Learning to Preserve: Q-Learning in the Two-Dimensional Causal Core Framework

Reward Attribution, Dual Crisis Prevention, and the Emergence of Conservative Self-Preservation Strategies

ALI Simulation Series, Stage 6

Wolfgang Stegemann

Abstract

We combine the Q-learning Ego of Stage 4 with the two-dimensional causal core (CC) of Stage 5, creating a learning agent that must balance energy and structural integrity through an internally derived reward signal without any external specification. The training process reveals a reward attribution failure: an implementation error that assigned the collection reward regardless of collection success produced a stable but fully dysfunctional strategy in which the agent repeatedly selected collect without ever picking up a packet. After correction, the agent converges to a viable strategy over 4000 episodes, achieving 100% survival across 20 test seeds — compared to 0% for the hardcoded Stage 5 Ego over the same episode length. However, mean delivery performance drops to 0.35 per run compared to approximately 4.0 for the hardcoded Ego. The central finding is a strategic inversion: instead of replicating the explicit priority ordering of the hardcoded Ego (integrity before energy under dual crisis), the Q-learning Ego discovers a preventive strategy that avoids dual crisis entirely through proactive rest allocation, using 54.8 rest actions per run compared to 13 in Stage 5. Zero dual crisis events occur across all 20 test seeds. This demonstrates that the same CC reward signal can generate qualitatively different behavioral equilibria, and that the learned equilibrium may achieve superior self-preservation at the cost of task performance. The reward attribution error is documented as a methodological finding with implications for Q-learning system design.

1. Introduction

The ALI simulation series has progressively extended the causal core (CC) framework across five stages. Stage 4 [3] demonstrated that a Q-learning Ego evaluated against a one-dimensional CC metric can converge to a viable strategy combining self-preservation and task execution. Stage 5 [4] introduced a two-dimensional CC integrity vector and showed that the hardcoded Ego discovers preventive norm compliance through proactive integrity management. Stage 6 asks whether these two results can be combined: can a Q-learning Ego, evaluated against a two-dimensional CC reward signal, learn to balance energy and structural integrity without any explicit priority ordering?

This combination is theoretically significant for two reasons. First, it tests whether the CC metric scales as an internal reward source when the metric itself is multi-dimensional. A two-dimensional CC produces a richer reward landscape than the scalar case: the same action can produce positive reward on one dimension and negative reward on the other, creating a training signal with internal tension. Second, it tests whether the learned strategy approximates, exceeds, or diverges from the hardcoded strategy of Stage 5, extending the comparison first made in Stage 4.

The results are unexpected in both respects. An implementation error in the reward attribution logic reveals a critical sensitivity of Q-learning systems to reward signal accuracy. After

correction, the learned strategy diverges substantially from the hardcoded one, achieving superior self-preservation through a preventive rest-heavy approach that the hardcoded Ego never discovered, but at a significant cost to task performance.

2. Architecture

2.1 State Space Extension

The state space of Stage 4 (six dimensions) is extended with two new dimensions: integrity level (four discrete levels, parallel to the existing energy level dimension) and dual crisis flag (binary: 1 when both energy and integrity are simultaneously below their respective warning thresholds). The resulting eight-dimensional state space has a theoretical maximum of approximately $4 \times 4 \times 2 \times 9 \times 8 \times 5 \times 4 \times 2 = 46{,}080$ states, of which 1,698 were visited during the 4000-episode training run, reflecting the sparse coverage characteristic of the ALI's behaviorally constrained operating region.

2.2 Two-Dimensional Reward Function

The reward function is extended from Stage 4 to incorporate both CC dimensions. The survival modulator, which previously rewarded stable energy, now reflects the joint state of both dimensions:

When both dimensions are critical (dual crisis): survival modulator = −2.0, double the single-dimension penalty. When one dimension is critical: survival modulator = −0.8. When both are stable: survival modulator = +0.15.

The rest action receives a dedicated reward: +1.5 × integrity delta when integrity actually increases, −0.2 when no recovery occurs. This makes rest conditionally rewarding: it is positively valued when it produces genuine integrity recovery, and penalised when it is performed without need or effect.

2.3 The Reward Attribution Error

During initial training, the collect action was assigned the collection reward (+1.2) unconditionally, regardless of whether a packet was actually present at the agent's position. The implementation assigned *collected = True* after calling *ich.collect()*, without checking the return value of that call. As a result, the agent received collection reward for every collect action, even when the action failed.

