

Agent mood training in the Unsupervised Environment Design framework

Antoine Khoury

Supervisors : Pr. Jun Wang, Pr. Argyris Stringaris

*A dissertation submitted in partial fulfillment
of the requirements for the degree of
Master of Science
of
Machine Learning.*

*Department of Computer Science
University College London
September 2022*

*This report is submitted as part requirement for the MSc Degree in Machine Learning/
CSML at University College London. It is substantially the result of my own
work except where explicitly indicated in the text.*

*I, Antoine Khoury , confirm that the work presented in this thesis is my own.
Where information has been derived from other sources, I confirm that this has been
indicated in the work.*

*Code for this project can be found in the following public repository:
<https://github.com/AntoineKhoury/UED-for-Mood-Training>
The report may be freely copied and distributed provided the source is explicitly
acknowledged.*

Abstract

Current machine learning algorithms make use of evolutionary concepts to improve their strategies in many games. The concept of using human behavior to solve complex tasks in reinforcement learning isn't new. But it is also interesting to see how multi-agent setting can help us learn and create new solutions to affect human behavior.

In the field of psychology, more and more research is done to put mathematical frameworks around the concepts of "depression" and "anxiety".

The aim of this report is to draw parallels between human behavior and multi-agent machine learning and see how multi-agent settings can be used to affect the behavior of an agent.

In this report is proposed the concept of reading Q-Values in Deep Q Networks as representations of the "anxiety" and "depression" of an agent.

The results observed didn't permit to establish any firm utility to the usage of Unsupervised Environment Design for mood training of the agents.

Acknowledgements

I would like to thank Pr. Jun for the time he dedicated furthering my knowledge on multi-agent settings and accepting to supervise my project.

I would also like to thank Pr. Argyris Stringaris for introducing me to the topic of Psychology and how it could benefit from the field of Machine Learning.

Contents

1. Introduction and Key results
2. Literature Review
 - 2.1. Multi-agent Machine learning
 - 2.1.1. Main concepts of Game Theory
 - 2.1.2. Current state of multi-agent machine learning
 - 2.1.3. Reinforcement Learning in game settings
 - 2.2. Passing information between agents
 - 2.3. Transfer Learning
 - 2.4. Understanding emotions and their roles in decisions processes
 - 2.5. Implementation of emotions in Machine learning
 - 2.6. Unsupervised Environment Design
3. Contribution
 - 3.1. Motivation of the project
 - 3.2. How to quantify depression and anxiety in an agent?
 - 3.3. Setup of the experiment
 - 3.4. Setup of the experiment
4. Experimental Results
 - 4.1. Learning Stages of the different Methods
 - 4.1.1. Method 1
 - 4.1.2. Method 2
 - 4.1.3. Method 3 and 4
 - 4.2. Associated Q-Values for each method in the different situations
5. Discussion of the results
6. Conclusion
7. Bibliography

1. Introduction and Key results

This project aims to work on establishing parallels between parameters observed in machine learning agents and human behavior.

4 different methods were used to see if Unsupervised Environment Design could be used in Mood training (training the agent so that its Q-values display signs of better behavior).

The Q-values of agents are proposed to be red as indications of “depression” and anxiety “levels”.

The results obtained didn’t permit to establish any interest in using UED for mood training of agents, but raised some interesting questions on how to do so.

2. Literature Review

2.1. Multi-agent Machine learning

2.1.1. Mains concepts of Game theory

The field of Machine Learning has many different branches all tackling different problems. But the concept of Multi-agent machine learning takes its origins in the field of Game theory. This field models situations in which multiple agents “play” a certain game and try to maximize their personal outcomes. The origins of that problem arose as the concept of Game Theory, from the field of Economics. Three main authors seem to have established the origins of that problem. “The Recherches” by Cournot Antoine Augustin was published in 1838 and presented the concepts of multiple game settings. He was first to introduce the concept of “Equilibrium” in the problem of production maximization of firms. Edgeworth, Francis published in 1881 “Mathematical Psychics” in which he demonstrated the concept of “competitive equilibrium”. “Communication on the Borel Notes”, J. von Neumann and M. Fréchet, *Econometrica* 1953 stated that the origin of the concept of Mixed strategy had to be attributed to Emile Borel whom wrote it on a note.

One of the major concepts of game theory is also the Nash equilibrium. It was introduced by John F. Nash Jr. in 1950 in the paper “Equilibrium points in n -person games”

It states that in a multi-player game, each agent must:

- Correctly predict the decision of the other agents

- Maximize its gain knowing these future decisions.

No player can regret it's choice seeing what the other player finally did. This concept also incorporates the concept of "rationality". However assuming a rational behavior might not necessarily lead to an optimal outcome [Rationality and Game Theory].

To best model unrational behaviors, "Modelling Bounded Rationality in Multi-Agent Interactions by Generalized Recursive Reasoning" provides a framework for agents of different level of rationality can interact together, permitting the most rational agent to optimize its behavior knowing other agents will display less rational behaviors.

In our topic, it is mainly important to understand the concept of "power" in games. In fact, agents might display different levels of power that might results in specific dynamics among them. In the paper "Power in game theory", it is explained why such differences in power levels arise. It is due to assymetries amongs the players. This paper proposes a framework to evaluate these differences and hence properly quantify how they impact the players behavior. This paper assumes rational behavior among all players.

Furthermore

2.1.2. Current state of multi-agent machine learning

Thanks to the great improvement of ressources available for mathematical computation and simulation, the field of Game Theory has been able to tackle more complex games and to simulate solutions for them. Multi-agent Machine Learning take the same concept of agents playing a game, each one of them trying to maximize their personal reward.

In their paper "Autocurricula and the Emergence of Innovation from Social Interaction: A Manifesto for Multi-Agent Intelligence Research", Joel Z. Leibo, Edward Hughes, Marc Lanctot and Thore Graepel present that the concept of innovation arises from the need to adapt of agents in a system that faced perturbation. This parallel is very interesting as it describes in which limits the concept of Multi-agent machine learning can be used to explain behaviors observed in the real world. The concept of Autocurricula is presented in this paper. It states that each new adaptation of the system on all of its levels will create new social challenges, to which the curricula will create an adaptation. Since the system generates those new challenges by itself, it was hence named Autocurricula. This concept is particularly interesting to our case as it permits to imply that human behaviors in real world scenarios can be seen as another problem in which the human (part of the system) will try to find solution.

