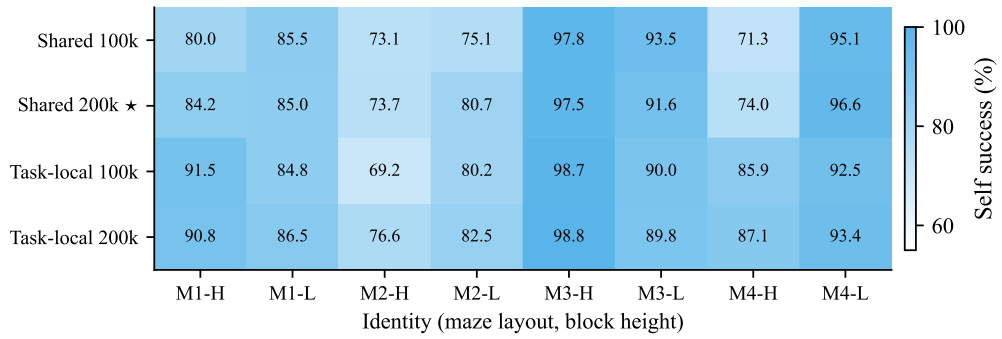


Figure 14: Identity-level self success (%) in AmazeVille source development, averaged over five training roots. Each cell is one configuration and one categorical identity. M1–M4 denote maze layouts, and H/L denote high or low blocks. The starred row is the shared-low 200k recipe selected within that family.

Table 15: Per-root AmazeVille development self success (%), averaged equally over eight identities across the same five training roots. Bold marks the best value and underline marks the second-best in each root.

Policy	Updates	Root 1	Root 2	Root 3	Root 4	Root 5
Shared	100k	79.81	84.28	85.19	85.22	85.09
	200k	83.47	<u>85.12</u>	87.44	86.03	85.00
Task-local	100k	<u>85.72</u>	84.88	**87.81**	<u>87.56</u>	<u>87.00</u>
	200k	**88.38**	**87.12**	<u>87.59</u>	**88.41**	**89.44**

I AmazeVille Retention Protocol and Detail

Scope and sources. The AmazeVille retention study uses three fixed training roots, a single fixed arrival order of the eight identities, and the final prefix at which all eight identities have arrived exactly once. Every target in the evaluation is therefore a distinct identity, and every non-self route crosses an identity boundary. Specialists use the hierarchical behavior-cloning backbone with the shared-low configuration trained for 200k updates per arrival, selected by the source-development study of Appendix H. The three roots are fixed design points of this study, so all summaries below average equally over roots and make no population-level claim over training seeds, arrival orders, or validation pools.

Validation and query schedules. Validation compares each candidate donor with the first-visit specialist of the target identity over 100 paired resets per edge, giving 64 paired cells per root and 192 cells in total. Because each identity arrives once, the first-visit reference coincides with PI-latest on every target. Recency-first querying begins with unknown outcomes. First-witness and uniform-witness schedules replay the completed outcome table, the latter averaged over 128 legal-order replicates, so their episode counts measure potential rather than realized savings. All three selective schedules match the donor and route decisions of full verification at every prefix of every root. Table 16 reports the per-root workloads. Candidate counts exclude the eight fixed reference evaluations per root, and the total-saving column includes them.

Retention configurations. Eleven retention configurations are replayed on the completed validation table of each root: VBC, empirical and greedy selection, fixed budgets from one to eight slots, and PI-latest. Table 17 reports retained donors and coverage for every configuration. Greedy selection matches the exact selector in both capacity and coverage on all three roots. The VBC, empirical configuration retains four, two, and three donors across the roots and covers all targets in each case, while Fixed-2 abstains on targets 6 and 8 of the first root and target 4 of the third. One fresh manifest evaluates PI-latest, VBC, empirical, and Fixed-2. A second manifest evaluates all eleven configurations.

Table 16: AmazeVille candidate query episodes per root, excluding the eight fixed reference evaluations (800 episodes per root). Recency-first begins with unknown outcomes. First-witness and uniform witness replay the completed outcome table. All schedules preserve full-verification decisions. Bold marks best and underline marks second-best (lower episodes, higher savings).

	Query episodes per root			Savings relative to full	
Schedule	**Root 1**	**Root 2**	**Root 3**	**Cand. saving (%)**	**Total saving (%)**
Full	4,700	4,600	4,900	0.00	0.00
Recency-first	**4,400**	**4,000**	4,600	**8.45**	**7.23**
First-witness	4,500	4,100	**4,500**	7.75	6.63
Uniform witness	4,451.56	4,044.53	4,533.59	8.24	7.05

Table 17: Retained donors (K) and target coverage of all eleven retention configurations replayed on the completed validation table of each root. All query schedules yield identical decisions. Bold highlights 100% coverage and the minimum capacity required to attain it on each root.

