On Human-like Performance Artificial Intelligence – A Demonstration Using an Atari Game

Seng-Beng Ho[1], Xiwen Yang[1], Therese Quieta[1], Gangeshwar Krishnamurthy[1]

and Fiona Liausvia[1]

[1] AI Progamme, A*STAR, Singapore
hosb@ihpc.a-star.edu.sg, Yang_Xiwen@scei.a-star.edu.sg,
therese_quieta@ihpc.a-star.edu.sg,
gangeshwar_krishnamurthy@ihpc.a-star.edu.sg,liausviaf@ihpc.a-star.edu.sg

Abstract. Despite the progress made in AI, especially in the successful deployment of deep learning for many useful tasks, these systems typically require a huge number of training instances, and hence a long time for training. As a result, these system are are not able to rapidly adapt to changing rules and constraints in the environment. This is unlike humans, who are usually able to learn with only a handful of experiences. This hampers the deployment of, say, an adaptive robot that can learn and act rapidly in the ever-changing environment of a home, office, factory, or disaster area. Thus, it is necessary for an AI or robotic system to achieve human performance not only in terms of the "level" or "score" (e.g., success rate in classification, score in Atari game playing, etc.) but also in terms of the speed with which the level or score can be achieved. In contrast with earlier DeepMind's effort on Atari games, in which they demonstrated the ability of a deep reinforcement learning system to learn and play the games at human level in terms of score, we describe a system that is able to learn causal rules rapidly in an Atari game environment and achieve human-like performance in terms of both *score* and *time*.

Keywords: Causal Learning, Human-like Performance, Atari Game Playing, Space Invaders Game Playing, Problem Solving.

1 Introduction

Artificial intelligence (AI) has taken great strides in many domains of applications, but there has been realization that even though many AI systems can perform certain tasks very well that normally require human intelligence, and sometimes even superseding human abilities in those tasks, their performance is not "human-like" in some aspects. For example, when deep learning is applied to pattern classification and recognition, the accuracy is very high and sometimes outstrips human performance, but humans usually require only a few instances of training examples to learn to classify and recognize the objects involved with high accuracy, whereas deep learning systems typically require many orders of magnitude of the number of training examples needed by humans. Thus, we can distinguish two aspects of judging the capability of an intelligent system, human or artificial. There is the *level* of performance, which is often a percentage score on the success on some tasks, such as classification, and the other is the *time* taken to learn. Human-like performance means the system must perform well on both measures. Currently AI systems largely satisfy only the "level" aspect.

One notable example recently is the DeepMind's seemingly successful attempt in using deep reinforcement learning to play Atari games, in which it was purported that the system is a general learning system that is applicable to a wide domain of applications [10]. The claim of generality stems from the fact that the one same algorithm, namely the deep reinforcement learning algorithm, was used quite successfully (in some cases more successfully than others) to play a slew of more than 50 Atari games with a large variety of game scenarios. Their measure of success in playing these games focuses on the "score" measure – i.e., is the system able to score well, at human (novice or expert) level? By that measure of score, they succeeded reasonably well (in more than 50% of the games involved, the system was able to score higher than that of humans), but by the measure of *time*, DeepMind's system plays at a speed many orders of magnitudes slower than that of human players. Tsividis also pointed out this large discrepancy between the time performance of DeepMind's system and that of human players [16].

As for DeepMind's approach and achievement vis-à-vis AlphaGO [15], we observe that a human Go (or for that matter chess) player only has the ability to search a vastly smaller state space compared to that searched by AlphaGo when playing the game, and yet is still able to perform at Grand Master level, despite the recent losses to AlphaGo. Relatively speaking, despite the fact that mechanisms are in place to reduce the search space (e.g., Monte Carlo search mechanisms), AlphaGo still employs basically relatively brute-force search with the aid of super powerful computers. Obviously, the mental processing mechanisms employed by human players on the one hand and AlphaGo on the other are vastly different.

