applied sciences

MDPI

Article

Intelligent Agents and Causal Inference: Enhancing Decision-Making through Causal Reasoning

Jairo Iván Vélez Bedoya [1,2], Manuel González Bedia [1] and Luis Fernando Castillo Ossa [2,3,*]

[1] Departamento de Informatica e Ingenieria de Sistemas, Universidadde Zaragoza, 50009 Zaragoza, Spain; 790853@unizar.es (J.I.V.B.); mgbedia@unizar.es (M.G.B.)
[2] Departamento de Sistemas e Informática, Facultad de Ingeniería, Universidad de Caldas, Manizales 170001, Colombia
[3] Departamento de Ingenieria Industrial, Universidad Nacional de Colombia Sede Manizales, Manizales 170001, Colombia
* Correspondence: luis.castillo@ucaldas.edu.co

Abstract: This study examines the incorporation of causal inference methods into intelligent entities and examines the benefits of utilizing causal reasoning to improve decision-making procedures. This study entails conducting an experimental evaluation within a video game setting to evaluate the performance of three separate agent types: ExplorerBOT, GuardBOT, and CausalBOT. The ExplorerBOT utilizes a stochastic path selection technique for task completion, whereas the GuardBOT remains immobile yet exhibits exceptional proficiency in identifying and neutralizing other bots. On the other hand, the CausalBOT utilizes sophisticated causal inference methods to examine the underlying factors contributing to the failures noticed in the task completion of the ExplorerBOT. The aforementioned feature allows CausalBOT to make informed decisions by selecting paths that have a greater likelihood of achieving success. The main purpose of these experiments is to assess and compare the effectiveness of two distinct bots, namely ExplorerBOT and CausalBOT, in accomplishing their respective objectives. To facilitate comparison, two iterations of the ExplorerBOT are utilized. The initial iteration is predicated exclusively on stochastic path selection and necessitates a more profound understanding of the variables that impact the achievement of tasks. On the other hand, the second version integrates an algorithm for informed search. In contrast, CausalBOT employs causal inference techniques to discover the underlying causes of failures exhibited by ExplorerBOTs and collect pertinent data. Through the process of discerning the fundamental causal mechanisms, CausalBOT is able to make well-informed decisions by selecting pathways that maximize the probability of successfully completing a given job. The utilization of this approach greatly boosts the decision-making powers of CausalBOT, hence enabling it to effectively adapt and overcome problems in a more efficient manner when compared to alternative agents.

Keywords: causal inference; artificial intelligence; intelligent agents; decision-making; causal reasoning

check for updates

Citation: Vélez Bedoya, J.I.; González Bedia, M.; Castillo Ossa, L.F. Intelligent Agents and Causal Inference: Enhancing Decision-Making through Causal Reasoning. *Appl. Sci.* **2024**, *14*, 3818. https://doi.org/10.3390/app14093818

Academic Editor: Yutaka Ishibashi

Received: 26 August 2023
Revised: 7 December 2023
Accepted: 27 December 2023
Published: 30 April 2024

1. Introduction

Intelligent agents, powered by artificial intelligence and machine learning, have become increasingly prevalent in various decision-making tasks across fields such as video games [1], autonomous driving [2], and IoT systems [3], among others. An intelligent agent perceives its environment, takes autonomous actions to achieve goals, and can improve performance through learning and knowledge acquisition [4]. They are designed to learn from collected data, adapt to their environment, and make informed decisions to carry out tasks.

In the advent of the fourth industrial revolution, the lack of transparency in artificial intelligence-based systems presents a pivotal hindrance to their use, leading to the application of explainable AI (XAI) [5] to these systems. For instance, conventional intelligent agents are trained to perform tasks without considering the inherent causal relationships underlying the problem they must solve. This is an issue that must be addressed. However, as pointed out by Maes et al. [6], incorporating causal inference into intelligent agents is challenging due to numerous hidden variables within the model. Nevertheless, there has been a recent surge of interest in developing algorithms that generate interpretable agent behavior regarding goals, plans, or rewards, as discussed by Chakraborti et al. [7]. That has paved the way for research, such as that of Neufeld and Kristtorn [8], Jezic et al. [9], and Meganck et al. [10], demonstrating the feasibility of integrating probabilistic reasoning and causal maps into the logic of intelligent agents. By adopting causal reasoning, intelligent agents can move beyond merely identifying correlations and discerning the fundamental causes of events, facilitating more robust, reliable, and interpretable decision-making processes. For example, Dasgupta et al. [11] have adopted "meta-reinforcement learning" to generate an agent capable of executing tasks through causal inferences, even without explicit knowledge of causality. Similarly, regarding designing a smart grid power system, in order to ensure that the system performs well and reduces the communication bandwidth, ref. [12] proposes a causal inference communication model (CICM). The efficiency of their algorithm was demonstrated in their experiments using navigation tasks in the virtual world of StarCraft II, a video game. Miao et al. [13] has presented a dynamic inference agent that uses numerical representations instead of symbolic representations for modeling, inference, and decision-making. Also, Ceballos and Cantu [14] have suggested a design method and agent architecture that build on the Beliefs, Desires, and Intentions (BDI) framework that [15] described for creating intelligent agents. Highlighting the difficulty of accurately modeling an agent's causal structure, Jensen [16] has emphasized this challenge using the Angry Birds AI Competition as an example, where agents must analyze levels and predict the physical consequences of their actions to achieve high scores, as described by Renz et al. [17] In response to this competency, Tziortziotis et al. [18] have developed an agent architecture employing Bayesian inference to enhance decision-making abilities.