	Capacity K			Coverage (%)		
Configuration	**Root 1**	**Root 2**	**Root 3**	**Root 1**	**Root 2**	**Root 3**
VBC, empirical	**4**	**2**	**3**	**100.0**	**100.0**	**100.0**
Greedy	**4**	**2**	**3**	**100.0**	**100.0**	**100.0**
Fixed-1	1	1	1	50.0	50.0	50.0
Fixed-2	2	**2**	2	75.0	**100.0**	87.5
Fixed-3	3	3	**3**	87.5	**100.0**	87.5
Fixed-4	**4**	4	4	**100.0**	**100.0**	**100.0**
Fixed-5	5	5	5	**100.0**	**100.0**	**100.0**
Fixed-6	6	6	6	**100.0**	**100.0**	**100.0**
Fixed-7	7	7	7	**100.0**	**100.0**	**100.0**
Fixed-8	8	8	8	**100.0**	**100.0**	**100.0**
PI-latest	8	8	8	**100.0**	**100.0**	**100.0**

Fresh evaluation and accounting. Routes are fixed before fresh evaluation begins. Each deployed route receives 100 fresh episodes per root–target pair, with evaluation randomness independent of validation. Content-identical policy and context combinations are deduplicated before execution, so 40 physical measurements carry the 93 logical configuration–route assignments. Deduplication reduces execution cost but does not change the number of root–target conditions. Table 18 summarizes the three-configuration manifest, Table 19 breaks it down by root, and Table 20 with Figure 15 gives the per-target detail. VBC, empirical improves over the reference on 13 pairs, ties on 9, and falls below it on 2, with gaps from -13 to $+18$ percentage points. For Fixed-2, success and paired gaps condition on covered targets, and the coverage column accounts separately for abstentions. A coverage-adjusted yield that scores abstained targets as zero gives 80.42% for Fixed-2. We report it only as a diagnostic, since the study designates coverage and covered-target success as the joint criteria rather than a single scalar.

Frontier on the eleven-configuration manifest. A second fresh manifest evaluates all eleven retention configurations, with 100 episodes per root–target pair and evaluation randomness independent of both validation and the three-configuration manifest. Table 21 reports the complete frontier, and Table 22 gives the per-root detail. VBC, empirical and greedy coincide on every root, with four, two, and three retained donors. Fixed-4 attains the highest service success on every root, so the descriptive advantage of adaptive selection over a well-chosen fixed budget does not hold on these three roots. VBC, empirical improves over the per-identity reference on 11 of 24 root–target pairs, ties on 10, and falls below it on 3, with a root-equal mean gap of $+4.38$ percentage points, while the three-configuration manifest gave $+5.13$ percentage points with 13 positive, 9 tied, and 2 negative pairs. The two manifests are reported separately because they are independent evaluation draws over the same frozen routes, not two samples of a larger training population. Abstained targets count as zero in service success, so the low-capacity configurations Fixed-1 to Fixed-3 rank far below their covered-target quality. Their high covered-target success shows that insufficient capacity manifests as lost coverage rather than degraded routes.

Table 18: Capacity–coverage–quality summary of the AmazeVille retention study on the three-configuration fresh manifest, over three fixed training roots and eight targets. Success averages 100-episode evaluations over deployed targets. Paired gaps subtract the first-visit reference of the same root and target. *Fixed-2 conditions on covered targets. Bold marks the best value among full-coverage configurations.

	Capacity	Coverage and quality		
Selector	K	**Coverage (%)**	**Success (%)**	**Paired gap (pp)**
PI-latest (first-visit reference)	8	**100.0**	86.29	0.00
VBC, empirical	**3**	**100.0**	**91.42**	**+5.13**
Fixed-2	2	87.5	91.87*	+4.92*

Table 19: Per-root fresh results of the AmazeVille retention study on the three-configuration manifest. Success averages fresh 100-episode evaluations over deployed targets within each root. Paired gaps subtract the per-target reference of the same root. *Fixed-2 conditions on covered targets. Bold marks the best value and underline marks the second-best in each root.

