SEPARATING SEMANTIC APPLICABILITY FROM EXECUTION COMPETENCE FOR TRANSFERABLE RULE-GUIDED HIERARCHICAL REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

A low skill-success rate does not establish that a rule is inapplicable, or that following the skill has no value. In a released rule-guided reinforcement-learning system, we observe skills that fail their symbolic subgoals while earning substantial game reward. Correcting every such failure can remove useful behaviour. We study a recovery interface that keeps semantic action judgements separate from empirical execution evidence. A fixed Jev model receives observed geometry and explicit route knowledge, while pre-evaluation skill success and option returns control intervention outside the model request. The interface preserves empirically productive skills and bounds corrections to the original action space. Across three frozen Kangaroo checkpoints and three fixed starting delays, the resulting system raises mean raw return from 200 to 233.33: three paired conditions improve by 100 and six are unchanged. An earlier recovery configuration averages 133.33, illustrating the cost of correcting productive failures. On three independently trained base policies evaluated every 50k actions up to two million, protected recovery scores above the rule path at all 41 budgets (for example 166.7 against 100 at 0.4M and 0.8M actions) and is never below it in any of 369 conditions, although the across-seed spread is large and the game is one. A deterministic geometry control exactly matches the Jev trajectories in every condition, so the gains support the recovery system rather than a model-specific reasoning advantage. Earlier selector and cross-game diagnostics identify additional limits on where semantic separation can affect behaviour. Budgeted runs of four external baseline families provide a common-evaluator comparison.

1 INTRODUCTION

Rules and learned skills answer different questions. A rule describes when an action or subgoal is appropriate; a skill determines what the agent can accomplish by executing it. Hierarchical reinforcement learning exposes this distinction through temporally extended behaviours (Sutton et al., 1999; Kulkarni et al., 2016). In rule-guided systems such as HRL-ID (Zhang et al., 2025), symbolic matching and measured skill success both influence control. Language-model agents show a related knowing–doing gap: they state the correct rule yet fail to act on it (Schmied et al., 2026). We investigate how to use semantic knowledge when the available skills have different levels of competence, without allowing an execution outcome to stand in for a semantic judgement.

The distinction has an additional practical consequence: *symbolic subgoal success and extrinsic usefulness need not agree*. In the Kangaroo implementation we study, a skill can miss a prescribed location while collecting reward. A success-only recovery gate treats that skill as unsuccessful and permits intervention, even when its behaviour contributes to the evaluation objective. More accurate direction following is therefore not sufficient for a better game return.

Our initial high-level selector study found a separate obstacle. Under the released symbolic abstraction and chain admissibility relation, matched candidate sets rarely create a ranking decision; on the converged configuration in the earlier protocol, the observed rule layer never overrides the neural meta-controller. This is a property of that configuration, not an impossibility result for semantic control. We retain those diagnostics in Appendix A and move the intervention point inside an active

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.

option, where the original skill proposes primitive actions that can be compared with observed route conditions.

The resulting interface separates three operational quantities: semantic suitability under stated observations and rules, empirical subgoal competence, and observed option return. The last two determine whether recovery is requested; they are withheld from the semantic model. Jev chooses among a small, explicit action set when a correction is allowed. A return-protection gate preserves a skill whose mean calibration option return is positive. This is a heuristic for retaining useful behaviour, not a guarantee that intervention will never reduce return.

The completed development comparison supports a bounded claim. With the same checkpoints, calibration, reset conditions and rule-guided reference, protected recovery increases the average return by 33.33 across nine conditions. Every condition is retained. The deterministic geometry control produces the same actions and returns, so the experiment does not identify a unique advantage of Jev reasoning. Three independently trained base policies, evaluated every 50k actions up to two million, place protected recovery above the rule path at all 41 budgets (Section 5.2); the geometry control again matches Jev exactly, and the seed-to-seed spread remains wide. The empirical contribution is the recovery interface and its diagnosis of productive failures, the three-seed learning comparison, and this boundary on model attribution.

Contributions. We provide (i) a measured distinction between symbolic skill success and reward-producing execution; (ii) a recovery interface that keeps semantic inputs separate from competence and return-based intervention decisions; (iii) a complete paired development comparison retaining both the earlier negative result and the positive protected variant, followed by a three-seed learning comparison on the same frozen-recovery protocol; and (iv) source-level and behavioural diagnostics that delimit when high-level rule selection or rule transfer can matter. We do not claim that combining semantic and execution information is itself new, or that these experiments establish general transfer between games. All appendices are supplied as a separate supplementary document; each "Appendix" reference in this paper points to it.