The behavioral consequence was immediate and stable: the agent converged to a strategy of continuously selecting collect regardless of position, receiving +1.2 per step, and never successfully picking up a packet. Over 3000 training episodes, the reward signal grew continuously (from −1.85 to +23.00 per episode mean) while deliveries remained at zero. The reward function had learned to reward a dysfunctional equilibrium.

This error illustrates a general principle for Q-learning in multi-action systems: reward must be conditioned on actual state transitions, not on action selection. An action that fails to produce its intended state change must either receive a neutral reward or a penalty, not the reward of success. After correction — returning *collected = False* when no packet is present, with a penalty of −0.2 for unsuccessful collection — the learning trajectory changed qualitatively within 500 episodes.

3. Results

3.1 Learning Trajectory After Correction

Over 4000 training episodes, the corrected agent passes through three phases. In episodes 1 to 500, survival rate is 18.6% and deliveries are zero, as in Stage 4. In episodes 501 to 2000, survival rate rises sharply to 74–90% as the agent learns stable self-preservation through a rest-heavy strategy. In episodes 2001 to 4000, survival rate stabilises at 87–95% and deliveries emerge, reaching a mean of 0.83 per episode in the final 500 episodes.

The Q-table grows from 0 to 1,698 states, confirming that the agent explores a consistent behavioral region. Rest actions dominate the action distribution: by convergence, the agent performs approximately 54 rest actions per 200-step episode, compared to approximately 13 in the hardcoded Stage 5 agent.

3.2 Strategic Inversion: Prevention vs. Recovery

The most significant finding of Stage 6 is a strategic inversion relative to Stage 5. The hardcoded Stage 5 Ego implements a reactive dual-crisis management strategy: when both dimensions become critical simultaneously, integrity takes priority. This reactive strategy produces dual crisis events in 93% of runs. The Q-learning Ego implements a preventive strategy: it rests so frequently that integrity never reaches the warning threshold, eliminating dual crisis events entirely.

Across all 20 test seeds at convergence ($\varepsilon = 0$), zero dual crisis events occur. The agent achieves 100% survival, compared to 0% for the hardcoded Stage 5 agent over 200 steps. This is not because the hardcoded agent is poorly designed: it achieves significantly higher task performance (approximately 4 deliveries per run vs. 0.35). The learned agent has found a different point on the survival-task tradeoff curve, one that maximises self-preservation at the expense of task execution.

3.3 Comparison with Stage 5 Hardcoded Ego

The comparison reveals complementary strengths. The hardcoded Ego is superior at task execution (approximately 11× more deliveries) but fails to survive. The learned Ego survives reliably but delivers rarely. The fundamental difference is in the allocation of rest actions: the hardcoded Ego rests reactively when integrity is low, the learned Ego rests proactively to prevent integrity from becoming low.

This difference reflects the distinct optimization targets. The hardcoded Ego optimises a fixed priority ordering designed by the system architect. The learned Ego optimises the CC reward signal, which weights survival positively in every step. Given sufficiently many episodes, Q-learning discovers that preventing crisis is more consistently rewarded than recovering from it — a result that emerges from the reward structure, not from explicit design.

3.4 N7 Activation

As in Stage 5, norm N7 (movement blocked when integrity is critical) is never activated in the test runs. This is no longer merely a property of the hardcoded priority ordering: the learned Ego, without any explicit instruction, has independently discovered a strategy that keeps integrity above the N7 threshold at all times. Preventive norm compliance is robust to the learning process: it emerges from Q-learning evaluated against the CC metric as reliably as it was encoded in the hardcoded priority ordering.

4. Discussion

4.1 Reward Attribution as a Design Problem

The reward attribution error documented in Stage 6 is not a minor implementation mistake. It reveals a structural sensitivity of Q-learning systems to the precision of reward conditioning. In systems with multiple possible outcomes for the same action (collect can succeed or fail depending on environmental state), reward must track actual outcome, not action selection. The same principle applies to any action whose success depends on environmental contingency: deliver (requires carrying, requires being at station, requires sufficient energy), eat (requires packet at current position), rest (produces repair only if integrity is below maximum).