When Reinforcement learning algorithms try to optimize their policies, they rely on the concept of Trust Region. If the policy hasn't reached maximum reward in the problem, there exist a better policy in the nearby "region" which can be attained by adapting the weights. The measure of how impactful this change is can be also computed. However in multi-agent settings this is much harder to simulate. Indeed, each agent will try to adapt its policy assuming other agents won't change theirs. To overcome this problem, "MULTI-AGENT TRUST REGION LEARNING" develops a method to generalize the concept of trust region to multi-agent settings, ie. Permitting a better optimization in the learning phase.

When it comes to equilibrium in multi-agent settings, it is not necessarily attained and hence some behaviors failed to be explained by the only concept of Nash Equilibrium. Another method that fits empirical data is the concept of "Cognitive Hierarchy", introduced in "A COGNITIVE HIERARCHY MODEL OF GAMES" which assumes each agent will expect the other agents response to be distributed over all the previous episodes faced in the game. This simpler method seems to model the behavior observed in many different games.

When we trying to make the most optimal decision, agents expect behaviors from the other agents. However regular multi-agent settings don't account for the fact that real world agents will learn adapt their behavior to account for the responses of other agents to their actions. This recursive reasoning best illustrates how humans take decision with their peers. This concept was introduced in "PROBABILISTIC RECURSIVE REASONING FOR MULTI-AGENT REINFORCEMENT LEARNING".

2.1.3. Reinforcement Learning in game settings

To optimize their "policies" (often denoted π), they use concepts from the field of Reinforcement Learning. Two main areas separate how these algorithms operate: Model Free and Model based Reinforcement Learning. Model Free models will try to estimate the expected reward of an action given a state the agent is in. This method is highly dependant on what was observed in the game. In contrary, model based RL will learn a representation of the problem and immediate outcomes through plays. Given that internal representation, it will then adjust its policy [Reinforcement learning: The Good, The Bad and The Ugly, 2008].

When it comes to how the policy is learned, different algorithms come in to play.

Deep Q-Networks. This method was used by the teams of Deepmind on Atari games. The agent was inputed pixels and derived which action was the best in that situation. The architecture of the game wasn't changed for the games, and it gave promising results. The

DQN even surpassed the human experts on 3 of the games, and surpassed any other method on 6 other games [Playing Atari with Deep Reinforcement Learning].

In a DQN algorithm, the goal is to maximize the cumulative reward:

$$R_{t_0} = \sum_{t=t_0}^{\infty} \gamma^{t-t_0} r_t \quad (1)$$

The discount factor γ is a constant between 0 and 1 which assigns an importance value to each step reward observed. The oldest events being the ones with lowest importance reward.

To maximize that cumulative reward, this algorithm uses the Q-values function which updates as following:

$$Q^{\pi}(s, a) = r + \gamma Q^{\pi}(s', \pi(s')) \quad (2)$$

At the beginning of the algorithm, the Q-values are initiated with the same value.

After each step, The Q-values are updated according to the new relationship between the current observed state s , the action chosen a , and the observed reward at the end of the sequence.

Hence, the Q-value function is used to estimate the reward given a certain action and state. When playing, the agent will then chose the action associated with the maximum Q-value.

The implementation of the Deep Q-Values Learning is simply the use of Deep Convolutional Neural Networks to estimate the Q-values associated with each action-state pair.

Another method is Proximal Policy Optimisation. Because of its structure, it permits to train agents on mini-batches of data, and is seen as a generic method that displays a good balance between “sample complexity, simplicity, and wall-time” [Proximal Policy Optimization Algorithms]

The concept of Deep Neural networks mimics the architecture of human brains as it links a network of neurons to transform an input into information that can be used by the algorithm. When trying to improve the structure of these algorithms, it is important that a single architecture can solve many games at once. The IMPALA architecture introduced in

[IMPALA: Scalable Distributed Deep-RL with Importance Weighted Actor-Learner Architectures] aims to solve this problem, by increasing the data efficiency in multi-task learning.

2.2. Passing informations between agents

When multiple agents tackle a certain problem, they don't usually share information one with another. In fact, they will implicitly learn the behavior of the other. Most multi-agent problems in machine learning consider the other agents to be part of the environnement, and might not even be aware of the existence of these other agents.

But it is crucial for some situations that agents share information with one another. In fact, many problems will involve partially observable informations and hence require agents to communicate together. The paper "Real World Multi-agent Systems: Information Sharing, Coordination and Planning" addresses this topic . It focuses on the fact that each agent only observes a part of the world, and it tries to combine those observations in the form of Distributed Perception Networks. This is then used to plan efficient coordination. This method can also prove useful for problems where all agents observe the same things. In fact, such behavior can easily be observed in real world problems where humans communicate before making actions in a specific setting, which is much more reliable than taking actions at first and estimate new actions later on.

Furthermore, when it comes to estimate intrinsic behaviors of agents, it is important for each agent to estimate the others. To address this topic, the thesis « Modeling Mutual Influence in Multi-Agent Reinforcement Learning » by Ying Wen develops on the influence agents have on one another. Our case seems to be parallel to this idea in the sense that agents directly share sentiments from one to another, and hence don't try to know what the policy of the other agents are.

2.3. Transfer Learning

The field of Transfer Learning studies how learning a certain problem can prove useful on an other set of problems. It has applications on many different areas, such as NLP where transformers are a direct implication of this concept [BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding]. The interest of training a machine

learning algorithm on a certain problem to easily solve another one arises because of the lack of resources. It can be a small number of the training set, a problem being of too high dimensions to be addressed abruptly etc. In our case, it is of interest as it permits to try to solve a complex task by teaching the agent to solve another case.