2 Related work

Options and hierarchical controllers structure the action space into skills (Sutton et al., 1999; Dietterich, 2000; Kulkarni et al., 2016; Bacon et al., 2017). Neural-symbolic reinforcement learning adds relational representations and logical reasoning (Ma et al., 2021; Zhang et al., 2025), blends logic and neural policies in Atari (Shindo et al., 2025), mines rules from a trained policy to evaluate and improve it (Tappler et al., 2025), or makes the actor itself a rule ensemble (Yang et al., 2026). Our recovery mechanism operates on a frozen rule-guided hierarchy; it does not replace its learning algorithm or establish a new baseline training result.

Language-guided control already separates task relevance from execution affordances. SayCan combines language-model skill usefulness with an affordance estimate (Ahn et al., 2022); frozen language models have since served as action priors folded into policy learning (Yan et al., 2025), as planners over learned executors (Hu et al., 2025), and as sources of semantic skill labels (Liu et al., 2025) or programmatic options (Moraes et al., 2025). We therefore do not claim novelty for two factors or for grounding language in execution. Here the skill proposes and the frozen model is consulted only at a gated recovery point. When to consult an external judge has been studied as asking a mentor to avoid catastrophic actions (Plaut et al., 2025); since current models abstain unreliably (Kirichenko et al., 2025) even when trained to express calibrated confidence (Bani-Harouni et al., 2026), our trigger uses skill statistics rather than the model's own confidence and restricts the answer to four actions. Our narrower question is how a success-based intervention decision behaves when a failed symbolic subgoal still produces extrinsic reward, and how to preserve that evidence without placing it inside the semantic request. Explicit state-dependent facts constrain the request, while a deterministic geometry control tests whether the model adds behaviour beyond those facts.

Transferable options and successor representations distinguish reusable structure from task-dependent quantities (Konidaris & Barto, 2007; Barreto et al., 2017), and recent hierarchical methods estimate which subgoals are reachable or uncertain before committing (Han et al., 2026; Wang et al., 2025); our competence estimate plays that role for a frozen policy, using calibration rollouts. Here the knowledge interface is reused across checkpoint competence levels and reset conditions.

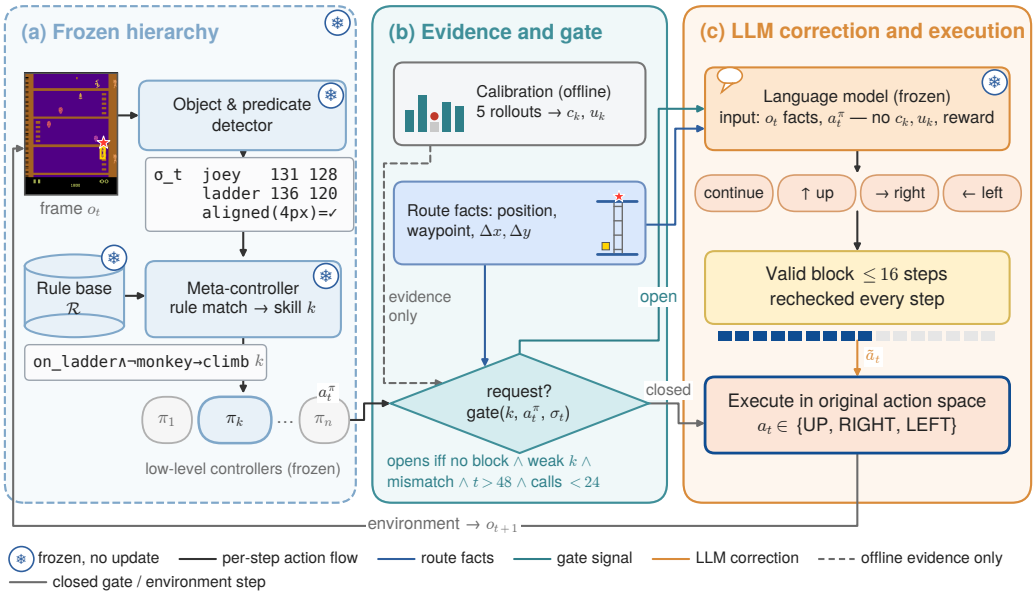

Figure 1: The implemented recovery interface. (a) The frozen rule-guided hierarchy maps the frame o_t to a symbolic state σ_t, matches rules to select the active skill k, and proposes a_t^π; nothing in it is updated. (b) Offline calibration yields c_k and u_k, which reach only the request gate (dashed) and never the request; route facts feed the gate and the model. (c) When the gate opens, the frozen language model chooses a correction that is held as a valid block of at most 16 steps, rechecked every step, and executed in the original action space; a closed gate keeps a_t^π. Solid: per-step flow; dashed: offline evidence; teal: gate signal; orange: correction.