Although some studies have addressed the development of intelligent agents that incorporate causal inference in their learning, there needs to be more experimentation on this topic, indicating the need for more empirical studies to gain a deeper understanding of how causal inference can enhance the decision-making capabilities of intelligent agents.

This research will thoroughly examine the intersection between intelligent agents and causal inference to explore how incorporating causal reasoning can significantly enhance decision-making abilities and task execution, providing these agents with a distinctive advantage over other systems. The primary purpose of this investigation is to contribute to the growing body of knowledge in the fields of intelligent agent systems and causal inference, shedding light on the promising potential of merging these areas to create more informed and transparent decision-making systems.

The present article is organized into four sections. In Section 1, we discuss the motivation behind the study, present notable contributions to the development of intelligent agents endowed with causal reasoning, and position our work as one of the pioneering initiatives exploring the advantages and possibilities presented by agents utilizing causal inference for decision-making and task accomplishment. In Section 2, we present the methodology for generating the virtual environment where the agents interact, specifying the behavior of three agent types: GuardBOT (GBOT), ExplorerBOT (EBOT), and CausalBOT (CBOT), which form an integral part of our experiment. We also describe the data generation and causal inference processes employed to test the hypothesis and determine the most effective task-completion agent. In Section 3, we analyze the results obtained from each process considered in the methodology. Finally, in In Section 4 we discuss the conclusions derived from the study.

2. Materials and Methods

The main goal of this experimental study was to develop an intelligent agent that utilizes causal inference to understand the reasons behind other agents' failures to complete a task. Subsequently, the agent is designed to execute the task by making informed decisions that increase the likelihood of successful completion.

2.1. Task Design

The task involved an intelligent agent navigating through a virtual environment, aiming to reach the position occupied by another agent who was in the same environment without being detected. Sections 2.3 and 2.5 provide detailed information about each participating agent in the task. The experiment's stages included designing and configuring the virtual environment, developing the explorer (EBOT) and guard (GBOT) agents, data collection, and implementing the causal agent (CBOT).

2.2. Scenario Design and Virtual Environment Configuration

For the experiments, we utilized Unreal Tournament, a First-Person Shooter (FPS) video game that enables developers to configure environments and program bot logic before deploying them into the virtual world. We employed Unreal Editor [19] to craft the test scenario, comprising a series of interconnected traversable corridors, as shown in Figure 1, to ensure access to any point of interest in the map through seven distinct paths:

- Path: S-A-C-E-H-K-T
- Path: S-B-D-E-H-K-T
- Path: S-A-C-F-H-K-T
- Path: S-B-D-G-H-K-T
- Path: S-A-C-F-I-K-T
- Path: S-B-D-G-J-K-T
- Path: S-B-D-G-J-M-N-T

These paths were carefully designed to ensure accessibility to various areas of interest in the virtual environment.

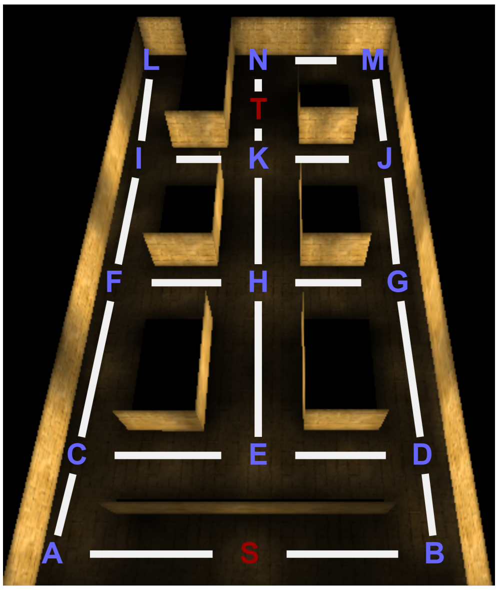


Figure 1. Configured paths in the game map.

Pogamut is a crucial component that ensures seamless agent interaction within the virtual environment. It serves as a middleware, empowering virtual agents' control across various environments game engines offer. Pogamut provides a Java API to generate and manage virtual agents and a user-friendly graphical interface for streamlined debugging purposes [20]. The architectural model facilitating the integration of CBOT with the virtual environment via Pogamut is illustrated in Figure 2.