The concept of transfer learning has already been studied in the case of psychology, such as in the study “Transfer in children's maze learning » where researches from the University of California Institute of Child Welfare studied how playing a real life sized maze helped young students solve a stylus maze showing similar patterns. They concluded that the behavior observed was mechanical and learning in an order or another didn't show any important difference.

2.4. Understanding emotions and their roles in decisions processes

Now that we have tackled the machine learning topics, it is important to address the state of the art of the field of psychology when it comes to emotions and their roles in behaviors.

First of all, an emotion is defined in “A Formal Valuation Framework for Emotions and Their Control”. This paper talks about putting valence scores on emotions, to later be able to assess them. In fact, it states that emotions are “limited, relatively fixed, number of universal,” “evolutionarily shaped” emotions.

What can be understood from this is that emotions show similarity with the way machine learning addresses behavioral sciences, as quantities that can be measured and vary over time due to the agents contact with the environment.

This accounts for the fact that we could use a single “emotion” from an agent and let it evolve with training in the multi-agent setup.

“The Evolutionary Origins of Mood and Its Disorders” by Daniel Nettle and Melissa Bateson. Definition of mood: The term ‘mood’ in its scientific usage refers to relatively enduring affective states that arise when negative or positive experience in one context or time period alters the individual's threshold for responding to potentially negative or positive events in subsequent contexts or time periods.

This paper explains why moods exist. It counts on the continuity of environment which permits to infer from negative events. In fact, if the environment from a single step from another changed completely, it would be impossible for agents to make use of previous experiences.

It also addresses how the differences between individuals induce different response levels. In fact, an agent with lower physical capabilities will not evaluate risk the same way as a strong agent, and hence will not for example compete for certain resources, again affecting its physical state. But the differences are not necessarily physical, they can also be cognitive differences. In fact, there is no proof that living individuals perceive their environment the same way, hence highly influencing their individual decisions. It is that correlation between the environment and the individuals that dictate their curricula and decisions to survive in the environment.

This paper also presents a very interesting mapping of anxiety and depression in a 2D plane.

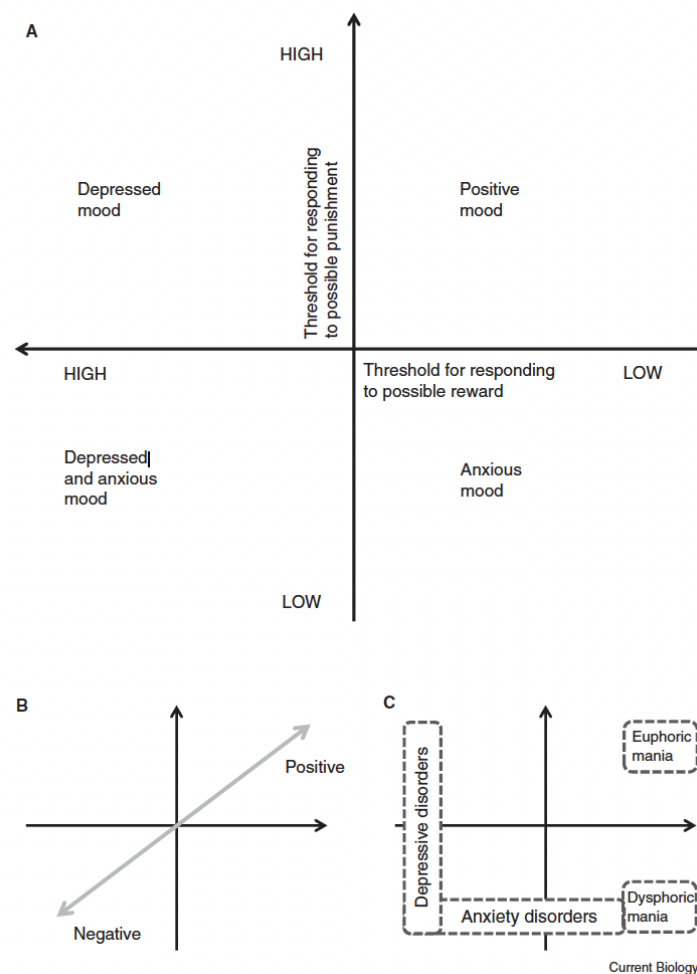


Figure 1. From this figure, it can be observed that anxiety and depression represent the two major components that affect mood at a high level. We can observe that a depressed mood is the result of an individual not being motivated by low rewards actions (an easy parallel can be done with daily actions that don't result immediate important reward but rather low but important over time, ie brushing your teeth). Furthermore, anxiety is described as a behavior observed in an individual who is resilient when facing low levels of pain. This behavior is

studied in this project as it aims to affect that resilience when facing losses but continuing to try to solve the problem.

When it comes to how mood is affected by events, it is important to consider the temporal relationship between experiences and the current mood of an agent. The paper “The temporal representation of experience in subjective mood” studies that behavior and states the mood can be explained by a primacy model. This type of representation was introduced in the paper “The primacy model: a new model of immediate serial recall”. It states that informations are stored in order and that the importance of action of old events in the list decreases to form a primacy gradient.

Hence, mood should be affected by recent informations, just as behavior of machine learning agents is mostly affected by recent events. If an agent was trained in a certain task and further trained on another task, the data it would have first observed would not fit the model as close anymore, as it would have started learning from the new set. This is mostly useful when learning a specific task derived from a general task. This concept proves very useful and is used in most transfer learning tasks. Here, we can observe a clear similarity between the implementation of agents and how they show direct similarity with machine learning agents.

When learning how emotions play a role in our decision processes, it is important to learn how humans learn of their experiences. The concept of meta-reasoning humans display in their decisions is addressed in the paper “A Formal Valuation Framework for Emotions and Their Control”. This paper proposes a framework to quantify emotions and see how they dictate certain behaviors to agents.

They also develop on the concept of pruning, i.e. Not following a certain decision path as it displays high losses since the early steps. Even if that path might result in higher late rewards, humans will display that behavior. Being aware of that concept permits to show the relationship of immediate emotions (fear of loss) impact decision solely based on the immediate reward. This can be observed in the Figure 2. Shown below.