As Go and chess are relatively complicated in terms of the mental processing mechanisms in the human player's mind, which are not fully understood, we believe human players use the learning and understanding of causality to learn how to play Atari games. Ho and Zhu's groups have developed a framework and method to learn causality from visual input [1-8]. We have applied the method to the Atari game, Space Invaders, and have been able to demonstrate that a framework based on learning of causal rules from the visual environment together with an AI problem solving framework can produce a system that achieves human-like performance *both* in terms of level (score) and time taken. Below we first describe the basic principles behind the causal learning and problem solving framework and then describe the application to Space Invaders.

To understand why this is important for AI and robotics, imagine the situations typically faced by a robot if it is to learn and perform tasks on a day to day basis such as operating a toaster or a washing machine. To operate a toaster, a robot is expected to learn that there is a causal connection between the pressing down of the lever on the toaster and the heating up of the toaster, hence the toasting of the bread. If a robot is to function satisfactorily as a house-help, it has to learn this at most within a small handful of demonstrations. Similarly, for a washing machine, a robot is expected to observe the process of loading clothes into a washing machine and then turning it on to wash the clothes, and learning this process perhaps within one or two demonstrations. Similarly, there are many rapid learning scenarios such as these with the various machines present in the offices, factories, disaster areas, and many other places in which a generally adaptive robot is expected to rapidly learn the causalities between actions and effects to be able to function effectively. Thus, human-like performance is indispensible. Deep reinforcement learning is not able to handle this kind of demands.

This paper attempts to use the example of the Atari game, Space Invaders, to firstly illustrate and define what it really means to achieve human-like performance for AI, and secondly demonstrate the mechanisms that enable human-like performance for an AI system.

2 A Causal Learning and Problem Solving Framework

2.1 Basic Idea behind Causal/Temporal Learning

Judea Pearl has recently announced in his book that causality is now a legitimate domain of study [14]. The reason why the study of causality and hence causal learning has been hampered in the past was that there has been an often chanted “mantra” that “correlation is not causality.” However, in the domain of statistics, a method for scientific cause-effect discovery has long been established, and it is basically to uncover the correlation between an intervention (such as the administering of a drug) and an effect (such as the curing of a disease) observed later, with other variables randomized (e.g., age, gender, nationality of the patients involved, etc.) [9, 13, 15]. Statistics takes a conservative stance with regards to correlations without intervention. E.g., if there is correlation in the data that smoking and lung cancer are related, it is not always possible to conclude that smoking causes lung cancer because there could be a gene that predisposes people to smoke and at the same time it predisposes people to develop lung cancer later in life (some tobacco companies have used that to their defense when facing lawsuits). This is the source of the mantra “correlation is not causality.”

Yang and Ho [17] take the stance that both causality and temporal correlations are important for AI’s purposes. If one can establish the correlation between an intervention/action and a subsequent effect, thus establishing the causality between the action and effect, one can use it for (i) prediction – if the action is taken, the effect is expected; and (ii) problem solving – to achieve the effect, one can take the action. On the other hand, if a temporal correlation is observed between two events, none of which is an intervention/action taken by the system/human, then the temporal correlation is useful only for prediction – if the first event is observed, the second event is likely to follow. Thus, whether there is a gene that causes both smoking and the subsequent lung cancer is not important for prediction – if there is indeed a correlation between the two events, then observing someone smoking is sufficient to conclude that lung cancer may follow, notwithstanding whether there is an underlying gene that is ultimately causally responsible for both.

In the following discussion, most of the concepts and constructs are applicable to both causality and temporal correlation as defined above. In some cases where they differ, we will highlight it explicitly, otherwise it can be assumed that the mechanisms work for both situations.

2.2 Diachronic vs Synchronic Causal Condition for Causal/Temporal Rule Learning

In the framework of causal/temporal learning set up by Ho’s group [3, 4, 17], a distinction is made between *diachronic* vs *synchronic* causal/temporal conditions. (As mentioned above, as the following discussion is applicable to both causality and temporal correlation, we will omit the “temporal” portion sometimes henceforth.) Fig. 1 illustrates the distinction with an example.