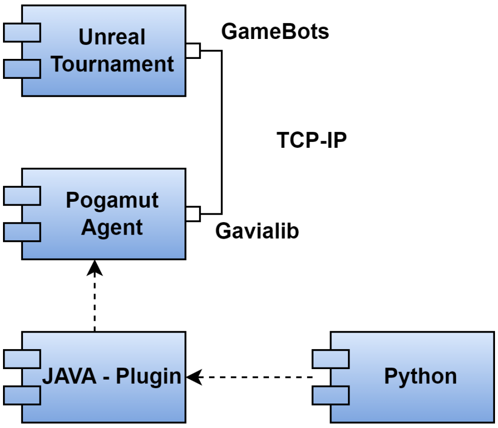
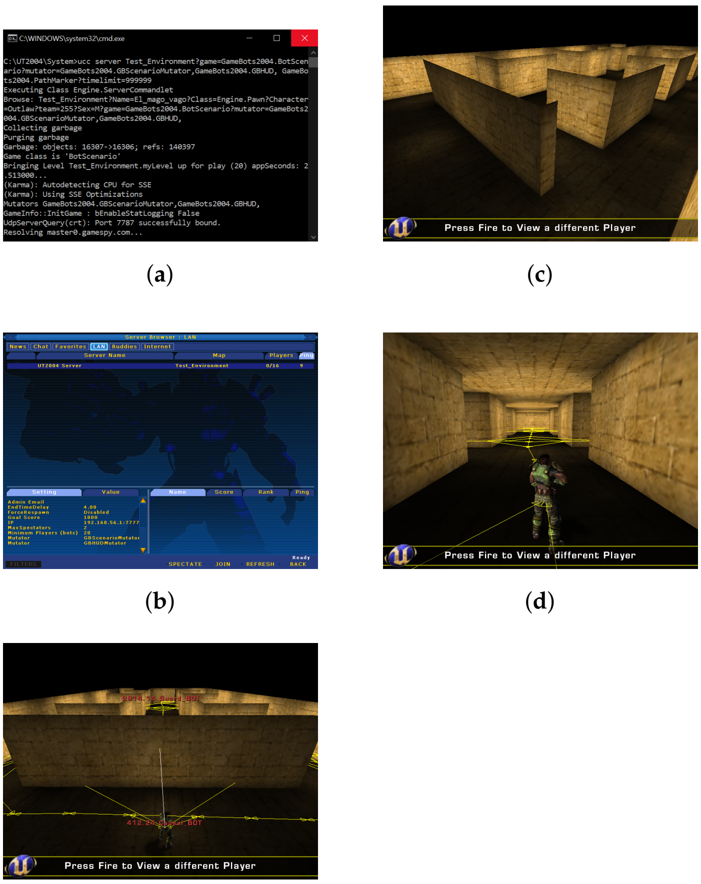


Figure 2. Architectural model for the execution of CBOT.

After installing the middleware, we configured the script for launching the DeathMatch server to load our specific test scenario. To develop the agent's behavior models, we used the Eclipse IDE tool and the pre-defined EmptyBot template provided by Pogamut. The execution process of a match is depicted in Figure 3.



(**a**) (**c**)

(**b**) (**d**)

(**e**)

Figure 3. Execution process of a match. (**a**) Console. (**b**) Server. (**c**) Empty Scenario. (**d**) GuardBOT View. (**e**) CausalBOT View.

The sequence of actions during the match execution is as follows: first, we executed the server responsible for loading the designated test environment (Figure 3a). Then, UT2004 was connected to this server (Figure 3b), resulting in the appearance of the empty test scenario (Figure 3c). Subsequently, we launched GBOT (Guard Bot) and EBOT (Explorer

Bot) from Eclipse into the virtual environment. Figure 3d,e depict the views of CBOT and GBOT, respectively.

By utilizing Pogamut version 3.7.0 and Eclipse® version 4.29.0 (Build 20230907-1323), we created a robust and efficient platform for designing, testing, and implementing the intricate behaviors of our agents. These tools were critical in bringing our vision of intelligent agents to life within a controlled and reproducible experimental setting.

2.3. Development of Explorer (EBOT) and Guardian (GBOT) Agents

The agents' characteristics were inherited from the avatars interacting in the video game Unreal Tournament [21]. However, we programmed their behavior models entirely using the Pogamut API.

2.3.1. Guardian Agent (GBOT)

The main task of the Guardian agent was to remove any other agent it identified approaching its position, following an alert from the virtual environment's global messaging system.

Behavior Model

The Guardian agent was designed to maintain a fixed position in the environment, with the flexibility to activate or deactivate its visual and auditory sensors. When the visual sensor was active and an enemy agent entered its field of view, the Guardian agent successfully identified them. On the other hand, if the visual sensor was inactive but the auditory sensor was active, the agent detected approaching footsteps and turned in the direction of the sound source, but it did not identify the agent producing the sound. However, when both sensors were active, the Guardian agent effectively detected and identified the approaching agent, regardless of the direction. The logic of the GBOT is clearly illustrated in Algorithm 1.

Algorithm 1: Logic process of GBOT

```
canSee ← agent.config.canSee;
canListen ← agent.config.canListen;
while true do
    if canSee then
        player ← game.players.getNearestPlayer() ;
        if player is not null then
            sendMessage(player, "you'vebeenidentified.");
            game.kick(player);
    if canListen is true then
        if isHearingNoise then
            turnTo(getNoiseSource())
```

2.3.2. Explorer Agent (EBOT)

The primary objective of the Explorer agent was to navigate a path that would lead it to the position occupied by the GBOT. The environment offered various paths, allowing the Explorer agent to approach the GBOT from different directions, whether from the front or the rear.

Behavior Model

We created two versions of this agent. The first version, which was used for generating data, utilized an uninformed search algorithm. The Explorer agent always started at a fixed point and randomly chose one of the available paths that would lead to the GBOT's location. The second version, which was used to evaluate the causal agent's performance, used A*, an informed search algorithm. The agent's execution ended in two ways: if the GBOT identified the agent or if the agent reached the GBOT's location successfully. Before concluding its execution, the Explorer agent recorded its performance data, which is shown in Table 1. The uninformed search EBOT and the informed search EBOT both follow the logic outlined in Algorithms 2 and 3, respectively.

Table 1. Structure of the dataset.

Label	Value	Description
BOT	[0. . . 1199]	The starting number that identifies EBOT
C	[1. . . 7]	The path taken by the EBOT
T_Nodes	[1. . . path_length]	The total number of nodes along the path
T_Visited	[1. . . path_length]	The number of nodes the explorer visited
Can_see	[0,1]	Can GBOT see?
Can_listen	[0,1]	Can GBOT listen?
Outcome	[0,1]	Did EBOT complete the task?