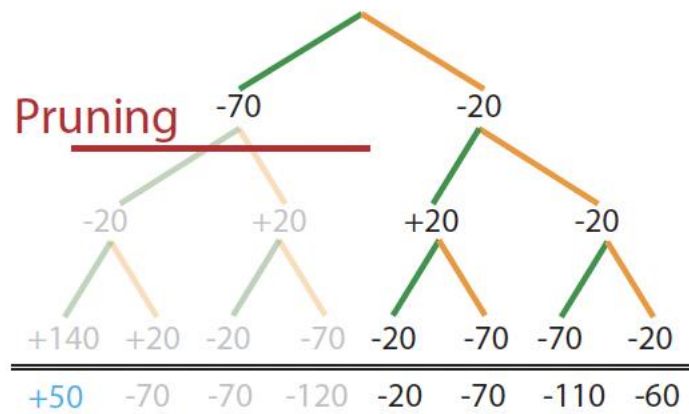


Figure 2. Understanding the concept of pruning and how it can affect decision processes in their early states.

It is important to note that the concept of pruning can also be observed in Reinforcement Learning algorithms. In fact, when trying to optimize to a certain task, an agent might face high losses as part of the game, and might fail to overcome this loss to find the solution to the problem. It is that specific task that was tackled in this project, trying to overcome a punctual loss to reach the final goal by overcoming this pruning behavior.

By not considering all the ranges of values, agents limit their reward potential, hence trading off immediate reward with long term reward. An interesting parallel can be done with the concept of anxiety explained earlier. In fact, anxiety being a response to recent exposure to loss, it might enforce a pruning behavior on an individual playing a game. But it is important to be aware of that behavior to be able to reassess that behavior if it doesn't prove to be useful for the agent.

In most problems, the computing resources are too demanding to simulate the outcomes of a sequence of actions. This is how this paper describes the role of emotions: implementing a metareasoning strategy. According to that paper, there are multiple meta reasoning mechanisms that happen in decision taking processes. The first one aligns with current Reinforcement Learning techniques to train agents:

“The second source of control could be, confusingly, model- free. Performers may learn from experience that a certain amount of catastrophizing improves their performance.”

The fourth one is particularly interesting in our case:

“The fourth evaluative process, again confusingly, could be model-based, where the precise consequences of particular emotions are examined and evaluated. This is rarely feasible and probably only commonly done in situational analyses in psychotherapy, where emotions, thoughts, behavior, and consequences are explicitly discussed. This allows patients to learn to consciously and explicitly assess whether a particular emotion is appropriate and helpful in a given situation, and to adapt it by using reappraisal and other emotion regulation strategies if necessary.”

In our case, we would have access to those information about a specific sentiment to be able to adapt it over training, and assess how it would be used to take actions.

When it comes to assessing good or bad behaviors to then learn from them, the differences of feelings that can be observed can't be solely explained by a simple valence explaining them. In fact, the origins of mixed feelings is described in “Levels of valence” as a component of “micro-valences” such as “appraisals of (un)pleasantness, goal obstructiveness/conduciveness, low or high power, self-(in)congruence, and moral badness/goodness ». These micro-valences can then be combined to form a macro-valence, model that fits most of the psychological representations as a “common currency” among models.

In multi agent settings, it is important to understand the concept of dominance. In fact, some agents might display dominant behaviors due to the nature of the game, the resources at their disposition or other factors. The concept of power game is a concept discussed earlier in the game theory section, but it has direct implications in psychology.

In most psychological fields, it is important to make a distinction between mood and emotion. In fact, they both interfere together but mood is what makes individuals make decisions based on past experience without clear idea of why it's the right decision. In fact, mood is a cognitive process that drives people in their daily actions. When actions have to be taken, it is impossible for a human being to know recall all its previous experiences and pick its decision while being conscious of what he previously faced. This is where the concept of

mood comes into play. Humans don't need to recall those experiences, but in contrary will follow an inner understanding of the problem to find a solution.

The paper “The Evolutionary Origins of Mood Review and Its Disorders” describes the concept of mood as “relatively enduring affective states that arise when negative or positive experience in one context or time period alters the individual's threshold for responding to potentially negative or positive events in subsequent contexts or time periods.”

Now to understand why studying emotions is of utmost importance, the paper “THE SCIENCE OF EMOTION: Exploring the Basics of Emotional Psychology » explains the importance of emotions in decision process. In fact, it is stated that “emotions are fundamentally constructive. They are influenced by what is good for our species overall and what we learned during our upbringing ». In fact, the role of emotions in decision processes is clear but still lacks explanations and parallel with how machine learning is currently being implemented.

2.5. Implementation of emotions in Machine learning

Now that we have studied how emotions can be described and explained in humans, it is important to see how this concept has been applied to machine learning, to develop artificial intelligences that mimic human behavior, ie. Reacting to situations based on their feelings or mood.

The idea of utilizing machine learning to understand human behavior is not a new concept. In fact, some studies have been trying to rely on those algorithms to copy certain stages of the human development.

The paper “BABYAI: A PLATFORM TO STUDY THE SAMPLE EFFICIENCY OF GROUNDED LANGUAGE LEARNING” studied that behavior by proposing problem sets that described different language learning stages of babies. This study also proposed an agent simulating a “human teacher” of the human language. This studies concludes that the resources given by current machine learning methods don't provide sample-efficient algorithms. In fact, babies can learn from very few examples whereas machine learning algorithms still require important amounts of data to learn intrinsic rules in the human language.

The sentiment concept is mainly used in Natural Language Processing, for which a sentence is analyzed by a machine learning algorithm which returns a score (in regression setup) or class (in classification setup) representing the sentiment of the person whom either wrote or is reading it.

The problem this project is trying to tackle is mood improvement of humans by playing games reacting to the humans mood and attitude. However, when we train algorithms with specific humans, it was proven that the model will not generalize well to other people. In the paper “Collaborating with Humans without Human Data”, the agents was trained in the setup of “fictitious play”. This method permitted generating larger sets of data without having observed those data.