The error produced a stable, high-reward trajectory that was completely dysfunctional. The agent learned a locally optimal policy under the wrong reward function. This has a broader implication for ALI design: the CC metric as internal reward source provides the correct optimization target, but its implementation in the reward function must reflect actual state transitions with precision. Incorrect reward attribution can produce stable dysfunction that is invisible from the reward curve alone.

4.2 Multiple Equilibria Under the Same CC Metric

Stage 6 demonstrates that the CC metric does not uniquely determine behavioral strategy. The hardcoded Stage 5 Ego and the learned Stage 6 Ego are both evaluated against the same CC integrity vector, but converge to substantially different equilibria. The hardcoded Ego finds a task-execution equilibrium with reactive integrity management. The learned Ego finds a survival equilibrium with preventive integrity management.

This multiplicity of equilibria has direct implications for ALI deployment. A system designer who specifies only the CC metric and allows the Ego to learn from it cannot predict which equilibrium the learning process will converge to. Additional constraints — either through explicit Ego priority ordering or through reward shaping — are necessary to select among equilibria. The Super-Ego's normative architecture provides one mechanism for this selection: N7 implicitly favours strategies that maintain integrity above threshold, which is compatible with both the reactive and preventive equilibria. But it does not select between them.

4.3 Implications for Stage 7

Stage 7 will move from simulation to a domain-specific prototype. The Stage 6 findings have three implications for this transition. First, the reward attribution principle must be applied with particular care in real systems, where action outcomes depend on complex environmental interactions that are harder to enumerate than in a grid world. Second, the multiplicity of equilibria suggests that real-world ALI deployment should specify not just the CC metric but a preferred equilibrium region — for instance, a minimum delivery rate requirement expressed as a Super-Ego norm rather than a reward signal. Third, the 100% survival rate of the learned Stage 6 Ego, if robust to more complex environments, represents a deployment-relevant property that the hardcoded Ego lacks.

5. Conclusion

Stage 6 establishes four findings. A reward attribution error — assigning collection reward independent of collection success — produces stable, high-reward, fully dysfunctional behavior, illustrating that Q-learning in multi-action systems requires precise outcome-conditioned reward assignment. After correction, the Q-learning Ego converges to 100% survival across 20 test seeds through a preventive rest-heavy strategy that eliminates dual crisis events entirely. This strategy inverts the reactive dual-crisis management of the hardcoded Stage 5 Ego, achieving superior self-preservation at the cost of task performance. Norm N7 is never activated, confirming that preventive norm compliance is robust to the learning process: it emerges from CC-evaluated Q-learning as reliably as from explicit priority ordering.

These findings establish that the CC metric generates multiple behavioral equilibria and that the learning process selects among them based on the reward landscape rather than designer intent. Explicit constraints — whether through Ego priority ordering or Super-Ego norm specification — remain necessary to select the deployment-relevant equilibrium.

References

[1] Stegemann, W. (2026). From the Myth of AGI to the Architecture of a Controllable ALI. Zenodo. https://doi.org/10.5281/zenodo.20378675

[2] Stegemann, W. (2026). Implementing the Causal Core: A Simulation of Artificial Local Intelligence. Zenodo. https://doi.org/10.5281/zenodo.20379008

[3] Stegemann, W. (2026). Learning Without External Reward: Q-Learning in the ALI Ego Instance. Zenodo. https://doi.org/10.5281/zenodo.20402389

[4] Stegemann, W. (2026). The Multi-Dimensional Causal Core. Zenodo. https://doi.org/10.5281/zenodo.20407218

[5] Stegemann, W. (2026). Two-Agent Dynamics in ALI. Zenodo. https://doi.org/10.5281/zenodo.20396009

[6] Stegemann, W. (2025). The Emergence of Causal Cores. Zenodo. https://doi.org/10.5281/zenodo.15558264

[7] Stegemann, W. (2025). The Causal Core as a Structural Principle of Free Will. Zenodo. https://doi.org/10.5281/zenodo.17057739

[8] Stegemann, W. (2026). The Causal Core as Interpreter. Zenodo. https://doi.org/10.5281/zenodo.18340973

[9] Simulation v6 on GitHub: https://github.com/drwolfgangstegemann-sudo/ALI-Simulation-v6-Q-Learning-Zweidimensionaler-Kausaler-Kern