The supplied layout remains Kangaroo-specific. Earlier bidirectional rule-mapping experiments failed to produce executable transferred rules (Appendix D); we retain this limitation rather than treating shared predicate names as evidence of successful transfer.

3 Method: Semantic Recovery with Execution Evidence

3.1 Frozen Hierarchy and Observable Evidence

Let o_t be the current frame and σ_t the symbolic state produced by the base system's object and predicate detector. Let k denote the active low-level skill selected by the existing rule-guided hierarchy, and let a_t^π be its proposed primitive action. We preserve the high-level selector and frozen skill parameters. The available primitive actions are UP, RIGHT and LEFT, with the same ALE action IDs as the base skills. The interface does not introduce a new action capability.

From the current RGB frame, an exact-colour detector estimates the player's position and bounding box. Ambiguous or missing detections remain unknown. Fixed route knowledge specifies platform heights, ladder locations and the active waypoint. The active waypoint can be an intermediate ladder position rather than the final fruit location. The request contains the current position, waypoint, recent observed positions, original action and explicit arithmetic facts, including a four-pixel horizontal alignment tolerance. It does not infer enemy identity, collisions or ground contact from missing evidence, and it does not read simulator RAM.

These facts are estimates about observed geometry, not guarantees of successful action execution. In particular, alignment with a waypoint does not imply safety or positive reward. That distinction is important when interpreting the earlier negative configuration.

3.2 CALIBRATION AND PRODUCTIVE-SKILL PROTECTION

For each actual attempt i at skill k in five pre-evaluation forced-chain calibration episodes, record symbolic success $z_{ki} \in \{0, 1\}$ and raw option return G_{ki}. With n_k attempts, define

$$c_k = \frac{1}{n_k} \sum_i z_{ki}, \qquad u_k = \frac{1}{n_k} \sum_i G_{ki}, \qquad P_k = \mathbf{1}\{n_k > 0 \text{ and } u_k > 0\}. \tag{1}$$

When $n_k = 0$, we retain the base convention $c_k = 0$, but mark u_k as unknown rather than as a measured zero. G_{ki} is the raw return difference during the option in the calibration evaluator, including preserved post-goal actions. It is distinct from the training sampler's shaped reward.

If $P_k = 1$, recovery preserves the original action for that skill. Positive mean calibration return is a natural sign-based protection rule; it is not a fitted test-return threshold. It may protect a weakly productive skill that could be improved. The calibration rolls use the same fixed reset condition as the evaluation cell; their repetition is not independent-environment sampling. Estimates are frozen before evaluation, and no test return updates them.

3.3 WHEN TO REQUEST A SEMANTIC CORRECTION

Let d_t be the deterministic route action: approach horizontally if the waypoint displacement exceeds four pixels, otherwise attempt UP. When no correction block is active, a new request requires known position and goal, together with

$$I_t = (\neg P_k) \wedge (c_k < 0.8) \wedge (t > 48) \wedge (a_t^\pi \neq d_t) \wedge (N_{\text{call}} < 24). \tag{2}$$

Here t counts main-loop actions in the evaluation episode, excluding reset noops. A new option does not restart the initial 48-action exclusion. The original high-level rule matching and competence-based choice remain unchanged; Equation 2 controls the additional primitive-action recovery interface.

The Jev request asks for one of `continue`, `up`, `right` or `left` under the supplied facts and route rules. It excludes c_k, u_k, success outcomes and rewards. The model's confidence is not interpreted as skill competence or used as an additional threshold. A deterministic comparator selects d_t under the same external gate and knowledge geometry. Appendix F gives the exact knowledge statements, choice criteria, a recorded request and response, and validation rules.

A selected direction can be held for at most 16 main-loop actions. At every step, a change in route, waypoint or geometric direction cancels the block; so do an unknown position, a new goal/skill or closure of the competence/protection gate. A mismatch is required for a *new request*, not for continuation of an otherwise valid block. A `continue` response retains each newly proposed original action. Request errors fall back to `continue` and consume the existing call budget.

3.4 WHAT THE SEPARATION DOES AND DOES NOT MEAN

Execution evidence determines when the semantic model is consulted; it does not change the observations supplied as semantic evidence or relabel the model's judgement. Execution outcomes can still affect the final action through the gate. The separation is therefore between evidence channels and decision roles, not a claim that execution evidence has no influence on control.

No theorem of reward improvement follows from this construction. In particular, a correct route action can be unhelpful for the game reward, calibration can miss productive behaviour, and the known layout is incomplete. The experiments test the concrete interface under these limitations.