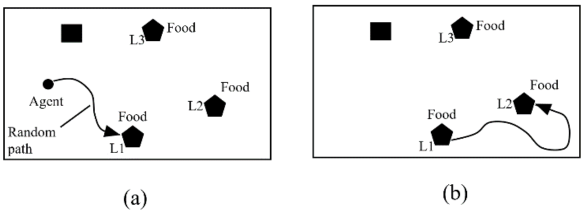


Fig. 1. (a) An Agent wanders around in the environment and Touches a piece of Food and experiences Energy_Increase. (b) The Agent encounters another experience of Energy_Increase.

In Fig. 1(a) it is shown that an Agent explores around in an environment and accidentally Touches a piece of Food at time *T1* and at location *L1* and it finds itself experiencing an increase in energy. A causal rule can be learned as such: *At(Agent, L1, T1) & Touch(Agent, Food, L1, T1)* ➔ *Energy_Increase(Agent, L1, T1).* This is a *specific* causal rule: it says as currently understood, the *Energy_Increase* can only take place if the *Agent Touches* the *Food* at location *L1* and time **T1**.

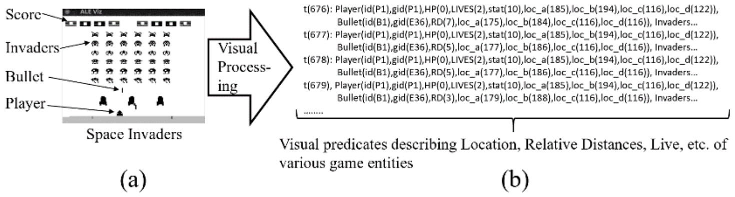
The action part of the causal rule, *Touch(Agent, Food, L1, T1),* is the *diachronic* causal condition. The “background” or “context” part of the rule, *At(Agent, L1, T1),* is the *synchronic* causal condition. The synchronic causal condition is an “enabling” condition – it enables the diachronic preconditional “cause” - *Touch(Agent, Food, L1, T1)* to give rise to the “effect” - *Energy_Increase(Agent, L1, T1).*

Now, suppose the Agent then wanders to another location *L2* and *Touching* another piece of Food at time *T2* and also experiencing an increase in energy. Another specific causal rule can be learned: *At(Agent, L2, T2) & Touch(Agent, Food, L2, T2)* ➔ *Energy_Increase(Agent, L2, T2).* At this stage, a general causal rule can be induced from the two specific instances: *At(Agent, Any L, Any T) & Touch(Agent, Food, Any same L, Any same T)* ➔ *Energy_Increase(Agent, Any same L, Any same T),* meaning that the exact values of location and time are not important for the *Touching* action to give rise to *Energy_Increase* (but they must be the same in the three predicates), but Touching the Food is necessary.

We term this process of unsupervised learning in which the causal rules are picked up, learned, and encoded from observation carried out on the environment the process of *causal learning* (of causal rules). The process is similar for *temporal correlation rule* learning – as we mentioned above, if the first event is not an action effected by the agent (in this case, *Touch*), instead it is just an event observed in the environment, then it would be just a temporal correlation rule useful for prediction but not problem solving (because the first event, not being an action, cannot be effected by the agent). Similar concepts of diachronic vs synchronic preconditions apply to temporal correlation rules.

2.3 The Atari Game Space Invaders

In this section, we describe the basic features of the Space Invaders game. Fig. 2(a) shows a screen shot of the game. Basically, the game consists of a bunch of “Space Invaders coming from the top part of the screen, moving horizontally and then slowly downward, toward a Player at the bottom of the screen. Bullets are fired by the Space Invaders toward the Player. The Player has three actions available to her: move left, move right, or fire a bullet upward to destroy the Space Invaders. Every time an Invader gets hit, the score for the Player will increase, and vice versa. The Player has three lives and when those are expended, the game is over. If the Player can destroy all the Invaders, the game proceeds to the next level. There are barriers lined up near the bottom of the screen between the Invaders and Players that can shield the bullets from either side.


