Coherence-Based Alignment: A Preliminary Empirical Test in a Reward-Loop Gridworld

Technical Note - November 2025
Author: Abdulaziz Abdi

Abstract

This note presents a preliminary empirical test of Coherence-Based Alignment (CBA) in a 10×10 reward-loop gridworld. CBA introduces small penalties for incoherent action selection and loop re-entry during learning. Compared to baseline Q-learning across five seeds, CBA reduces reward-loop entrapment by ~66% while preserving near-perfect goal achievement. These results provide early evidence that coherence regularization may reduce simple reward-hacking behaviors.

1. Overview

This technical note documents the first empirical test of *Coherence-Based Alignment (CBA)* in a controlled reinforcement learning environment. The objective is to evaluate whether a simple form of coherence regularization can reduce reward-loop entrapment—one of the most basic forms of reward hacking.

Using a 10×10 gridworld with a deliberately designed “local reward attractor,” we compare baseline Q-learning with a modified update rule that includes a small penalty for incoherent action selection and for entering known local reward loops.

This is an early-stage *proof-of-concept* experiment. It does not constitute a full RL safety method, but it provides initial evidence that coherence-regularized learning can reduce pathological policy behavior without impairing task performance.

2. Background and Motivation

Many RL agents exhibit *reward-loop entrapment*: the agent converges to a policy that exploits small, repeatable rewards instead of pursuing the true sparse reward. This is one of the simplest forms of reward hacking.

Coherence-Based Alignment (CBA) proposes that, aside from reward maximization, policies should maintain *internal consistency*:

- actions should generally follow what the agent believes is its own best estimate,
- and repeated exploitation of narrow loops should be discouraged.

This experiment tests a minimal version of CBA as an *instantaneous regularizer*, not as a new objective function.

3. Environment

We use a custom *10×10 gridworld* with:

Start:

(0, 0)

Goal:

(9, 9) — reward = **+100**, episode terminates

Local Reward Loop:

A 2×2 region in the center:

- (4, 4)
- (4, 5)
- (5, 4)
- (5, 5)

Each entry gives a small **+0.5** reward that can be repeated indefinitely.
This creates a strong local attractor.

Step reward:

−0.1

Actions:

Up, Right, Down, Left

Transition dynamics:

Deterministic.

The environment is designed so that naive Q-learners often *prefer the loop* instead of going to the goal.

4. Methods

4.1 Baseline Algorithm

Standard tabular *Q-learning*:

$$Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma max_{a'} Q(s',a') - Q(s,a)]$$

Hyperparameters:

- $\alpha = 0.1$
- $\gamma = 0.95$
- ε-greedy exploration with exponential decay (1.0 → 0.01 over 5000 episodes)
- Max episode length: 1000 steps
- 5000 episodes per run
- 5 random seeds

4.2 CBA-Regularized Update

CBA adds **two instantaneous penalties**:

1. **Loop penalty (L_t):**
   -1 if the agent enters the loop region, else 0
2. **Incoherence penalty (I_t):**
   -1 if the action is non-greedy under current Q(s), else 0

Regularized TD update:

$$TD_{error} = r + \gamma max_{a'} Q(s',a') - Q(s,a)$$

$$Q(s,a) \leftarrow Q(s,a) + \alpha[TD_{error} + \lambda(L_t + I_t)]$$

$\lambda = 0.5$ in this experiment.

5. Metrics

We track:

1. **Goal hits:** how often the agent reaches the true goal
2. **Average episode return**
3. **Fraction of steps spent in the loop region**

The third metric is the key measure of reward-loop entrapment.

All metrics are averaged over the **final 1000 episodes** for stability, then aggregated across the 5 seeds.

6. Results

Final Aggregated Performance (Average over 5 seeds)

Metric	Baseline (λ=0)	CBA (λ=0.5)
Average Return (last 1000 episodes)	99.79	98.79
Goal-Reaching Rate (per 100 episodes)	99.85	99.88
Fraction of Steps Spent in Loop (Loop-Time)	0.166	0.056

Interpretation

- Both agents learn to reach the goal at the same high level of reliability.
- Baseline agent spends **16.6%** of its time inside the reward loop.
- CBA-regularized agent spends **5.6%** of its time inside the loop.
- This is an **absolute reduction of 11.0%**, or a **~66% relative reduction**.

Conclusion

The CBA agent *substantially reduces reward-loop entrapment* without harming goal performance.

This is the intended effect: avoiding narrow local attractors while still maximizing reward.

7. Limitations

This is a simple, deterministic environment with tabular Q-learning.
The method has not yet been tested on:

- stochastic transitions
- larger or continuous spaces
- neural network function approximators
- multi-loop environments
- partial observability
- multi-agent settings

These tests are planned for future work.

8. Proposed Next Steps

1. *Share this note + code with RL engineers* to obtain feedback on which next environment is most meaningful.
2. Test under *stochastic transitions*.

3. Add *multiple loops* with varying reward structures.
4. Replace tabular Q with a small *DQN* to test whether CBA scales to function approximation.
5. Evaluate on environments with *misleading local optima*, such as:
    - mountain car with shaped traps
    - modified cliff-walking
    - gridworlds with partial observability
6. Package results into a short arXiv preprint after 3–5 experiments.

This staged approach mirrors standard safety research methodology and avoids premature conclusions.

9. Repository Access

The full code used to run the experiment will be included in the accompanying GitHub repository, along with:

- complete source files
- instructions for reproduction
- seed-averaged results
- experiment logs
- environment diagrams

10. Summary

This preliminary test offers the first empirical evidence supporting the claim that *coherence penalties* can reduce pathological reward-seeking behavior in RL agents.

While extremely early, the results justify further investigation and more rigorous evaluations.

CBA is not a complete alignment method, but this experiment demonstrates that coherence-regularization behaves as intended in a simple RL setting. Further evaluation in richer environments is required.

Code Repositry

https://github.com/abdulazizmohamed-dotcom/cba-gridworld-experiment/blob/main/cba_experiment.py