4 EXPERIMENTAL DESIGN

Checkpoints and environments. We evaluate three frozen Kangaroo checkpoints, denoted Stage 1, Stage 2 and Episode 400. Their different competence profiles provide distinct intervention settings; they are not independent training seeds. Each is evaluated after exactly 1, 10 or 20 initial NOOP actions. The original environment uses ALE seed 0, frame skip 4 and no sticky actions. Policy-side seed 6401 and a separate episode index do not turn these fixed delays into independent

random starts. There are three distinct post-reset RGB hashes, matched across arms within each condition.

We preserve the base evaluator's life-loss/game-over terminal semantics, its score-based and goal-based stopping branches, and its step-limit convention. The parameter is 500 main-loop steps with an inclusive loop check; post-goal actions are recorded in raw return. Thus a score of 300 is neither the game's maximum nor evidence of completing the full game. Reset rewards enter the episode total; reset actions are reported separately from main-loop steps.

Arms and development sequence. Each experiment contains $3 \times 3 \times 4 = 36$ cells, one evaluation episode per cell, with five matched calibration episodes. The four arms are the HRL-ID rule-path reference, coverage-only, deterministic geometry recovery and Jev recovery. Coverage-only fills an otherwise missing candidate with a goal already permitted by the original chain; it does not create a new route. The reference explicitly exercises the rule-guided evaluation path, and is not claimed to reproduce the published neural-only test routine exactly.

The earlier configuration, E9, uses the competence/mismatch gate without productive-skill protection. E10 adds Equation 1; all arms recompute and log the same calibration utility, while recovery arms use it to suppress intervention. We keep all three checkpoints and all three delays. E10 was developed from earlier observed trajectories and reuses these development conditions. Neither the earlier negative result nor this development reuse is hidden by treating E10 as an independent confirmation.

The main comparison is the equal-weight mean of nine paired Jev-minus-reference returns. We also report every pair, the coverage-only and geometry controls, exact action/option-trace agreement and actual model calls. There is no minimum-effect or significance cutoff for counting an observed gain.

Source and execution checks. Every cell records source/checkpoint hashes, pre/post parameter digests, full calibration episodes, reset events, action/reward logs and model request receipts. E10's 18-item per-cell source/framework/checkpoint manifests match the intended pins. One earlier unused postprocessing script was received in the outer source package; all 21 other package entries match, and that script was neither uploaded with the runtime nor invoked. We retain both package manifests and validate results using the corrected postprocessor. This transport exception does not justify claiming complete package-byte identity.

External baseline scope. The external comparison set remains DQN, A2C, NSRL and HDQN, following the original HRL-ID study. These are not replaced by the internal geometry or coverage controls. DQN and A2C use a pinned SLM-Lab implementation through the original ALE interface. NSRL (Ma et al., 2021) is a Kangaroo reimplementation with task, head and reward adaptations disclosed in Appendix E. HDQN uses the matched five-goal domain. All four candidates have completed the three-checkpoint protocol (Appendix G). At increasing action budgets, DQN returns 0/200/200 at every start; A2C returns zero throughout, and NSRL returns 0/100/0. HDQN returns 0/0/0, then 100/0/0 at both later checkpoints (NOOP 1/10/20). Published curves with different metrics or budgets are not substituted as reruns. Appendix E specifies the implementations, training budgets and common evaluation. That evaluator removes the inherited option-boundary score cutoff for comparability with flat policies; actual ALE replay preserves all 36 recorded E10 episode returns under this change, without establishing equivalence for arbitrary trajectories.

5 Results

5.1 Positive mean gain without observed regressions

Across the nine paired E10 conditions, the rule-path reference and coverage-only arm both average 200 raw reward. Protected Jev and protected geometry average 233.33, a difference of 33.33 (16.67% relative to the reference). Each Stage 2 condition improves from 100 to 200; all six Stage 1 and Episode 400 conditions are unchanged. There is no observed regression in these fixed development conditions. This establishes the measured mean gain without requiring a significance claim from deterministic repeats.



Figure 2: Recovery under two evidence regimes. (a) Development conditions: three related checkpoints, each averaged over three fixed starts. Unprotected recovery (E9) corrects reward-producing failures and loses return at Episode 400 and Stage 1; protected recovery (E10) keeps them and retains the Stage 2 gain. Geometry matches Jev exactly in both experiments. (b) Three independent training seeds: mean of the per-seed means and ± 1 sample SD across seeds at every saved 50k-action snapshot; large markers are the pre-registered budgets, small markers the post-hoc dense budgets. Protected geometry (dashed) coincides with protected Jev wherever both were run; the Jev arm was run at 22 of the 41 budgets. One seed-11 Jev start at 2.0M is a single authorized rerun after an API failure (Appendix I).

Table 1: Every E10 paired condition. Coverage-only matches the rule-path reference. Geometry and Jev both include return protection. These are fixed-delay development conditions, not independent seeds.