Score
Invaders
Bullet
Player
Space Invaders
(a)
Visual Processing
Visual predicates describing Location, Relative Distances, Live, etc. of various game entities
(b)

```
t(676): Player(id(P1),gid(P1),HP(0),LIVES(2),stat(10),loc_a(185),loc_b(194),loc_c(116),loc_d(122)),
        Bullet(id(B1),gid(E36),RD(7),loc_a(175),loc_b(184),loc_c(116),loc_d(116)), Invaders...
t(677): Player(id(P1),gid(P1),HP(0),LIVES(2),stat(10),loc_a(185),loc_b(194),loc_c(116),loc_d(122)),
        Bullet(id(B1),gid(E36),RD(5),loc_a(177),loc_b(186),loc_c(116),loc_d(116)), Invaders...
t(678): Player(id(P1),gid(P1),HP(0),LIVES(2),stat(10),loc_a(185),loc_b(194),loc_c(116),loc_d(122)),
        Bullet(id(B1),gid(E36),RD(5),loc_a(177),loc_b(186),loc_c(116),loc_d(116)), Invaders...
t(679), Player(id(P1),gid(P1),HP(0),LIVES(2),stat(10),loc_a(185),loc_b(194),loc_c(116),loc_d(122)),
        Bullet(id(B1),gid(E36),RD(3),loc_a(179),loc_b(188),loc_c(116),loc_d(116)), Invaders...
........
```

Fig. 2. (a) The Space Invaders Game; (b) The symbolic predicates extracted from the scene.

In Fig. 2(b), we show the symbolic predicate description of the scene of the Space Invaders game. The data is organized in a temporal form: at each time frame, there is a description of every entity and their associated parameters, much like the predicate description associated with the discussion in Fig. 1. So, for example, the description of a few of the Space Invaders and the Player at time frame *Time(t1)* could be: *Time(t1) - Invader(ID=10, LocationX = x1, LocationY = y1), Invader(ID=11, LocationX=x2, LocationY=y2)… Player(LocationX = x10)….* If there are bullets in the corresponding time frames, they will be included. We encoded a “vision module” to extract this information from every time frame of the Space Invaders game scene generated by the Atari game engine.

2.4 A System for Causal/Temporal Learning, Reasoning, and Problem Solving

In traditional AI, there was a sub-domain of study on problem solving in which there was an attempt to formulate a General Problem Solver (GPS) that has general problem solving mechanisms for all situations, that can act on the facts and knowledge involved in particular domains to derive solutions for particular situations [11]. Our approach is similar here, except that the knowledge, in the form of causal rules, is learned from causal/temporal learning, while in traditional AI research associated with GPS, the knowledge involved was typically hand-coded. Fig. 3(a) shows this basic structure.

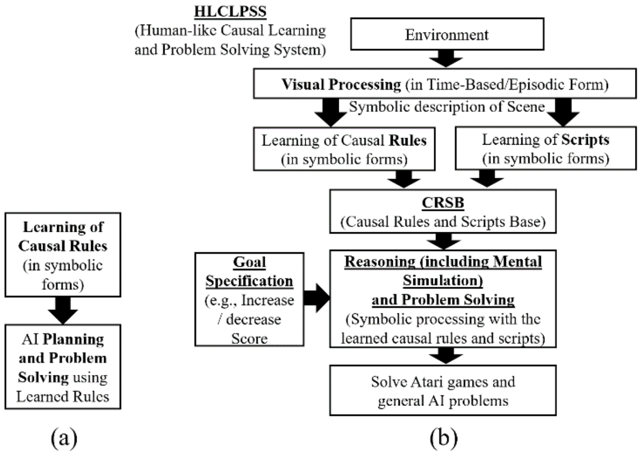

Learning of Causal Rules (in symbolic forms)
AI Planning and Problem Solving using Learned Rules
(a)
HLCLPSS
(Human-like Causal Learning and Problem Solving System)
Environment
Visual Processing (in Time-Based/Episodic Form)
Symbolic description of Scene
Learning of Causal Rules (in symbolic forms)
Learning of Scripts (in symbolic forms)
CRSB
(Causal Rules and Scripts Base)
Goal Specification (e.g., Increase / decrease Score
Reasoning (including Mental Simulation) and Problem Solving
(Symbolic processing with the learned causal rules and scripts)
Solve Atari games and general AI problems
(b)

Fig. 3. (a) The basic overall causal learning and problem solving framework. (b) The detailed processing modules of the Human-like Causal Learning and Problem Solving System (HLCLPSSS).