This can prove highly useful in our setup to permit to generalize our solution to more humans, so that their intrinsic differences regarding mood training can be leveled, and hence expect the same results.

In the context of Reinforcement learning, emotions can be interpreted from behaviors observed in the agents.

The survey “Emotion in reinforcement learning agents and robots: a survey” presents how this psychological concept can be interpreted in many different frameworks.

According to this paper, different types of emotions valences are “extrinsic/homeostatic, intrinsic/appraisal, value function and reward-based, and finally hardwired”.

In our setup, we only consider the case of reward-based emotions, as they are the ones that have to be tackled to improve the reward and behavior of an agent in a certain setup.

2.6. Unsupervised Environment Design

A major concept in multi-agent settings that can help solve our problem is the solution proposed by Unsupervised Environment Design.

This field of machine learning studies reinforcement learning of single or multiple-agent games where certain variables of the game aren’t hard encoded by the player.

This field is of major importance because it permits to generate new data to train the agents on without having to hard code them humanly.

A simple method to generate new data is Domain Randomization. This method was used in “Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World”. This method proved to be useful to train an object localization task model with a precision of 1.5 cm with data completely generated artificially.

Hence, when it comes to data augmentation, Domain randomization can prove to be highly effective.

However, purely randomizing variables in a game setup might not lead to games of increasing complexity. In fact, in a maze with complex architecture, many angles, many dead ends can’t be generated with pure domain randomization. To this end, another solution was developed in “Emergent Complexity and Zero-shot Transfer via Unsupervised Environment Design”.

This paper tackles the problem of zero shot learning, where an agent is trained to solve mazes and is tested on a completely new one.

Because maze can showcase many different levels of complexity, they represent an easy game to train agents on, on which their efficiency can be easily tested.

The concept of Unsupervised Environment design in this setup means that walls can be placed randomly across the maze, while making sure the maze is doable.

Previous solutions failed to train agents gradually if the maze builder (first agent of the game) and the player (second agent) were put in direct conflict. Such method is called “minimax adversary” since the maze builder is rewarded when the maze player loses and punished the the maze player finishes the game.

In this paper, they present a solution called Paired: Protagonist Antagonist Induced Regret Environment Design. In this setup, another agent is added to the game and tries to solve the maze.

Now, the maze builders goal is to make the second maze player find a solution while the first one can’t solve it. Hence, if the maze created is too hard for both players, the maze builder isn’t rewarded and will go for a simpler solution, while still increasing in difficulty.

This creates an autocurricula as mentioned previously in this report.

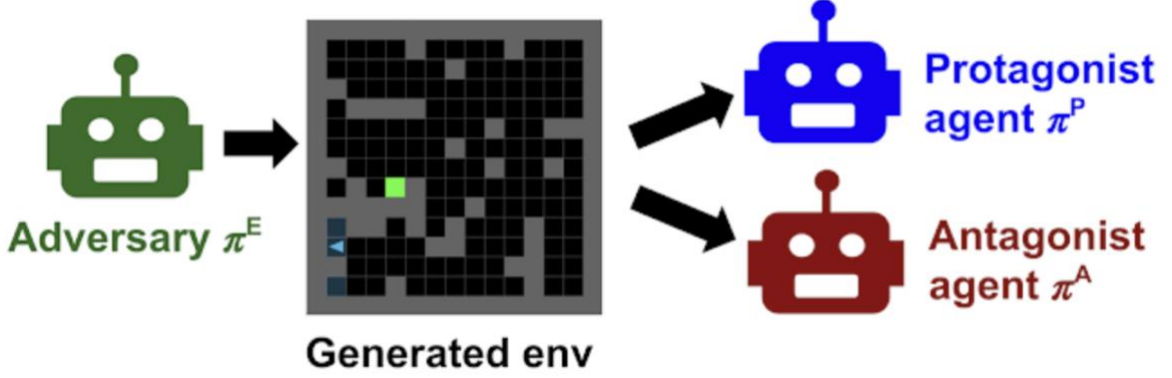


Figure 3. PAIRED method, to create an autocurricula of maze difficulty.

However, an agent supposed to generate an entire maze has outputs in a higher number of dimensions than a regular agent playing the maze.

To that end, a specific type of neural architectures was used: Multigrid Neural Networks.

This kind of architecture was presented in “Multigrid Neural Architectures“. The motivation behind such an architecture are mainly to conserve the structure of the desired output. This kind of architecture proved to be useful particularly on image processing. In fact, on the CIFAR data set it showed accuracy improvement by benefiting from the pyramid structured data representation.

In the field of Reinforcement Learning, it is important for an agent to optimize on previous experiences. To adapt this concept to the field of unsupervised environment design, the paper “Replay-Guided Adversarial Environment Design” makes use of prioritized level replay [Prioritized Level Replay]. This method permits to order the importance of previous mazes by their learning potential to the agent.

3. Contribution

The interest of that project is to study how human mood can be improved by playing games adapted to the player. To that end, we simulate a human behavior with an agent playing a

maze, and see if changing the maze itself can result in changing behavior for the agent, and potentially for a human.

We want to simulate a life example of a game played by a human to affect his current state of depression and anxiety.

In this project, multiple aspects had to be discussed and decided before implementation to restrict the scope of research.

3.1. Motivation of the project

The goal of this project was to observe how the behavior of an agent playing a complicated maze could be changed to observe what humans would identify as anxiety and depression as explained in “The Evolutionary Origins of Mood and Its Disorders”.

Recalling Figure1. Which explained depression as a lack of interest toward low reward actions and anxiety the fear caused by low levels of pain/loss.

To that end, this project proposes an idea to identify those two parameters within an agent.

3.2. How to quantify depression and anxiety in an agent?

Knowing that depression is a function of how much interest an agent must gain to take an action, we can identify it by forwarding a complicated maze in the DQN and see if any action reaches a certain value of Q-value in the Q-value learning framework. In fact, if at the beginning of the maze the agent only outputs low Q-values, it means it doesn't necessarily prefer an action over another, hence lacking motivation.