Algorithm 2: EBOT—Uninformed search

```
g ← generateGraph(game.map) ;
lstPaths ← getAllPaths(g, S, T) ;
(path, pathId) ← chooseRandomPath(lstPaths) ;
while true do
    visited ← 0 ;
    foreach node in path do
        agent.moveTo(node);
        visited ← visited + 1;
        if getMessage()is"you'vebeenidentified." then
            saveData(game.agentId, pathId, path.length, visited, game.GBOT.
             canListen(), game.GBOT.canSee(), 1);
            return;
        if nodeisT then
            saveData(game.agentId, pathId, path.length, visited, game.GBOT.
             canListen(), game.GBOT.canSee(), 0);
            return;
```

The temporal complexity of Algorithm A* is contingent upon both the chosen heuristic and the underlying structure of the graph. The best-case scenario, which is defined by a heuristic that is both allowable and consistent, has a computational complexity of O(d), where d is the length of the shortest path. The exponential nature of the worst-case scenario can be attributed to the presence of ineffective heuristics or complex graphs. The space complexity of a graph is determined by its size as well as the number of opened and closed nodes. In the event of the most unfavorable circumstances, the computational cost can also be substantial as a result of the storage space needed to retain data on traversed nodes.

How long it takes for the informed search algorithm to work depends on both how the graph is structured and which heuristic is used. In an optimal situation, the acceptable heuristic has the potential to approach a time complexity of O(d), where 'd' represents the length of the shortest path. The worst-case situation has the potential to exhibit exponential growth. An algorithm's space complexity depends on both the size of the graph it is processing and the particular data structures it uses.

Algorithm 3: EBOT- Informed search

```
Data: start, goal
Result: Shortest path from start to goal
Function A*(start, goal):
    openSet ← {start};
    cameFrom ← an empty map;
    gScore[start] ← 0;
    fScore[start] ← heuristic(start, goal);
    while openSet is not empty do
        current ← the node in openSet with the lowest fScore value;
        if current is goal then
            return reconstructPath(cameFrom, current);
        openSet.remove(current);
        for each neighbor of current do
            tentativeGScore ← gScore[current] + distance(current, neighbor);
            if tentativeGScore < gScore[neighbor] then
                cameFrom[neighbor] ← current;
                gScore[neighbor] ← tentativeGScore;
                fScore[neighbor] ← gScore[neighbor] + heuristic(neighbor, goal);
                if neighbor not in openSet then
                    openSet.add(neighbor);
    return an empty set
Function heuristic(node, goal):
    return | node.x − goal.x | + | node.y − goal.y | +penalty
```

2.4. Data Collection

To gather the necessary data, we conducted 1200 matches in the test environment, considering the complexity of the tasks assigned to the GBOT and EBOT agents. Each match followed the sequence of launching the GBOT first, followed by the EBOT. At the end of each match, the EBOT agent recorded and stored the relevant data.

Figure 4a depicts the starting positions of the agents in the virtual environment at the beginning of a match. Additionally, Figure 4b provides a snapshot of the dataset generated during one of these matches.

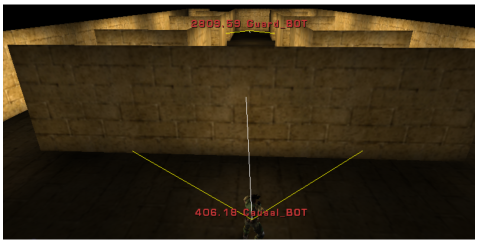
2.5. Implementation of Causal Agent (CBOT)

The role of the CBOT agent is identical to that of the EBOT, but its process is guided by causal inference before entering the virtual environment.

2.5.1. Causal Inference

The causal inference process commenced by loading the data from the 1200 matches. A structural causal model (SCM) was employed to estimate the joint distribution of the dataset [22]. Subsequently, we designed the SCM, illustrated in Figure 5, to further inform the CBOT agent's decision-making during its execution.

In this model, we considered the visual sensor (S) and the listening sensor (L) explanatory variables. The outcome (Y) was determined based on the state of these two sensors, with the CBOT believing one of these senses influenced the task's success. To test the validity of this belief, we consulted the model and predicted the effect of specific interventions on causal relationships, following the approach proposed by [23].



(**a**)

BOT	C	T-Nodes	T-Visited	Can_See	Can_Listen	Outcome
0	5	25	12	1	0	0
1	5	25	25	0	1	1
2	2	27	13	1	0	0
3	2	27	13	1	0	0
4	4	25	16	1	0	0
...	...	...	...	...	...	...
1195	3	24	24	0	1	1
1196	7	30	30	0	1	1
1197	2	27	13	1	0	0
1198	3	24	21	1	1	0
1199	7	30	26	1	1	0

(**b**)

Figure 4. Example of a match and the data generated. (**a**) Starting positions of the agents. (**b**) Snapshot of the dataset.

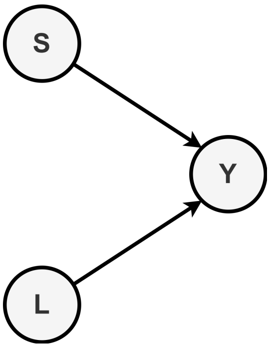


Figure 5. Structural Causal Model (SCM).

