Causal Expectations for Decentralised Coordination in Multi-Agent Systems

John R. E. Mills[1][0009−0004−2111−6245] and Peter R. Lewis[2][0000−0003−4271−8611]

[1] Ontario Tech University, Oshawa ON, Canada john.mills@ontariotechu.net
[2] Ontario Tech University, Oshawa ON, Canada
peter.lewis@ontariotechu.ca

Abstract. Coordination in multi-agent systems (MAS) is commonly seen as a problem of distributed decision-making under shared constraints. However, coordination within decentralised MAS environments also requires agents to anticipate the causal consequences of their own actions and those of others. This paper explores whether decentralised coordination can emerge through the creation and reuse of explicit causal expectations without communication, centralised control, or hardcoded coordination rules. We implement Popperian Expectations in autonomous agents operating in an intersection environment without coordination mechanisms. We represent causal expectations using the Expectation Event Calculus (EEC), enabling agents to form them internally and revise them when they are falsified. Over 2000 crossing attempts, agents achieved a 98.6% task completion rate with 733,116 total simulations, compared to the baseline model which achieved 99.9% task completion but with a total of 6,530,568 total simulations. The results show that agents converge to a stable set of causal expectation rules while significantly reducing the number of simulations needed, with minimal reduction in safety. The results suggest that explicit causal expectations can provide a promising mechanism for decentralised coordination in MAS.

Keywords: Expectation · Coordination · Contention · Beliefs · Popperian Expectations · Expectation Event Calculus · Multi-agent Systems · Causal Expectations · Causal Learning

1 Introduction

Coordination in multi-agent systems (MAS) is commonly framed as a problem of distributed decision-making, in which agents must align their actions to satisfy objectives or constraints using game-theoretic, algorithmic, or logical mechanisms [1]. Traditional approaches to coordination in multi-agent systems primarily focus on selecting mutually compatible strategies under predefined constraints or payoff structures. However, we argue that coordination also requires causal anticipation, in which agents can represent and reason about which actions will cause rather than merely identify compatible strategies.

Previous work, such as the Popperian Expectations Framework [2] [3], describes the formal mechanisms for extracting, storing, using, and revising explicit causal expectations learnt from internal models.

To accomplish this, we propose implementing Popperian Expectations in decentralised agents to allow agents to create and use causal rules for decision-making. We showcase this in an autonomous intersection navigation task where four different agents all approach a shared intersection from different directions. The environment has no traffic signals, means of communication, or specified right-of-way rules. In the environment, agents must learn when it is safe to proceed and when it must stop purely from learnt causal expectations.

Popperian Expectations are structured condition-effect rules represented in the Expectation Event Calculus (EEC) [4]. When confidence in a causal expectation exceeds a threshold, the agent can reuse the expectations to guide their action selection, reducing redundant simulations.

In this paper, we show that decentralised coordination can be achieved through the use of explicit causal expectations where agents use confidence-gated causal expectations to bypass simulations, falling back onto simulations only when no causal expectation exists for that scenario. In an experimental extension of this concept, agents achieved a 98.63% task completion rate in a decentralised contention environment over 2000 total attempts while using substantially fewer simulations than a Monte Carlo baseline.

This paper seeks to make these contributions.

1. Implementation and use of explicit causal expectations in MAS using Popperian Expectations.
2. A proof of concept demonstration that causal expectations can support sufficient decentralised coordination.

2 Related Work

2.1 Winfield's Consequence Engine

The architecture first described by Winfield [7] [8] introduced the framework for consequence engines within autonomous agents. The consequence engine framework enabled agents to reason internally about their actions and internally test different hypotheses before acting upon them in the real world. However, their knowledge was transient, so they did not possess the ability to have long-term causal reasoning within their environment.

2.2 Popperian Expectations

The architecture for Popperian Expectations [2] [3] introduced an extension to Winfields consequence engine framework by allowing agents to capture and use causal knowledge using the EEC. This enabled agents to reason about causal expectations within their environment and capture previously transient knowledge as causal rules to be reused and updated instead of relying purely on simulation.

2.3 Expectation Event Calculus

The Expectation Event Calculus (EEC) is a formal language to represent and reason about causal expectations [4]. It was introduced as an extension of the Event Calculus (EC) [9]. The EEC allows for the representation of events and their effects, allowing explicit representation of cause-and-effect relationships.

2.4 Monte Carlo

Monte Carlo methods estimate action values by randomly sampling repeatedly rather than performing exhaustive searches [10]. Flat Monte Carlo does not require domain-specific heuristics, only a forward model of the environment [10]. Unlike the proposed approach, Monte Carlo does not store or reuse causal knowledge and therefore repeatedly resimulates similar scenarios.