In Fig. 3(b), we show the further detail on the various modules in the system. The processing begins with the Environment, which for Space Invaders, would be the Space Invaders' visual scene. A Visual Processing module converts that to a time-based, episodic form as described above.

Next, causal rules, much like those discussed in connection with Fig. 1 are learned and encoded. A "clean" causal rule extracted could be something like: *At(Player, Any Location, Any Time) & Contact(Player, Invader_Bullet, Any Same Location, Any Same Time)* ➔ *Destroyed(Player, Any Same Location, Any Same Time + 1).* (In the system we implemented, we used a "causal-noise" filtering mechanism (such as [17]) to remove the "noise" present in the Space Invaders visual scene that may intervene between a "true" cause-effect pair of events, but typically, very "clean" rules – meaning rules that do not have "unwanted" predicates - such as that above are not always easily obtained through the unsupervised causal learning process, but they sufficed for the subsequent problem solving processes that yielded the results that we could use to demonstrate the idea of human-like performance for AI). A small handful of visual predicates, such as Contact, Destroyed, etc. are built-in recognition mechanisms which we assume a typical visual system, natural or artificial, should provide.

In order to facilitate problem solving, the system also learns and encodes Scripts, as shown in Fig. 3(b). Scripts are basically longer chains of actions. E.g., a sequence of movements of a bullet from the starting point to the ending point could be a "Movement

Script”. Currently, in this system, sequences of any 5 actions observed in the environment are stored as Scripts. This vastly cuts down the search space of the problem solving process.

The Problem Solving process of the system basically uses the traditional backward chaining process of AI – given a Goal specification, what are the rules and scripts from the Causal Rules and Script Base (CRSB) that can be assembled to concoct a solution. As will be described below, there are two kinds of Goals – Goal to achieve a desired state (Increase of Score) and Goal to avoid an undesired state (avoid Decrease of Score). As part of the Reasoning process, Mental Simulation is carried out to determine if there is any undesired state – e.g., the movement rules and scripts of the Invader’s bullet can be used to project into the future to see if they might hit the Player, much like a human mentally imagining the sequences of known event changes that lead to a certain consequence.

2.5 Goal-Directed Problem Solving

As mentioned above, there are two kinds of goals - Goal to achieve a desired state (Increase of Score) and Goal to avoid an undesired state (avoid Decrease of Score). These are described as follows:

Goal to Achieve a Desired State and the Associated Learning Process

Fig. 4 illustrates a typical situation in Space Invaders in which there is a desired Goal to achieve.

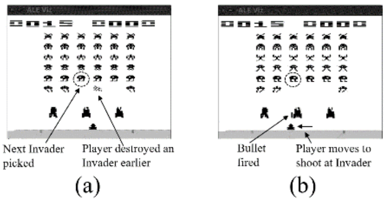

Next Invader picked
Player destroyed an Invader earlier
(a)
Bullet fired
Player moves to shoot at Invader
(b)

Fig. 4. To achieve a desired goal in Space Invaders: Player finds a solution to destroy an Invader. (a) Player is not in the correct position to destroy the Invader. (b) Player moves to the correct position as directed by the solution of a problem solving process.

In Fig. 4(a), it is shown that the Player is not in a position to fire a bullet to destroy an Invader. The Player carries out a backward chained problem solving process and obtains a solution – moves to a location at which the Invader is in the direct line of fire and fires a bullet to destroy the Invader.