2.5.2. Conditional Probability Distribution (CPD)

A conditional probability distribution (CPD) is a table that shows the probabilities of one variable with respect to others [24]. According to [25] these structures allowed us to make queries to the model and reason about counterfactuals using Equation (1).

$$P(A \mid B) = P(A \cap B)/P(B) \tag{1}$$

where:

- $P(A \mid B)$: Represents the probability of event A occurring given that event B has already occurred.
- $P(A \cap B)$: Is the probability of both events A and B occurring simultaneously.
- $P(B)$: Is the probability of event B occurring, without any additional conditions.

We used the pgmpy package [26] to generate a CPD and conducted the following queries:

- What would happen if the GBOT could both see and listen?
- What would happen if the GBOT could see but not listen?
- What would happen if the GBOT could listen but not see?

To validate the results of these "what if" queries [27], we divided the sample into a control group and a treatment group using the causalnex package [28]. By employing Equation (2), we were able to quantify the effects of various interventions, such as activating or deactivating a sensor, using the average treatment effect (ATE). Through a comprehensive analysis of this metric, we have successfully ascertained the level of accuracy pertaining to the CBOT's confidence in its decision-making process.

$$ATE = \frac{1}{N} \underbrace{\sum_{i=1}^{N}(Y_1(i))}_{\text{Total outcome with treatment}} - \underbrace{\sum_{i=1}^{N}(Y_0(i))}_{\text{Total outcome without treatment}} \tag{2}$$

where:

- N: Number of samples.
- $Y_0(i)$: Outcome without treatment for sample i.
- $Y_1(i)$: Outcome with treatment for sample i.
- i: Index representing individual samples.

2.5.3. Behavior Model

After conducting the causal inference process, the CBOT gained insight into the underlying cause of failure based on its beliefs about achieving the task. Following a logical process similar to that of the EBOT, where the path is chosen randomly, the CBOT executes the path with the highest likelihood of success.

3. Results and Discussion

In the context of this research, we examined the evaluation of agent performance in a virtual environment where the main objective is to have agents (EBOT) move from an initial point to the position occupied by a guardian agent (GBOT) without being detected by the guardian agent, whose sense of sight and hearing may be activated or deactivated. Since this unique configuration has not been extensively explored in the academic literature consulted, direct comparisons with previous studies have proven difficult due to the lack of fully analogous studies. As a result, in order to place our findings in the broader context of decision-making methods, we conducted an evaluation by considering key metrics such as task success rate, agents' energy consumption, and the ratio of agents' energy consumption to their success and failure rates. While previous studies have focused on a number of different approaches and contexts, the comparative analysis we performed enables us to discuss the implications of our findings within the context of artificial intelligence-mediated decision-making.

3.1. Causal Inference

By calculating the conditional probability distribution, as shown in Table 2, we could consult the causal model to answer counterfactual questions. That allowed us to determine the probability of success ($Y = 1$) or failure ($Y = 0$) for each combination of states or possible values of the explanatory variables (L, S).

Table 2. CPD of the structural causal model.

L **S**	**L(0)** **S(0)**	**L(0)** **S(1)**	**L(1)** **S(0)**	**L(1)** **S(1)**
Y(0)	0.5	0.8204	0.0	1.0
Y(1)	0.5	0.1795	1.0	0.0

The CBOT believed that the success of task execution was determined by one of the GBOT's sensors, necessitating interventions for both sensors to reinforce this belief. Figure 6 illustrates the effects of the interventions resulting from the model queries, aiming to deduce the cause of task failure.

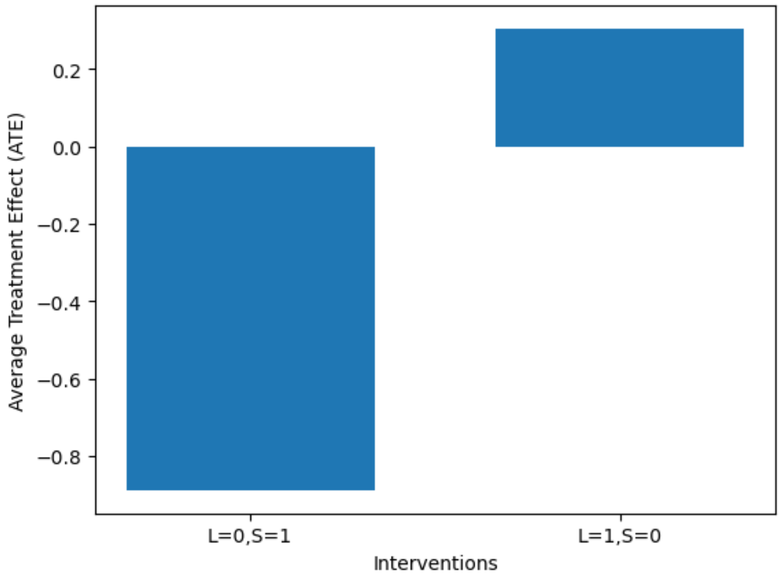


Figure 6. Effects of the interventions.