3 Decentralised Coordination Setting

3.1 Environmental Model

We consider a decentralised multi-agent environment in which agents must traverse a shared intersection without communication, predefined coordination rules, or centralised control. At each timestep, agents select one action from the set

$$U = \{\text{SPEED}, \text{GO}, \text{SLOW}, \text{STOP}\}.$$

Collisions occur when two or more agents occupy the same spatial region simultaneously. The goal of each agent is to successfully cross the intersection while avoiding collisions and deadlocks.

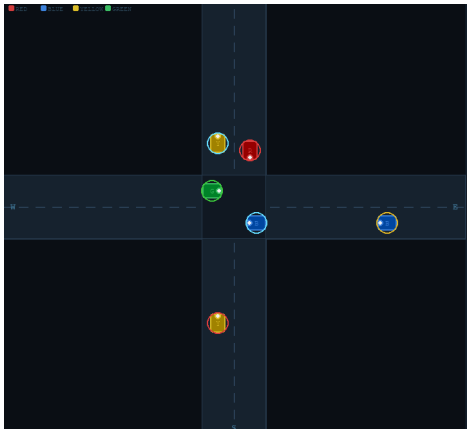

Fig. 1. Model of the MAS environment used within the experiment.

3.2 Coordination Criteria

Successful coordination is evaluated using task completion and collision avoidance. Agents are considered successful if they reach their destination without collisions or deadlock.

4 Expectation-Driven Coordination

4.1 Expectation Definition

An expectation is a structured causal representation that captures the predicted relationship between conditions and their anticipated effects. Within the Popperian Expectations framework, expectations are formed through internal simulation or observation and stored for reuse in future decision-making.

Expectations are represented as bounded executable EEC-style rules of the form `exp_rule (Cond, Exp)`. For example, an agent may learn that a TBoneCollision fluent is initiated if it executes GO action when another vehicle approaches the intersection with low relative time to intersection. The system represents these expectations as condition-effect rules, which are reused in future decision-making. Expectations are formed through internal simulation, stored in memory, and reused when confidence exceeds a predefined threshold. Repeated confirmations of expectations increase confidence, while violated expectations are weakened or pruned. The confidence thresholds were empirically selected for the implementation and were not theoretically optimised.

5 Agent Architecture

5.1 Agent Control Loop

At each time step, agents evaluate candidate actions using either stored expectations or bounded forward simulation. If a sufficiently confident expectation exists for the current context, the agent reuses the stored expectation and bypasses simulation. Otherwise, the agent performs a bounded forward simulation to evaluate actions and generate new expectations. After execution, expectations are confirmed or weakened based on observed outcomes. The forward simulation uses the current local observation state of the agent, including relative time to reach the intersection, nearby agents, intersection occupancy, and movement direction, to predict future states over a bounded horizon. Agents possess a deterministic forward model of environmental dynamics by which they evaluate candidate actions through an internal simulation before execution. Expectations are extracted from simulated state transitions and stored as condition-effect rules. When expectation confidence exceeds a defined threshold, the agent will then bypass simulation and reuse the stored expectation for its action selection.

6 Methodology

This experiment evaluates whether decentralised coordination can emerge through internally formed causal expectations without communication or predefined coordination rules. The simulation was run for 2000 crossing attempts using four agents operating in the shared intersection environment.

Monte Carlo Baseline Comparison To evaluate the contribution and usefulness of using causal expectation learning, we compare our results with a Monte Carlo baseline agent. This agent was implemented using an identical simulation environment and action set. The baseline does not create, store, or reuse any causal knowledge it has learnt from its simulations, which requires it to resimulate every potential action regardless of how often it has seen that scenario. The tested environment is simplified as a proof of concept setting consisting of a fixed four-way intersection where agents only travel straight through the intersection without communicating and without any heterogeneous behaviours.

7 Results

Over 2000 total attempts, agents achieved 1946 successful crossings and 27 crashes, yielding a success rate of 98.63%. The agents learnt 40,544 causal expectation rules and pruned 856 low-confidence rules. The agents reused causal expectations 245,286 times and relied on simulation 733,116 times.