When it comes to anxiety, it is the behavior observed when an agent is scared to face a certain level of loss. To quantify that value, we can simply add “traps” to the maze that will lead to a punctual of moderate impact compared to the final reward. Hence, to see the level of anxiety of an agent we can simply place it in a maze, surrounded by traps and see the variation of the Q values once again.

3.3. Setup of the experiment

The Player Agent throughout this report will describe the agent playing the game.

Whenever he makes a move, he loses reward proportional to the size of the maze, given by the following formula:

$$Loss_{useless\ step} = -\frac{1}{Height * Width} \quad (1)$$

This loss was chosen so that longer maze don't have highly decreasing rewards.

Whenever he passes through a trap, the agent will lose twice the previous loss:

$$Loss_{trap} = 2 * Loss_{useless\ step} \quad (2)$$


If the agent found the end of the maze, the game would also break and add 1 to the total reward. By summing the rewards over steps, it permits the Player Agent to try to improve itself by using shorter paths to the ending point, useless steps decreasing its cumulative reward.

The Training Maze in which the agent plays will be of definite size 5x5 to limit the complexity required by the agent to train in. It will be generated using Domain Randomization. The number of traps and walls will be specified by us, to see if we can increase the complexity.


Note that a side function checks if the maze is still doable, even if it wasn't generated by Domain Randomization. This function is called Maze_solver.

In the mazes, the followings elements represent different parts of the maze:

“0” represents the position of the player

“” are the walls of the maze

“” is the end of the maze

“” are the traps of the maze, they lead to twice the loss of a useless step

The Difficult Maze will be of size 5x5 and will have the structure showed in the following figure.



Figure 4. Difficult maze architecture

To train each agent, Deep Q-Networks were implemented to keep a general powerful solution available.

The optimization function was already detailed in the literature review.

The code to implement the maze is made available in the Git Repository, with the code to train the agent in it.

When training, the agent had 75 steps to find the end of the maze, before the game would stop and count its current reward as the final reward.

It was then trained for 100 epochs on the data it obtained.

3.4. Methodology of the experiment

The methodology used to assess our method is as following:

1. **Method 1:** We will first train an agent on the difficult maze, and see if it finds a solution.
2. We will then find the Q-values associated with these situations:

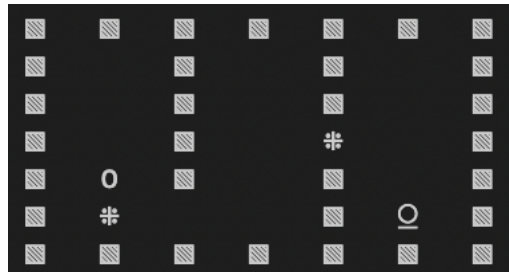


Figure 5. Situations 1 from which we will try to see if we observe any “anxiety” of going down

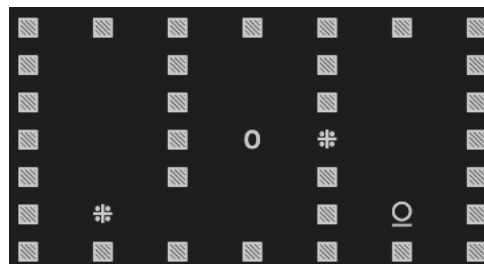


Figure 6. Situation 2 from which we try to observe “anxiety” of going right

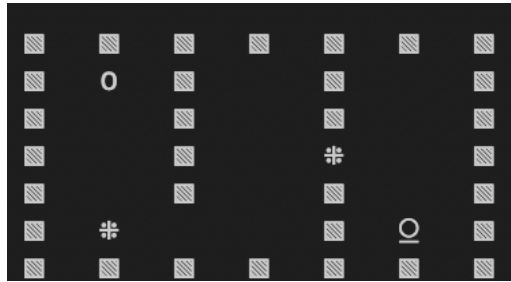


Figure 7. Situation 3 from which we try to observe depression.

3. **Method 2:** We will train another agent using Domain Randomization, with walls and traps placed randomly, but still in doable mazes.
4. Test that agent on the difficult maze and compute the “anxiety” and “depression” scores from the situations mentioned earlier.
5. **Method 3:** Train another agent on N different mazes SEPERATLY. This will permit us to compute the “difficulty” of each randomly generated maze. To estimate this difficulty, we will compute the average epise time while training. In fact, if a maze is really difficult, the agent will hardly find a solution, or it will take time for it. Hence, the shorter the episodes, the easier the mazes are.
6. Based on those difficulties, order the mazes, reset the agent’s parameters and train the agent on the mazes ordered by difficulty. Then test on the Hard Maze.
7. Compute the “depression” and “anxiety” Q-values.
8. **Method 4:** Just like Method 3 but we will train the agent on the hard maze at the end.

The main goal of this experiment isn’t really for the agent to solve the hard maze, but to see the Q-values after each method associated with certain situations, to see if the agent displays certain “depression” or “anxiety” behaviors.

The following figures represents the methodology used to see if this method affected the agent.



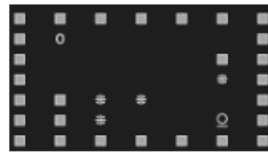
Method 1:

Train and test on
Hard Maze



Method 2:

Train on N random
mazes, one after the
other



⋮



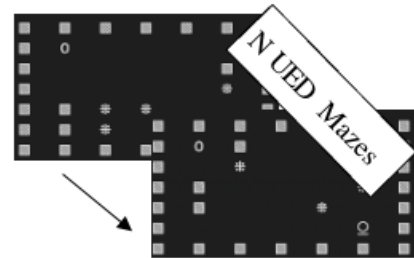
N UED Mazes

Test on Hard Maze



Method 3:

Train on N random
mazes,
SEPERATLY



Order the mazes by
difficulty

Train on N random
mazes, one after the
other by order of
difficulty



⋮



N UED Mazes

Test on Hard Maze

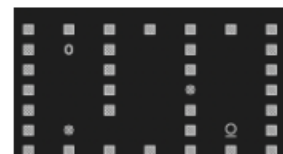
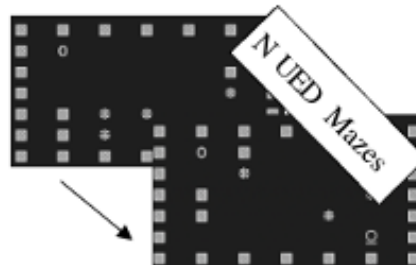


Figure 8. Methodology of the 3 first methods, to test UED on Mood Training

Method 4:

Train on N random
mazes,
SEPERATLY



Order the mazes by
difficulty

Train on N random
mazes, one after the
other by order of
difficulty



⋮



N UED Mazes

Train and test on
Hard Maze



Figure 9. Method 4 for Mood training with UED

Some concerns rose as the limits of the experiment were discussed:

Should the maze randomly generated always have a solution?