If the GBOT could see and listen $(S = 1, L = 1)$, the probability of failure $(Y = 0)$ expressed as $P(Y = 0|S = 1, L = 1) = 1.0$ was absolute, implying no possibility of achieving the task. Despite reinforcing the CBOT's belief, it was not possible to identify which sensor determined task success. On the other hand, if the GBOT could see but not hear $(S = 1, L = 0)$, the probability of failure $(Y = 0)$ expressed as $P(Y = 1|S = 1, L = 0) = 0.8204$ was 82.04% regardless of the path taken. Activating the vision sensor $(S = 1)$ had an unfavorable impact on the variable of interest (Y) with an effect of $ATE = -0.888$, which means that such activation has had an adverse effect or decreased the likelihood of the CBOT being undetected compared to the situation where the activation was not applied. The CBOT's belief was reinforced, making a first inference with considerable certainty about which sensor determined task success. Finally, if the GBOT could not see but could listen $(S = 0, L = 1)$, the probability of failure $(Y = 0)$ expressed as $P(Y = 1|S = 0, L = 1) = 0.0$ regardless of the path taken. Activating the hearing sensor $(L = 1)$ had a favorable impact on the variable of interest (Y) with an effect of $ATE = 0.304$ as long as the vision sensor remained inactive $(S = 0)$. That reinforced the belief that the vision sensor determined task success.

3.2. Performance Evaluation of CBOT

The performance of CBOT was compared to two EBOTs: one utilizing a blind search algorithm (uninformed) and the other an informed search algorithm (A*). Table 3 presents the results from 250 matches in the same testing scenario.

Table 3. Performance evaluation results.

Agent	Logic	Successes	Failures	Success Rate	Failure Rate	Energy (Success)	Energy (Failure)
EBOT-US [a]	Random search	75	175	30	70	76.39	49.5
EBOT-IS [b]	A*	198	52	79.2	20.8	37.7	84.9
CBOT	Causal inference	205	45	82	18	35.4	40.8

[a] EBOT Uninformed Search. [b] EBOT Informed Search.

As expected, the EBOT with the uninformed search algorithm performed poorly compared to others in Figure 7. The CBOT narrowly outperformed the EBOT with A*, primarily because the latter's heuristic calculation included a penalty for being detected while calculating the path, considering it a failure. However, this EBOT's behavior was based on a heuristic calculation that did not consider causal elements.

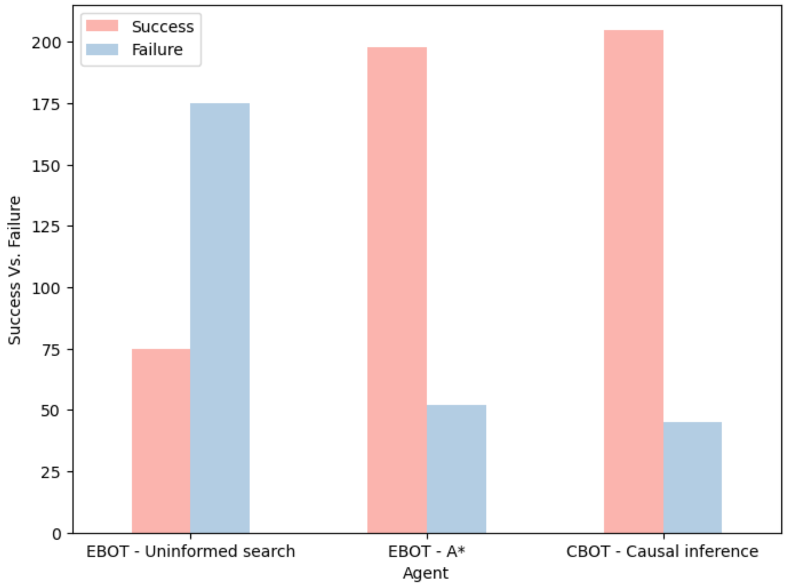


Figure 7. Validation results.

An analysis of the agents' success rate as a function of the GBOT view sensor status is presented in Figure 8. We observe a clear difference between cases in which the view sensor is enabled and cases in which it is disabled. The ability to evade detection is more effective when the enemy agent cannot use its sense of sight to track them when the view sensor is disabled.

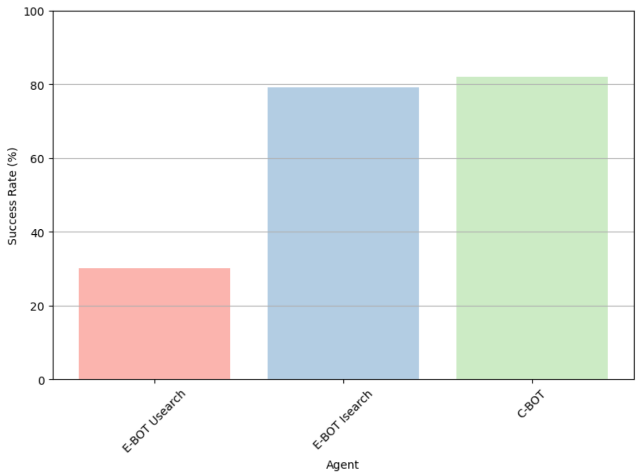


Figure 8. Success rate according to sight sensor.

An analysis of the relationship between the energy consumption of the agents and the state of the enemy agent's view sensor is presented in Figure 9. There is no doubt that agents operating in an environment where the view sensor is disabled consume less energy than those operating in environments where the view sensor is enabled. This finding illustrates how the ability to evade visual detection can affect the efficiency with which agents use their energy resources.

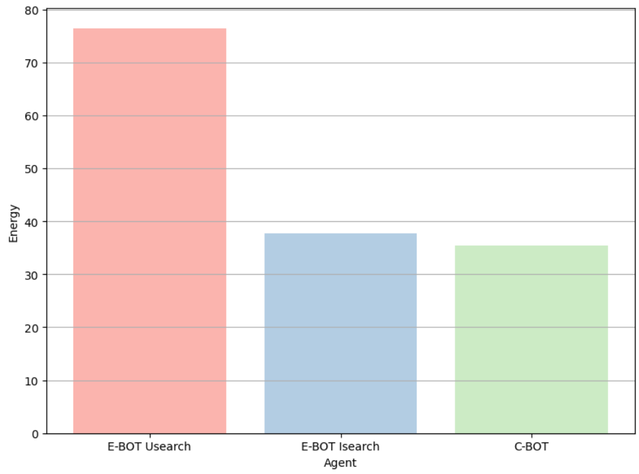


Figure 9. Energy consumption according to sight sensor.