Checkpoint	NOOP	Reference	Coverage	Geometry	Jev	Difference
Stage 1	1	300	300	300	300	+0
Stage 2	1	100	100	200	200	+100
Episode 400	1	200	200	200	200	+0
Stage 1	10	300	300	300	300	+0
Stage 2	10	100	100	200	200	+100
Episode 400	10	200	200	200	200	+0
Stage 1	20	300	300	300	300	+0
Stage 2	20	100	100	200	200	+100
Episode 400	20	200	200	200	200	+0
Mean	—	200	200	233.33	233.33	+33.33

The 18 reference and coverage-only cells reproduce E9's action/reward projections, providing a direct check that the improvement is not due to a changed reference rollout. Weights and rules are unchanged, and full calibration records match across the four arms of each condition.

5.2 Three independent training seeds

The development comparison above uses related checkpoints from two training runs. To measure recovery as the base policy learns, we train three independent HRL-ID Kangaroo policies with the released eight-worker configuration (seeds 11, 29 and 47; common ALE seed 0), save the first pre-update snapshot crossing each of eight pre-registered budgets (0, 50k, 100k, 200k, 400k, 800k, 1.2M actions, and the final post-update weights at 2.0M), and evaluate all four arms on each snapshot under the same three fixed starts and the E10 rules. After that grid was complete we additionally evaluated the three arms that make no model request at every saved 50k snapshot (41 budgets in all) and the Jev arm at every 100k budget (22 budgets); the addition is post hoc and is marked as such in Figure 2(b) and Appendix I. Every point in Figure 2(b) is the mean of the three per-seed means; the band is one sample standard deviation across seeds (ddof=1), the multi-seed reporting practice of

Table 2: Three independent training seeds at the common action grid: per-seed mean over the three fixed starts (seed 11 / 29 / 47) and across-seed mean±sample SD (ddof=1). Geometry and Jev act identically in all 72 conditions. The seed-11 Jev cell at 2.0M (start NOOP 20) is a single authorized rerun after an HTTP 529 failure (Appendix I).

Actions (M)	HRL-ID		Chain only		Geometry		Jev	
	per seed	mean	per seed	mean	per seed	mean	per seed	mean
0	0 / 0 / 0	0.0±0.0	0 / 0 / 200	66.7±115.5	67 / 67 / 200	111.1±77.0	67 / 67 / 200	111.1±77.0
0.05	0 / 0 / 0	0.0±0.0	0 / 0 / 0	0.0±0.0	0 / 133 / 100	77.8±69.4	0 / 133 / 100	77.8±69.4
0.10	0 / 0 / 0	0.0±0.0	200 / 0 / 0	66.7±115.5	200 / 67 / 200	155.6±77.0	200 / 67 / 200	155.6±77.0
0.20	100 / 0 / 0	33.3±57.7	100 / 100 / 100	100.0±0.0	200 / 100 / 100	133.3±57.7	200 / 100 / 100	133.3±57.7
0.40	100 / 100 / 100	100.0±0.0	100 / 100 / 100	100.0±0.0	200 / 100 / 200	166.7±57.7	200 / 100 / 200	166.7±57.7
0.80	100 / 100 / 100	100.0±0.0	100 / 100 / 100	100.0±0.0	200 / 100 / 200	166.7±57.7	200 / 100 / 200	166.7±57.7
1.20	0 / 100 / 100	66.7±57.7	100 / 100 / 100	100.0±0.0	200 / 100 / 200	166.7±57.7	200 / 100 / 200	166.7±57.7
2.00	0 / 100 / 100	66.7±57.7	100 / 100 / 100	100.0±0.0	200 / 100 / 200	166.7±57.7	200 / 100 / 200	166.7±57.7

Table 3: Mean raw return at explicitly matched nominal training budgets (three fixed NOOP starts). Checkpoint budgets are reconstructed from logs; Episode 400 is from E1, whereas Stage 1 and Stage 2 are from E1_fresh. These are neither independent seeds nor a single recovery training trajectory. Actual baseline counts and all scores are in Appendices E and G; recovery calibration and API costs are additional.

Method Nominal actions	Episode 400 78,566	Stage 1 1,105,035	Stage 2 1,126,718
HRL-ID rule path	200	300	100
Coverage only	200	300	100
Protected geometry	200	300	200
Protected Jev	200	300	200
DQN (candidate)	0	200	200
A2C (candidate)	0	0	0
NSRL (reimplementation)	0	100	0
HDQN (five-goal adaptation)	0	33.33	33.33

Atari studies (Machado et al., 2018; Kooi et al., 2025). Starting delays are never pooled as repeats, and a grid point with fewer than three seeds is withheld rather than filled. Each training run passes an independent raw-ledger check (eight per-worker ledgers, 194 updated parameter tensors, 2.7–3.1 million optimizer steps); Appendix I lists the protocol, the actual action counts and the request accounting.