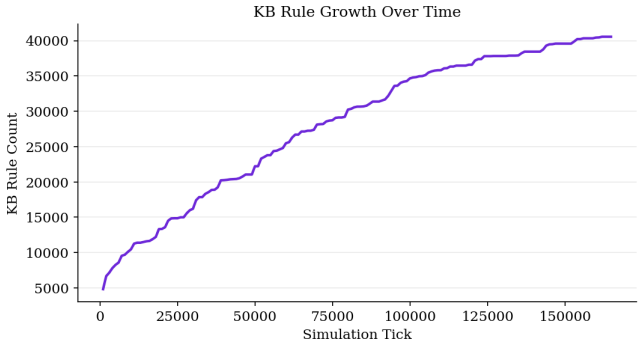

Fig. 2. The growth of KB rules over time

As shown in Fig. 2, the number of learned expectations rapidly increases before stabilising, suggesting convergence toward a stable causal representation of the environment.

Metric	EEC	Baseline
Success rate	98.63%	99.95%
Crashes	27	1
Total simulations	733,116	6,530,568
Simulations per attempt	366.6	3,265.3
KB coverage	67.6%	–
Conflict alignment	0.975	0.917
Rules learned	40,544	–
Rules pruned	856	–

Table 1. Performance comparison between EEC and the baseline simulation.

7.1 Baseline comparison

The baseline achieved a slightly higher success rate but required 8.9 times more simulations. The EEC agent achieved comparable coordination performance while substantially reducing simulation overhead by reusing expectations. Conflict alignment measures the degree in which agents select mutually compatible actions when in conflict.

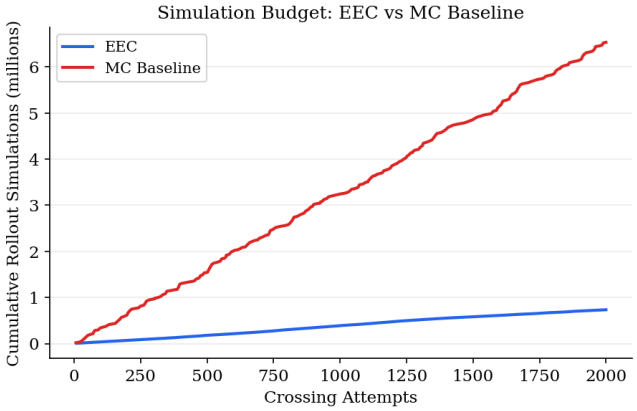

Fig. 3. The number of simulations used by each agent program over time

As shown in Fig. 3 the number of simulations used by the baseline drastically exceeds the number used by the EEC program.

8 Conclusion

The results demonstrate that explicit causal expectations can support decentralised coordination while substantially reducing simulation overhead. The agents converged on reusable causal expectations and achieved high task completion

without communication or predefined coordination rules. However, current implementation remains limited to a small deterministic environment, and scalability to more complex coordination scenarios remains an open challenge. Future work will explore the integration of causal expectations with more advanced coordination frameworks. “‘

References

1. Shoham, Y., Leyton-Brown, K.: Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations. Cambridge University Press, Cambridge (2008)
2. Anonymous Authors: Title withheld for review, ACSOS-C 2025
3. Anonymous Authors: Title withheld for review, ACSOS-C 2025
4. Cranefield, S.: Agents and Expectations. In: Balke, T., Dignum, F., van Riemsdijk, M.B., Chopra, A.K. (eds.) Coordination, Organizations, Institutions, and Norms in Agent Systems IX. Springer International Publishing, pp. 234–255 (2014)
5. Dennett, D.C.: Kinds of Minds: Toward an Understanding of Consciousness. Basic Books, New York (1996)
6. Dennett, D.C.: Darwin's Dangerous Idea: Evolution and the Meanings of Life. Simon & Schuster, New York (1995)
7. Winfield, A.F.: Robots with Internal Models: A Route to Self-Aware and Hence Safer Robots. In: The Computer After Me, pp. 237–252 (2014).
8. Winfield, A.F.T.: How Intelligent is your Intelligent Robot? arXiv preprint arXiv:1712.08878 (2017). DOI: 10.48550/arXiv.1712.08878
9. Shanahan, M.: The Event Calculus Explained. In: Wooldridge, M., Veloso, M. (eds.) Artificial Intelligence Today: Recent Trends and Developments. Springer, Berlin, Heidelberg, pp. 409–430 (2001)
10. Browne, C.B., Powley, E., Whitehouse, D., Lucas, S.M., Cowling, P.I., Rohlfshagen, P., Tavener, S., Perez, D., Samothrakis, S., Colton, S.: A Survey of Monte Carlo Tree Search Methods. IEEE Transactions on Computational Intelligence and AI in Games 4(1), 1–43 (2012)