The learning process proceeds as follows. In an initial “exploration phase” (much like the exploration phase of reinforcement learning), the Player fires at random, and occasionally a bullet would hit an Invader and destroy the Invader. After a few instances of similar experience, a causal rule such as this is learned: *At(Player_Bullet, Any Location, Any Time) & Contact(Player_Bullet, Invader(Any ID), Any Same Location, Any*

Same Time) ➔ *Destroyed(Invader(Any Same ID), Any Same Location, Any Same Time + 1)*. (As mentioned above, in our implemented system, the learned rules may not look as "clean", as there are other "noisy" diachronic and synchronic conditions that "creep" into the rule, but they suffice for problem solving purposes and this clean rule is good for illustrating the basic idea.). Subsequently, the Score goes up, and that is a desired Goal. In principle, the system should also learn the causal rule *Destroyed(Invader, Any Location, Any Time)* ➔ *Increase_Score(Any Time + 1).* (Our system can handle some degree of intervening noise between two causally or temporally linked events and learn the causality or temporal correlation [17], but the current Space Invaders environment proves too noisy to achieve this, so in our current implementation we use *Destroy(Invader(Any ID), Any Location, Any Time)* as our "built-in" Goal.)

After all these causal rules have been learned, the system is ready to carry our backward chained problem solving. At all times, the system is in the mode of looking for ways to achieve a desired Goal, and in this case, it would be Increase_Score or *Destroyed(Invader(Any ID, Any Location, Any Time).* So, the above rules are then used in a backward chained process – in order to achieve destruction of an (any) Invader, a *Player_Bullet* must be made in *Contact* with it. In order for a *Player_Bullet* to be made in *Contact* with it, the *Player_Bullet* must be fired from a certain position (encoded in the scripts learned earlier in an unsupervised learning process), in order for the *Player_Bullet* to be fired at a certain position the Player must *Move* to a certain location, etc. Thus, the solution shown in Fig. 4(b) is obtained. The Invader target can be selected at random or the nearest one to the Player is selected.

The above process closely simulates the problem solving processes carried out by a human player rapidly in her mind about what to do to achieve the Goal of increasing the Score, and also simulates the rapid learning process for a human player to reach some decent level of score performance.

Goal to Avoid an Undesired State and the Associated Learning Process

Fig. 5 illustrate a typical situation in Space Invaders in which there is an undesired Goal to avoid.

In Fig. 5(a), it is shown that an Invader fires a bullet at the Player. Using *mental simulation* based on the earlier learned, known rules of the bullet's behavior, the system knows that some time in the future the bullet will hit the Player (because it is in the bullet's path). The system therefore concocts a plan to *prevent* this from happening. The solution is to move left a little bit as shown in Fig. 5(b). In Fig. 5(c) it is shown that the bullet is successfully avoided.

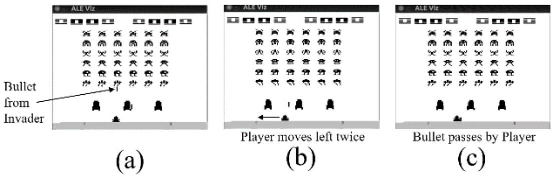


Fig. 5. Player avoids an undesired Goal of being destroyed by a bullet from an Invader. (a) A bullet is fired from an Invader toward the Player, and in mental simulation, the system knows

that the Player will get hit. (b) The system concocts a plan after some problem solving process to avoid the bullet. (c) The bullet is avoided.

This learning of this process capitalizes on the idea of contrapositivity in logic. Consider the following logical statement:

A and B and C and D… ➔ *E* (1)

The contrapositive of the above is:

Not E ➔ *Not A or Not B or Not C or Not D…* (2)

This contrapositive reasoning process is built into the system and used in the process of reasoning out a way to avoid an unnecessary goal. For example, in the beginning of the Space Invaders game, the situation of experiencing the undesired goal is first learned in a few instances in which the Invaders fire bullets and they destroy the Player (this is not random – the game engine deliberately does that). The entire script containing a sequence of events of a bullet appearing, moving step by step toward the Player, contacting the Player, and then the Player gets destroyed is learned:

Appear(Bullet, Loc1) ***and*** *Move(Bullet, Loc2)* ***and*** *Move(Bullet, Loc3)* ***and****…*
Contact(Bullet, Player, Loc10) ➔ *Destroyed(Player, Loc10)* (3)

This sequence is a *conjunction* of a series of events that must happen for the player to be destroyed. (After the system has encountered more instances of (3), a generalized version with "any location" in the location parameters will be learned.)