Protected recovery exceeds the rule path at every one of the 41 budgets. The rule path stays at zero mean return through 100k actions, rises to 166.7 at 300k, holds 100 from 350k to 1.15M and, from 1.2M on, alternates between 100 and 66.7 (33.3 at 1.9M) as one seed (seed 11 at every such budget except 1.8M, where it is seed 29; both at 1.9M) loses its return at all three starts; the protected arms peak at 266.7 at 250k and 200 at 300k and 600k, and otherwise hold 166.7 from 350k on. Over the 369 seed–budget–start conditions with a geometry arm, protected geometry scores above the rule path in 259 and ties in 110; over the 198 conditions with a Jev arm, Jev is above in 139 and ties in 59; no condition is worse for either arm. The advantage is still a small-sample estimate: at most budgets one seed sits 100 points from the other two, so the seed SD is 58–77 and the bands overlap.

Chain completion alone reaches 100 by 200k and, apart from the same peaks at 250k–300k and 600k, stays at 100 thereafter; from 350k on it equals the rule path wherever the rule path is at 100 and exceeds it where the rule path drops. Restoring a missing candidate with a goal the chain already permits therefore explains the 100-point plateau, but not the protected arms' additional 66.7 (33.3 at 600k). Geometry and Jev again produce identical action and option traces in all 198 conditions where both were run, and 15 of the 198 Jev episodes issue no request because the protection gate keeps every proposed action. The curve therefore credits the recovery interface and its gate, not model reasoning beyond the supplied route facts, and 2.0M actions is the declared initial window, not convergence.

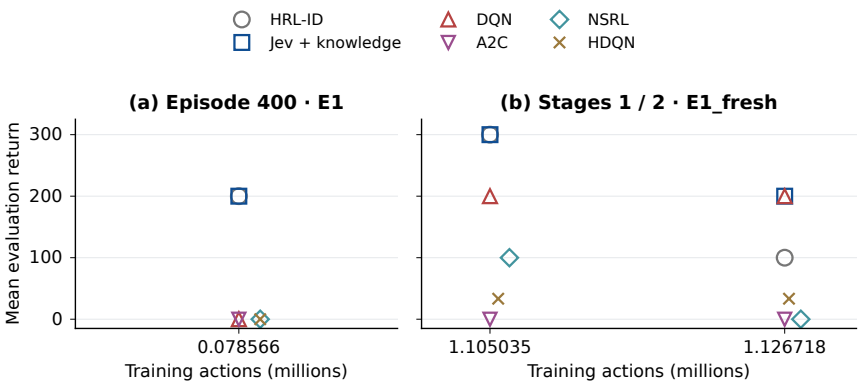

Figure 3: Recorded common-evaluator returns versus training budget. The two panels separate the E1 and E1_fresh reference runs. Baselines use their actual recorded counts; HRL-ID budgets are reconstructed from logs. Markers denote only measured points, with no interpolation or seed confidence bands. Coverage-only equals the reference; protected geometry equals protected Jev at every point and is omitted to avoid duplicate markers.

5.3 External baseline comparison

Table 3 places the recorded recovery arms alongside the completed external baseline candidates. At the Stage 1 budget, DQN reaches 200 while both protected recovery arms reach 300; at the Stage 2 budget, all three reach 200. The Episode 400 reference comes from a separate training run. The table reports each checkpoint separately, including the intermediate NSRL result and HDQN's sensitivity to starting delay; it does not select a best checkpoint or establish independent-seed superiority. Figure 3 plots these sparse evaluation points against their training budgets without interpolation: Episode 400 and the two stage checkpoints belong to different runs, and recovery itself freezes policy weights. Appendix H reports the available dense baseline training diagnostics and the distinction from the learning-efficiency comparison in the original HRL-ID study. Section 5.2 presents the three-seed learning comparison for the recovery arms; the matching three-seed baseline runs on the same grid (Appendix I, Table S12) leave A2C at zero and HDQN and NSRL at or below 55.6, while DQN, with two seeds, reaches 200 by 800k and 300 at 1.2M, above every recovery arm there; recovery adds to a frozen hierarchical policy and does not claim to beat a flat learner.