By ensuring that, we reduce the effect of randomness we set in the problem, and affect the way the agent will learn that, for example, being surrounded by walls isn't a situation it should be placed in.

What variables can the maze randomization affect?

It would be too complicated to randomize more variables as it would be hard to discern the effect of each of them on the results.

4. Experimental Results

4.1. Learning Stages of the different Methods

4.1.1. Method 1

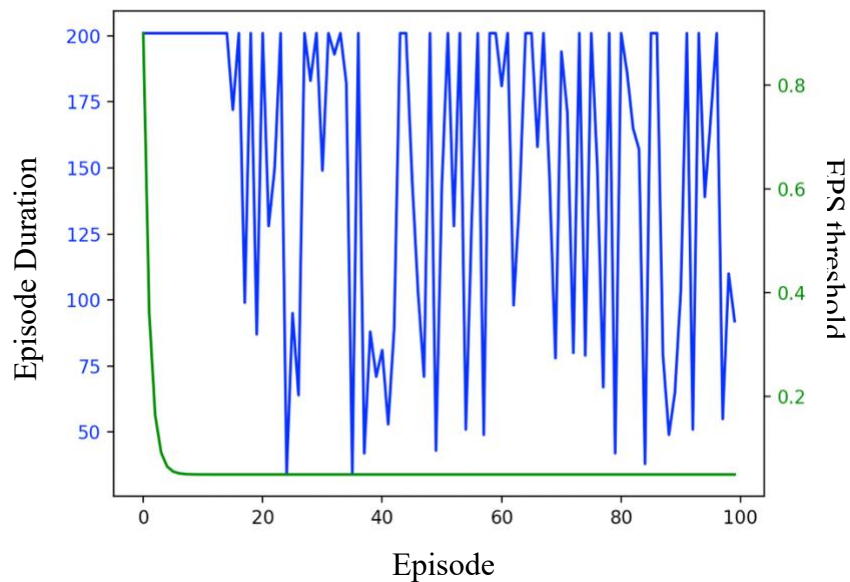


Figure 10. Episode Duration and EPS threshold for **Method 1**

The Method 1 didn't solve the Hard Maze.