Figure 10 illustrates the relationship between energy consumption and GBOT detection. As can be seen, energy consumption is generally lower in cases in which agents are not detected. This relationship, however, has some outliers that suggest a degree of variation. Accordingly, although there is a correlation between detection and energy consumption, other factors, such as the existence of a listening sensor, could also affect this factor.

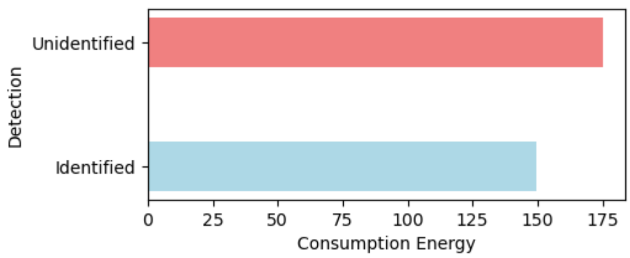


Figure 10. Energy consumption vs. Identification of EBOTs.

4. Conclusions

In this study, we investigated the convergence of intelligent agents and causal inference. Our findings shed light on the transformative effects of incorporating causal reasoning into task execution and decision-making, thereby bestowing a notable competitive advantage upon these agents.

The utilization of causal inference in the CBOT showcased a profound advantage over alternative methodologies. It provided a deep understanding of the causes behind detection events, which let the CBOT make smart choices and come up with plans that made task completion more efficient. In contrast, the EBOTs relied solely on random path selection, demanding a comprehensive understanding of influencing factors. Similarly, the GBOT focused on eliminating agents without explicit consideration of causal relationships.

This experiment showed how important conditional probability distributions (CPDs) are for figuring out what might happen and proving hypotheses. However, it also illuminated the contextual nature of these methods, as elucidated by [29], stressing the importance of reasoning techniques for intelligent agents operating in diverse physical simulation environments. The experiment also showed how powerful it could be to add causal inference to smart agents, especially concerning its use across numerous different situations where sensors can show what behaviors are caused, which would help them make better decisions.

For future experiments, it would be valuable to incorporate additional observations and design tasks in more complex scenarios. Furthermore, exploring real-time data generation for causal inference instead of relying solely on historical game data could present an exciting avenue, allowing the CBOT to adapt its strategies dynamically. Furthermore, the employed causal inference approach may necessitate refinements and customizations to suit applications in intricate and diverse domains.

Author Contributions: Conceptualization, methodology, formal analysis, M.G.B. and J.I.V.B.; software, J.I.V.B.; validation, writing—review and editing, M.G.B., J.I.V.B. and L.F.C.O.; writing—original draft preparation, J.I.V.B.; supervision, M.G.B. and L.F.C.O. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available at https://zenodo.org/badge/latestdoi/669330810.

Acknowledgments: Jairo I. Vélez: I thank the Universidad de Zaragoza and Banco Santander (Universidad de Zaragoza-Santander Universidades) for providing me with one of the scholarship for Ibero-Americans in doctoral studies, thus making this research possible.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

API	Application Programming Interface
ATE	Average Treatment Effect
BDI	Beliefs Desires Intentions
CBOT	Causal BOT
CPD	Conditional Probability Distribution
EBOT	Explorer BOT
GBOT	Guard BOT
SCM	Structural Causal Model
CICM	Causal Inference Communication Model

References

1. Zhao, Y.; Borovikov, I.; de Mesentier Silva, F.; Beirami, A.; Rupert, J.; Somers, C.; Harder, J.; Kolen, J.; Pinto, J.; Pourabolghasem, R.; et al. Winning Is Not Everything: Enhancing Game Development With Intelligent Agents. *IEEE Trans. Games* **2020**, *12*, 199–212. [CrossRef]
2. Palanisamy, P. Multi-agent connected autonomous driving using deep reinforcement learning. In Proceedings of the International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, 19–24 July 2020; pp. 1–7.
3. Lei, L.; Tan, Y.; Zheng, K.; Liu, S.; Zhang, K.; Shen, X. Deep reinforcement learning for autonomous internet of things: Model, applications and challenges. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 1722–1760. [CrossRef]
4. Russell, S.J. *Artificial Intelligence a Modern Approach*; Pearson Education, Inc.: London, UK, 2010.
5. Adadi, A.; Berrada, M. Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI). *IEEE Access* **2018**, *6*, 52138–52160. [CrossRef]
6. Maes, S.; Meganck, S.; Manderick, B. Inference in multi-agent causal models. *Int. J. Approx. Reason.* **2007**, *46*, 274–299. [CrossRef]
7. Chakraborti, T.; Kulkarni, A.; Sreedharan, S.; Smith, D.E.; Kambhampati, S. Explicability? legibility? predictability? transparency? privacy? security? The emerging landscape of interpretable agent behavior. In Proceedings of the International Conference on Automated Planning and Scheduling, ICAPS, Berkeley, CA, USA, 11–15 July 2019; pp. 86–96. [CrossRef]
8. Neufeld, E.; Kristtorn, S. Does non-correlation imply non-causation? *Int. J. Approx. Reason.* **2007**, *46*, 257–273. [CrossRef]
9. Jezic, G.; Kusek, M.; Lovrek, I.; Howlett, R.J.; Jain, L.C. Preface. *Adv. Intell. Syst. Comput.* **2014**, *296*, 5–6. [CrossRef]
10. Meganck, S.; Maes, S.; Manderick, B.; Leray, P. Distributed learning of multi-agent causal models. In Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology, IAT'05, Compiegne, France, 19–22 September 2005; Volume 2005, pp. 285–288. [CrossRef]
11. Dasgupta, I.; Wang, J.; Chiappa, S.; Mitrovic, J.; Ortega, P.; Raposo, D.; Hughes, E.; Battaglia, P.; Botvinick, M.; Kurth-Nelson, Z. Causal Reasoning from Meta-reinforcement Learning. *arXiv* **2019**, arXiv:1901.08162.
12. Zhang, X.; Liu, Y.; Li, W.; Gong, C. Pruning the Communication Bandwidth between Reinforcement Learning Agents through Causal Inference: An Innovative Approach to Designing a Smart Grid Power System. *Sensors* **2022**, *22*, 7785. [CrossRef] [PubMed]