5.4 Why protecting productive failures matters here

In E9, Jev and geometry both average 133.33 against the same 200 reference. Stage 1 loses 100 and Episode 400 loses 200 at every starting delay, while Stage 2 improves by 100. The calibration option-return vectors in E10 are

$$\begin{aligned} \text{Stage 1}: &\quad [0, 100, 200, \text{unknown}], \\ \text{Stage 2}: &\quad [0, 100, 0, \text{unknown}], \\ \text{Episode 400}: &\quad [200, \text{unknown}, \text{unknown}, \text{unknown}]. \end{aligned}$$

The Stage 1 skill that misses its location target still produces 200 reward, and the first Episode 400 skill also earns 200 without symbolic success. The new gate preserves those actions. Stage 2 retains a zero-return, low-success skill on which correction remains possible. These observations explain the gate's operation on the measured trajectories; they are not a theorem that reward protection will improve other policies.

Across all 36 E10 cells, 5,168 protected main-loop steps retain the original action and make no model call; 1,292 of these steps are in the Jev arm. Complete per-step checks verify both properties. Unattempted skills remain explicitly unknown rather than being reclassified as measured zero-return skills.

5.5 Attribution and model use

E10 makes 27 successful official Jev calls with 55,416 input tokens and no recorded error, compared with E9's 86 calls and 181,413 input tokens. No policy training occurs in either experiment. All nine geometry/Jev pairs have identical action/reward sequences and option traces. The observed improvement therefore supports the shared recovery system and the new intervention gate, but does not demonstrate additional behavioural value from Jev over the deterministic control.

This limitation does not invalidate the paired system-level return difference; it determines which component can be credited. These experiments do not establish the model's necessity or its superiority over geometry.

5.6 Relation to the earlier selector and transfer diagnostics

The earlier local-NLI selector study observes 800 high-level decisions but only four distinct contexts in its deterministic converged configuration; singleton admissible candidate sets make the intended ranking distinction inactive there, which motivated moving recovery to a decision point where action proposals differ.

The earlier bidirectional rule transfer tests also remain negative: shared predicate names did not yield executable mapped rules across the two object/admissibility interfaces, and E10 does not repair those cross-game failures. Appendix A retains the original component results.

6 Limitations and conclusion

The evidence comes from one game. The development comparison uses related checkpoints, deterministic reset delays and repeated calibration rollouts, and the protection rule was motivated by earlier outcomes. The three-seed comparison has only three runs, a two-million-action window without convergence, a seed SD of 58–77 at most budgets, and one Jev cell rerun once after an API failure. The external baselines are reported on the same three-seed grid (Appendix I); DQN, with two seeds, exceeds every recovery arm from 800k, and no method-superiority claim is made. Cross-game transfer remains unproven, and the layout knowledge is domain-specific.

The return-protection rule can also be wrong: any positive calibration mean protects a skill, including a weak skill that could benefit from intervention. Conversely, a nonproductive calibration sample does not prove that the original skill is unhelpful. Unknown pixel states suppress correction; unobserved hazards can make a geometrically justified action unsafe. A sixteen-action block remains a control commitment, despite per-step route checks.

Within these boundaries, separating semantic action evidence from execution and reward-based intervention decisions preserves useful behaviour while recovering additional return: a 33.33 mean improvement under the development conditions, and protected arms above the rule path at every budget of the three-seed comparison with no condition worse. Exact geometry-model agreement in all completed conditions limits model-specific attribution, and the earlier negative results show why neither a better subgoal-success rate nor a more explicit semantic request alone is evidence of better reinforcement-learning performance.

AI USE STATEMENT

Generative AI tools (Codex with GPT-6-Sol) assisted with code and data analysis scripts.

ETHICS STATEMENT

The work uses an Atari research environment, without human-subject interaction. Earlier selector experiments used a local open-weight NLI model; the Jev recovery experiments use an external official API receiving game observations and explicit rule requests. Authentication remains in the local gateway and is not sent to the game host. No credential material is included in experimental reports.

REPRODUCIBILITY STATEMENT

The recovery runs freeze policy weights, rules, route knowledge and model identifier `jev-1.13.0`. Logs retain original proposals, executed actions, raw rewards, calibration attempts, unknown values, source/checkpoint hashes and API receipts. A verifier independently recomputes utility, paired returns, call accounting and parameter invariance. The supplied source supports replaying the recorded checks; future live API outputs are not guaranteed byte-identical. Appendices document the earlier selector and the budgeted baseline training and evaluation. The appendices are provided as a separate supplementary PDF.

REFERENCES

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Daniel Ho, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Eric Jang, Rosario Jauregui Ruano, Kyle Jeffrey, Sally Jesmonth, Nikhil J Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Kuang-Huei Lee, Sergey Levine, et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022.

Pierre-Luc Bacon, Jean Harb, and Doina Precup. The option-critic architecture. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017. doi: 10.1609/aaai.v31i1.10916.