Apply the contrapositive reasoning process, this is converted into:

Not(Destroyed(Player, Loc10) ➔
Not(Appear(Bull Appear(Bullet, Loc1) ***or***
Not(Move(Bullet, Loc2) ***or***
Not(Move(Bullet, Loc3) ***or***
Not(Contact(Bullet, Player, Loc10))

Which means that *any* of the actions taken to negate the original events in the sequence is sufficient to achieve a *negation* of *Destroyed(Player, Loc10),* which is the desired Goal of avoiding an undesired state.

The system then queries its Causal Rule and Script Base (CRSB) in Fig. 3 to see if there is any solution to effect at least one of the negations. If the system can make the Bullet *not* Appear, at Loc1, where it Appears in the shooting situation, that will lead to the *non-destruction* of the Player. Or, if the system can Stop the Movement *(Not(Move))* of one of the Bullets in any one of the locations Loc2, Loc3, etc., it will also be able to prevent the destruction of the Player (it is like a super-hero being able to stop a bullet in mid-flight). The last choice is to achieve a *Not(Contact(Bullet, Player, Loc10)).* It turns out for the Space Invaders game, the only action available is to move the Player right or left, or to fire a bullet from the Player, none of which actions can immediately achieve any of the above negations. When the system encounters this situation of no available solutions from the CRSB, it will nevertheless try to emit *any* action at its disposal (this is built-in as a general *"try anything to see if there is a solution"* procedure). It turns out that in this case by randomly emitting the left and right movements of the Player, a *Not(Contact(Bullet, Player, Loc10))* can be achieved. This involves a search process with a small state space.

Therefore, in general, causal learning, unlike the traditional AI search processes, when used in problem solving, can arrive at problem solutions very rapidly. Though there may still be some search processes involved like in our example above, the search space is miniscule compared to those typically effected by traditional AI search processes, such as the A* search process.

3 Results of Human-like Performance Space Invaders Game Playing System

The various causal learning, reasoning, and problem solving processes, including internal mental simulations processes have been implemented and tested on the Space Invaders game. The results are shown in Fig. 6.

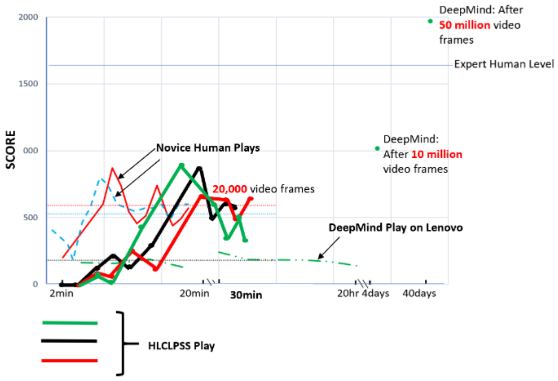


Fig. 6. The results of the Human-like Causal Learning and Problem Solving System (HCLPSS) applied to the Space Invaders game, along with DeepMind's results (as reported in their paper [10]) for comparison.

In Fig. 6, the results from 3 trials, starting from the beginning of a typical Space Invaders game, are shown along with (i) human novice performance; and (ii) DeepMind's deep reinforcement learning results, as reported in their paper [10], time-scaled to assuming that it was run on the same computer as ours, based on the total number of video frames needed before certain performance is achieved. We also executed DeepMind's publicly accessible code to obtain its performance in the early part of the game.

It can be seen that our system's speed is close to the order of magnitude of that of the human novice. Compared with human novice though, we began with no prior knowledge of the game. Humans typically would have some fundamental prior knowledge relevant to the game when they begin to play it, thus learning at an even faster rate – typically in fewer than a few minutes.

Compared to DeepMind's deep reinforcement learning though, the system's speed of learning is obviously faster by orders of magnitude.

4 Conclusion

In this paper, we first define what we mean by human-like performance AI, which is a system that is not only able to achieve human performance in terms of "level" or "score" (like the percentage accuracy in classification or the game score in a computer game), but must also achieve the level or score in reasonably short, human-like time frame. Then we describe a causal learning and problem solving framework to demonstrate how, when applied to an Atari game Space Invaders, it is able to achieve human-like performance accordingly – achieving human-like game score in human-like time frame. The key idea behind the framework is the learning of the causalities taking place between different events, with "true understanding." Explainability is an inherent property of the system right from the beginning.