13. Miao, C.; Goh, A.; Miao, Y.; Yang, Z. A dynamic inference model for intelligent agents. *Int. J. Softw. Eng. Knowl. Eng.* **2001**, *11*, 509–528. [CrossRef]
14. Ceballos, H.G.; Cantu, F.J. Modelling intelligent agents through causality theory. In Proceedings of the 6th Mexican International Conference on Artificial Intelligence, Special Session—MICAI 2007, Aquascalientes, Mexico, 4–10 November 2007; pp. 201–210. [CrossRef]
15. Mascardi, V.; Demergasso, D.; Ancona, D. Languages for programming BDI-style agents: An overview. In Proceedings of the WOA 2005—6th AI*IA/TABOO Joint Workshop "From Objects to Agents": Simulation and Formal Analysis of Complex Systems, Camerino, Italy, 14–16 November 2005; pp. 9–15.
16. Jensen, D.D. Improving Causal Inference by Increasing Model Expressiveness. In Proceedings of the 35th AAAI Conference on Artificial Intelligence, AAAI 2021, Virtual, 2–9 February 2021; Volume 35, pp. 15053–15057. [CrossRef]
17. Renz, J.; Ge, X.Y.; Verma, R.; Zhang, P. Angry birds as a challenge for artificial intelligence. In Proceedings of the 30th AAAI Conference on Artificial Intelligence, AAAI 2016, Phoenix, AZ, USA, 12–17 February 2016; pp. 4338–4339. [CrossRef]
18. Tziortziotis, N.; Papagiannis, G.; Blekas, K. A Bayesian Ensemble Regression Framework on the Angry Birds Game. *IEEE Trans. Comput. Intell. AI Games* **2016**, *8*, 104–115. [CrossRef]
19. Epic Games. Unreal Engine. Available online: https://www.unrealengine.com/es-ES (accessed on 12 June 2023)
20. Gemrot, J.; Kadlec, R.; Bida, M.; Burkert, O.; Pibil, R.; Havlicek, J.; Zemcak, L.; Simlovic, J.; Vansa, R.; Stolba, M.; et al. Pogamut 3 Can Assist Developers in Building AI (Not Only) for Their Videogame Agents. In *Agents for Games and Simulations: Trends in Techniques, Concepts and Design*; Springer: Berlin/Heidelberg, Germany, 2012; Volume 7522, pp. 1–15.
21. Adams, D. Unreal Tournament 2004 Review Does Epic's Latest Do the Series Proud? Available online: https://www.ign.com/articles/2004/03/12/unreal-tournament-2004-review (accessed on 29 May 2023).
22. Pearl, J.; Glymour, M.; Jewell, N.P. *Causal Inference in Statistics: A Primer*; John Wiley & Sons Ltd.: Chichester, UK, 2016; ISBN 978-1-119-18686-1.
23. Pearl, J.; Mackenzie, D. *The Book of Why*, 1st ed.; Basic Books: New York, NY, USA, 2018.
24. Murphy, K.P. *Machine Learning: A Probabilistic Perspective*; The MIT Press: Cambridge, MA, USA, 2012; Volume 16, pp. 1–1098.
25. Pawlowski, N.; Castro, D.C.; Glocker, B. Deep structural causal models for tractable counterfactual inference. In Proceedings of the 34th International Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 6–12 December 2020.
26. Ankan, A.; Panda, A. pgmpy: Probabilistic graphical models using python. In Proceedings of the 14th Python in Science Conference (SCIPY 2015), Austin, TX, USA, 6–12 July 2015.
27. Hernán, M.A.; Robins, J.M. *Causal Inference: What If*, 1st ed.; Taylor & Francis Group: Abingdon, UK, 2020; p. 312.
28. Beaumont, P.; Horsburgh, B.; Pilgerstorfer, P.; Droth, A.; Oentaryo, R.; Ler, S.; Nguyen, H.; Ferreira, G.A.; Patel, Z.; Leong, W. CausalNex. Available online: https://github.com/mckinsey/causalnex (accessed on 4 April 2023).
29. Stephenson, M.; Renz, J. Creating a hyper-agent for solving angry birds levels. In Proceedings of the 13th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, AIIDE 2017, Snowbird, UT, USA, 5–9 October 2017; pp. 234–240. [CrossRef]