David Bani-Harouni, Chantal Pellegrini, Paul Stangel, Ege Özsoy, Kamilia Zaripova, Nassir Navab, and Matthias Keicher. Rewarding doubt: A reinforcement learning approach to calibrated confidence expression of large language models. In *International Conference on Learning Representations*, 2026.

André Barreto, Will Dabney, Rémi Munos, Jonathan J. Hunt, Tom Schaul, Hado van Hasselt, and David Silver. Successor features for transfer in reinforcement learning. In *Advances in Neural Information Processing Systems*, 2017.

Thomas G. Dietterich. Hierarchical reinforcement learning with the MAXQ value function decomposition. *Journal of Artificial Intelligence Research*, 13:227–303, 2000. doi: 10.1613/jair.639.

Seungyul Han, Jaebak Hwang, Sanghyeon Lee, and Jeongmo Kim. Strict subgoal execution: Reliable long-horizon planning in hierarchical reinforcement learning. In *International Conference on Learning Representations*, 2026.

Zican Hu, Wei Liu, Xiaoye Qu, Xiangyu Yue, Chunlin Chen, Zhi Wang, and Yu Cheng. Divide and conquer: Grounding LLMs as efficient decision-making agents via offline hierarchical reinforcement learning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 24570–24590, 2025.

Polina Kirichenko, Mark Ibrahim, Kamalika Chaudhuri, and Samuel J. Bell. AbstentionBench: Reasoning LLMs fail on unanswerable questions. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

George Konidaris and Andrew Barto. Building portable options: Skill transfer in reinforcement learning. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, 2007.

Jacob Eeuwe Kooi, Zhao Yang, and Vincent François-Lavet. Hadamax encoding: Elevating performance in model-free Atari. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Tejas D. Kulkarni, Karthik R. Narasimhan, Ardavan Saeedi, and Joshua B. Tenenbaum. Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation. In *Advances in Neural Information Processing Systems*, 2016.

Tian-Shuo Liu, Xu-Hui Liu, Ruifeng Chen, Lixuan Jin, Pengyuan Wang, Zhilong Zhang, and Yang Yu. Semantic temporal abstraction via vision-language model guidance for efficient reinforcement learning. In *International Conference on Learning Representations*, 2025.

Zhihao Ma, Yuzheng Zhuang, Paul Weng, Hankz Hankui Zhuo, Dong Li, Wulong Liu, and Jianye Hao. Learning symbolic rules for interpretable deep reinforcement learning. *arXiv preprint arXiv:2103.08228*, 2021.

Marlos C. Machado, Marc G. Bellemare, Erik Talvitie, Joel Veness, Matthew Hausknecht, and Michael Bowling. Revisiting the arcade learning environment: Evaluation protocols and open problems for general agents. *Journal of Artificial Intelligence Research*, 61:523–562, 2018. doi: 10.1613/jair.5699.

Rubens O. Moraes, Quazi Asif Sadmine, Hendrik Baier, and Levi H. S. Lelis. InnateCoder: Learning programmatic options with foundation models. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, 2025.

Benjamin Plaut, Hanlin Zhu, and Stuart Russell. Avoiding catastrophe in online learning by asking for help. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 49537–49568, 2025.

Thomas Schmied, Jörg Bornschein, Jordi Grau-Moya, Markus Wulfmeier, and Razvan Pascanu. LLMs are greedy agents: Effects of RL fine-tuning on decision-making abilities. In *International Conference on Learning Representations*, 2026.

Hikaru Shindo, Quentin Delfosse, Devendra Singh Dhami, and Kristian Kersting. BlendRL: A framework for merging symbolic and neural policy learning. In *International Conference on Learning Representations*, 2025.

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

Martin Tappler, Ignacio D. Lopez-Miguel, Sebastian Tschiatschek, and Ezio Bartocci. Rule-guided reinforcement learning policy evaluation and improvement. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, 2025.

Vivienne Huiling Wang, Tinghuai Wang, and Joni Pajarinen. Hierarchical reinforcement learning with uncertainty-guided diffusional subgoals. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 64750–64772, 2025.

Xue Yan, Yan Song, Xidong Feng, Mengyue Yang, Haifeng Zhang, Haitham Bou Ammar, and Jun Wang. Efficient reinforcement learning with large language model priors. In *International Conference on Learning Representations*, 2025.

Yue Yang, Fan Yang, Yu Bai, and Hao Wang. Neural+symbolic approaches for interpretable actor-critic reinforcement learning. In *International Conference on Learning Representations*, 2026.

Haodi Zhang, Xiangyu Zeng, Junyang Chen, Yuanfeng Song, Rui Mao, and Fangzhen Lin. Deduction with induction: Combining knowledge discovery and reasoning for interpretable deep reinforcement learning. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pp. 6984–6992, 2025. doi: 10.24963/ijcai.2025/777.