We believe the general framework of causal learning and problem solving as depicted in Fig. 3 and demonstrated in this paper using Space Invaders has a general applicability to the situations that could be encountered by an AI or robotic system discussed in the Introduction section.

What have demonstrated in this paper is the ability of the system to reach human-like performance at the human novice level. We are currently continuing to enrich the basic framework of Fig. 3 to allow the system to reach human expert level performance.

Future research will apply the basic system to more Atari games to further explore some fundamental issues, as well as to apply the basic causal learning and problem solving framework to real world robotic situations.

References

1. Fire, A., Zhu, S.-C.: Learning perceptual causality from video. ACM Transactions on Intelligent Systems and Technology, 7(2), 23 (2012).
2. Fire, A., Zhu, S.-C,: Inferring Hidden Statuses and Actions in Video by Causal Reasoning, In: IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 9-16. IEEE Press, Piscataway, New Jersey (2017).
3. Ho, S.-B.: On effective causal learning. In: Proceedings of the 7th International Conference on Artificial General Intelligence, pp. 43-52. Springer-Verlag, Berlin. (2014).
4. Ho, S.-B.: Principles of Noology: Toward a Theory and Science of Intelligence. Springer International Publishing, Switzerland (2016).
5. Ho, S.-B.: Deep thinking and quick learning for viable AI. In: Proceedings of the Future Technologies Conference, pp. 156-164. IEEE Press, Piscataway, New Jersey (2016).

6. Ho, S.-B.: The Role of Synchronic Causal Conditions in Visual Knowledge Learning. In: IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 9-16, IEEE Press. Piscataway, New Jersey (2017).
7. Ho, S.-B., Liausvia, F.: A Ground Level Causal Learning Algorithm. In: Proceedings of the IEEE Symposium Series on Computational Intelligence for Human-like Intelligence, pp. 110-117. IEEE Press, Piscataway, New Jersey (2016).
8. Ho, S.-B., Liausvia, F.: On Inductive Learning of Causal Knowledge for Problem Solving. In: Technical Reports of the Workshops of the 31st AAAI Conference on Artificial Intelligence, pp. 735-742. AAAI, Palo Alto, California (2017).
9. Ho, S.-B., Edmonds, M., S.-C., Zhu (2018). Actional-Perceptual Causality: Concepts and Inductive Learning for AI and Robotics. Presented at the Workshop on Perspectives on Robot Learning: Imitation and Causality, Robotics: Science and Systems Conference, Carnegie Mellon University, Pittsburgh, Pennsylvania (2018).
10. Mnih, V., et al.: Human-level Control through Deep Reinforcement Learning. Nature 518, 529-533 (2015).
11. Newell, A., Shaw, J.C., Simon, H.A.: Report on a general problem-solving program. In: Proceedings of the International Conference on Information Processing, pp. 256–264. (1959).
12. Neyman, J.: Sur les applications de la theorie des probabilites aux experiences agricoles: Essai des principes. Master's Thesis. Excerpts reprinted in English, Statistical Science, 5, pp. 463–472. (D. M. Dabrowska, and T. P. Speed, Translators.) (1923).
13. Pearl, J.: The Book of Why: The New Science of Cause and Effect. Basic Books: New York (2018).
14. Rubin, D.: Causal Inference Using Potential Outcomees. Journal of American Statistical Association, 100(496), 322-321. doi:10.1198/016214504000001880. (2005).
15. Silver, D. et al.: Mastering the game of Go with deep neural networks and tree search. Nature 529(28), 484-489. doi:10.1038/nature16961 (2016).
16. Tsividis, P. A., Pouncy, T., Xu, J. L., Tenenbaum, J. B., Gershman, S. J.: Human learning in Atari. In: AAAI Spring Symposium Technical Report, pp. 643-646. AAAI: Palo Alto, California (2017).
17. Yang, X., Ho, S.-B.: Learning correlations and causality through an inductive bootstrapping process. In: IEEE Symposium on Computational Intelligence, IEEE Press, Piscataway, New Jersey (2018).