		Capacity	Coverage and quality		
Root	**Selector**	K	**Coverage (%)**	**Success (%)**	**Paired gap (pp)**
Root 1	Reference	8	**100.0**	85.50	0.00
	VBC, empirical	**4**	**100.0**	**91.75**	**+6.25**
	Fixed-2	2	75.0	91.50*	+5.17*
Root 2	Reference	8	**100.0**	86.50	0.00
	VBC, empirical	**2**	**100.0**	**92.25**	**+5.75**
	Fixed-2	2	**100.0**	**92.25***	**+5.75***
Root 3	Reference	8	**100.0**	86.88	0.00
	VBC, empirical	**3**	**100.0**	<u>90.25</u>	<u>+3.38</u>
	Fixed-2	2	87.5	**91.86***	**+3.86***

Deployment packages and byte accounting. For each root and configuration we assembled two deployable packages. The inference package contains the shared low-level policy, the retained high-level modules, the route map, normalizers, and a small runtime loader. The validation-retention package additionally keeps every first-visit reference module, so all eleven configurations retain eight high-level modules under this accounting and the payload stays at the PI-latest level. Table 23 breaks down representative packages by component. The low-level policy and the per-module high-level components dominate, while routing and bookkeeping records together remain below 16 KiB. All 66 packages were loaded in an isolated environment without access to the training checkpoints or the shared file systems, and their exported modules reproduced the source forward outputs within a maximum absolute error of 10^{-6}. The packages rely on the external Python and Torch runtime and are not self-contained executables. Logical payload counts every component of a package in full. A content-addressed store that hard-links identical blobs across packages reduces archival footprint but is not credited to any single package.

Table 20: Fresh success (%) per root and target on the three-configuration manifest of the AmazeVille retention study (100 episodes per root–target pair). Reference is the first-visit specialist of each identity. Δ gives paired gaps in percentage points against that reference. Fixed-2 abstains on marked targets. Bold marks the best non-reference success and gap on each target.

		Ref.	VBC, empirical		Fixed-2	
Root	**Target**	**Success**	**Success**	**Δ**	**Success**	**Δ**
Root 1	1	83	**100**	**+17**	**100**	**+17**
	2	82	**93**	**+11**	74	-8
	3	84	**100**	**+16**	**100**	**+16**
	4	77	**77**	**0**	**77**	**0**
	5	92	**98**	**+6**	**98**	**+6**
	6	92	**92**	**0**	—	—
	7	100	**100**	**0**	**100**	**0**
	8	74	**74**	**0**	—	—
Root 2	1	79	**93**	**+14**	**93**	**+14**
	2	89	**91**	**+2**	**91**	**+2**
	3	80	**98**	**+18**	**98**	**+18**
	4	73	**89**	**+16**	**89**	**+16**
	5	87	**96**	**+9**	**96**	**+9**
	6	98	**85**	**-13**	**85**	**-13**
	7	96	**96**	**0**	**96**	**0**
	8	90	**90**	**0**	**90**	**0**
Root 3	1	83	**94**	**+11**	**94**	**+11**
	2	83	**88**	**+5**	**88**	**+5**
	3	86	**96**	**+10**	**96**	**+10**
	4	79	**79**	**0**	—	—
	5	91	**97**	**+6**	**97**	**+6**
	6	94	**89**	**-5**	**89**	**-5**
	7	96	**96**	**0**	**96**	**0**
	8	83	**83**	**0**	**83**	**0**



Figure 15: Paired fresh gaps (pp) of VBC, empirical against the first-visit reference on the three-configuration fresh manifest, for every root–target pair. Every cell is a measured gap. A printed zero is a tie, not a missing evaluation. Fixed-2 abstentions are omitted from this figure and listed in Table 20.

Table 21: Complete eleven-configuration frontier of the AmazeVille retention study on the common fresh manifest, averaged over the three roots. Covered-target success averages over deployed targets. Service success scores abstained targets as zero. Non-self counts targets served by a donor other than the first-visit specialist of that target, averaged over roots. Bold marks the best value and underline marks the second-best in each column (for capacity and storage, evaluated among full-coverage configurations). Under validation-retention accounting every configuration keeps all eight reference modules at 6.25 MiB.

	Capacity	**Coverage & quality**					**Storage**	
Selector	K	**Cov. (%)**	**Covered (%)**	**Service (%)**	**Gap (pp)**	**Non-self**	**Infer. (MiB)**	**Saved (%)**
PI-latest	8.0	**100.0**	87.42	87.42	0.00	0.0	6.25	0.0
VBC, empirical	**3.0**	**100.0**	91.79	91.79	+4.38	**5.0**	**2.79**	**55.4**
Greedy	**3.0**	**100.0**	91.79	91.79	+4.38	**5.0**	**2.79**	**55.4**
Fixed-1	1.0	50.0	97.00	48.50	-38.92	3.0	1.40	77.6
Fixed-2	2.0	87.5	91.91	80.46	-6.96	**5.0**	2.09	66.5
Fixed-3	3.0	91.7	90.93	83.33	-4.08	4.3	2.79	55.4
Fixed-4	4.0	**100.0**	92.58	**92.58**	**+5.17**	4.0	3.48	44.3
Fixed-5	5.0	**100.0**	91.04	91.04	+3.62	3.0	4.17	33.3
Fixed-6	6.0	**100.0**	89.54	89.54	+2.12	2.0	4.87	22.2
Fixed-7	7.0	**100.0**	88.46	88.46	+1.04	1.0	5.56	11.1
Fixed-8	8.0	**100.0**	87.42	87.42	0.00	0.0	6.25	0.0

Table 22: Per-root detail of the eleven-configuration AmazeVille frontier on the common fresh manifest: retained donors (K), target coverage, and service success (%). Bold marks the best value and underline marks the second-best in each root.

	Capacity K			**Coverage (%)**			**Service success (%)**		
Selector	**Root 1**	**Root 2**	**Root 3**	**Root 1**	**Root 2**	**Root 3**	**Root 1**	**Root 2**	**Root 3**
VBC, empirical	**4**	**2**	**3**	**100.0**	**100.0**	**100.0**	**93.00**	92.88	89.50
Greedy	**4**	**2**	**3**	**100.0**	**100.0**	**100.0**	**93.00**	92.88	89.50
PI-latest	8	8	8	**100.0**	**100.0**	**100.0**	86.75	87.50	88.00
Fixed-1	1	1	1	50.0	50.0	50.0	50.00	49.00	46.50
Fixed-2	2	**2**	2	75.0	**100.0**	87.5	69.00	92.88	79.50
Fixed-3	3	3	**3**	87.5	**100.0**	87.5	78.25	90.38	81.38
Fixed-4	**4**	4	4	**100.0**	**100.0**	**100.0**	**93.00**	**93.37**	**91.38**
Fixed-5	5	5	5	**100.0**	**100.0**	**100.0**	91.25	91.25	90.62
Fixed-6	6	6	6	**100.0**	**100.0**	**100.0**	89.62	89.88	89.12
Fixed-7	7	7	7	**100.0**	**100.0**	**100.0**	88.50	88.75	88.12
Fixed-8	8	8	8	**100.0**	**100.0**	**100.0**	86.75	87.50	88.00

Table 23: Component-level byte breakdown of representative deployment packages, averaged equally over the three roots (in KiB except Total in MiB). High (ret.) is the sum over retained high-level modules. High (ref.) is the sum over reference modules not otherwise retained. Other combines the route map, normalizers, runtime loader, and remaining bookkeeping records (¡12 KiB). Bold marks the smallest inference payload with full coverage, and underline marks the second-smallest.

Package	**Selector**	**Common-low**	**High (ret.)**	**High (ref.)**	**Other**	**Total (MiB)**
Inference	PI-latest	718.0	5671.6	0.0	15.1	6.25
	VBC, empirical	718.0	2126.9	0.0	10.0	**2.79**
	Fixed-4	718.0	2835.8	0.0	11.0	3.48
	Fixed-1	718.0	709.0	0.0	7.9	1.40
Valid.-retent.	PI-latest	718.0	5671.6	0.0	15.1	6.25
	VBC, empirical	718.0	2126.9	3544.8	15.1	6.25
	Fixed-4	718.0	2835.8	2835.8	15.1	6.25
	Fixed-1	718.0	709.0	4962.7	15.0	6.25

J REPRODUCIBILITY AND SCOPE

Environment access. SimpleTown and AmazeVille require the Continual NavBench Godot runtime. PointMaze uses a separate MuJoCo backend. All experiments use deterministic seeds for training, validation, and evaluation. No online reward signal is used to modify policies after offline training completes.

Computational resources. Experiments were conducted on a workstation with a 16-core CPU and a 16 GB GPU. Training uses CPU and GPU computation, while simulator evaluation is primarily CPU-bound. Training updates, validation episodes, and persistent bytes are reported separately because they measure different resource requirements.

Scope and limitations. Our framework assumes stable task identities, resettable historical environments, and bounded identity sets. Specialists are still trained upon arrival, query reductions measure decision-stage testing rather than end-to-end training work, and statistical certificates require paired-reset sampling assumptions. Extending verified reuse to unsegmented task streams and broader control domains represents an important direction for future work.

AI USE STATEMENT

In this work, we used generative AI tools to formulate mathematical claims, provide ingredients for proving mathematical claims, give feedback on research methodology and experiments, implement methods, assist with translation, create or modify scientific figures, and draft and edit parts of this paper. We did not use generative AI tools to develop theoretical models or conceptual frameworks, propose or refine hypotheses, interpret results, suggest experimental parameters, or identify relevant literature. Generating synthetic data sets and similar uses are not applicable to this work. We reviewed all AI-assisted work: the authors manually checked drafted text against the underlying experimental records, verified AI-assisted code through tests, and reconstructed passages that did not match the research design. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.