4.1.2. Method 2

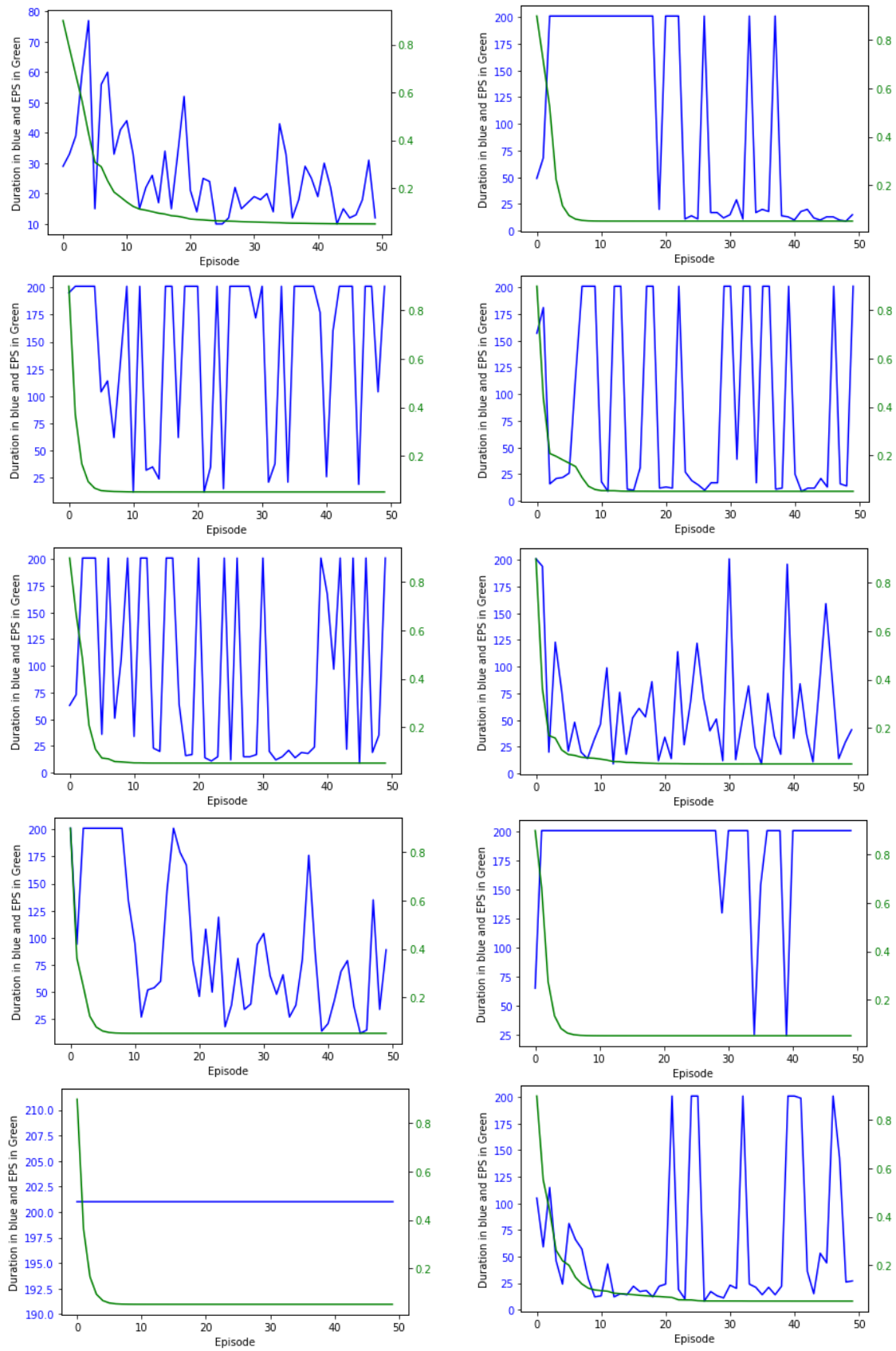
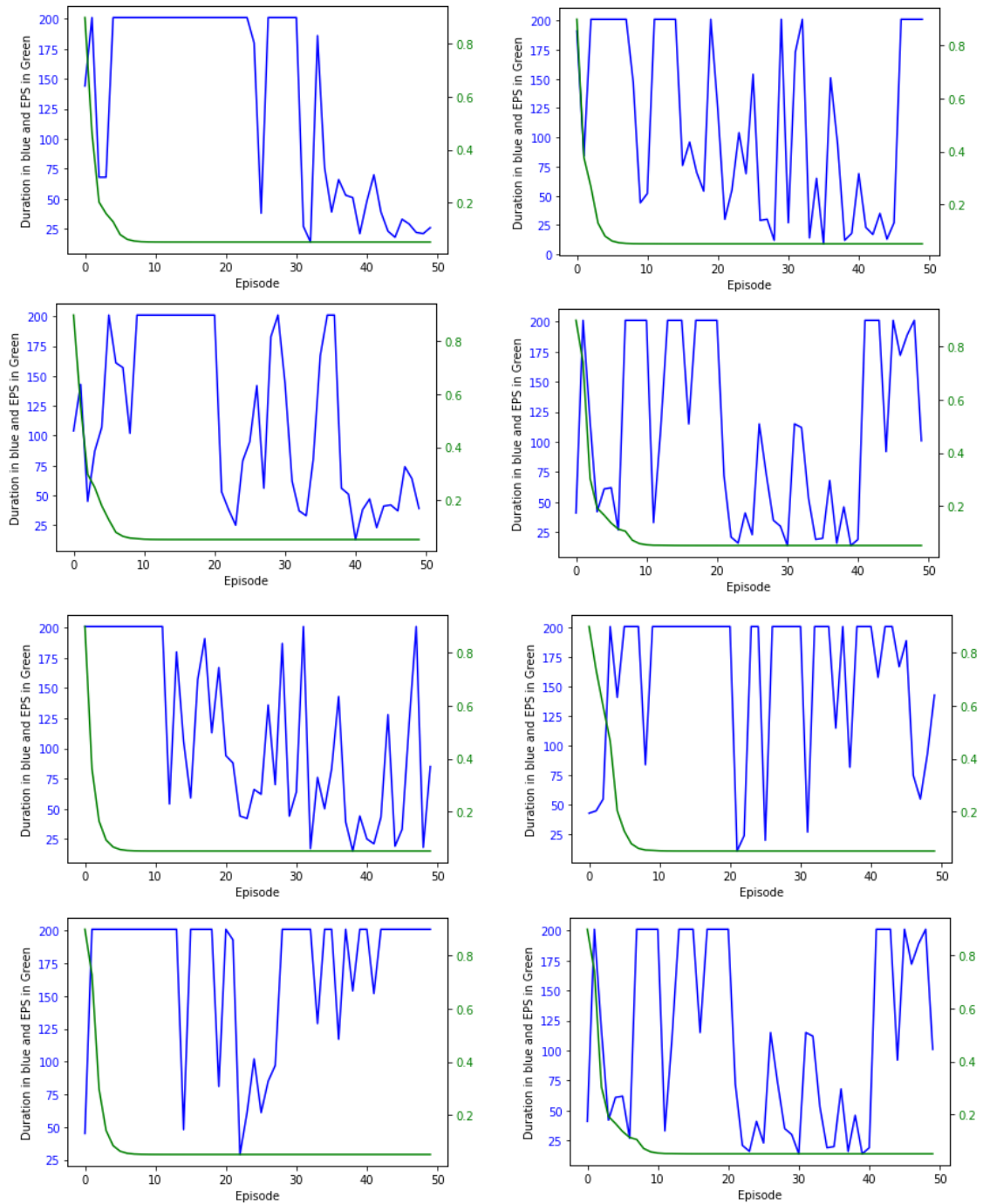


Figure 11. Episode Duration and EPS threshold for Method 2 on the 10 generated maze

The Method 2 didn't solve the hard maze.

4.1.3. Method 3 and 4



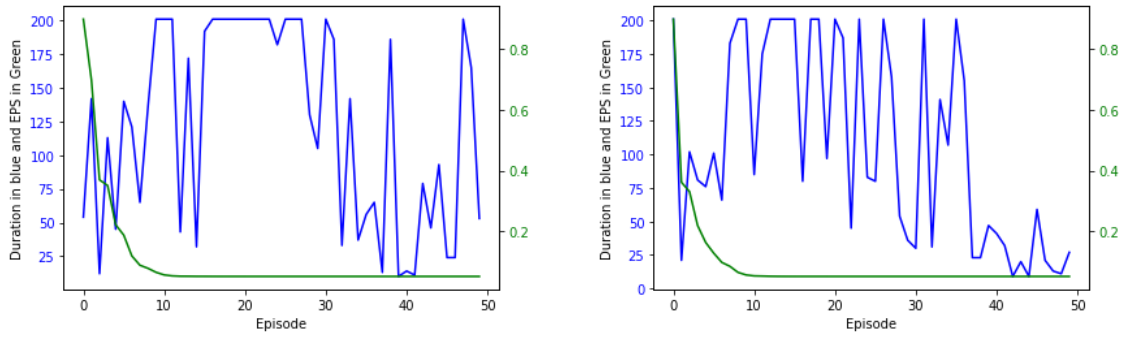
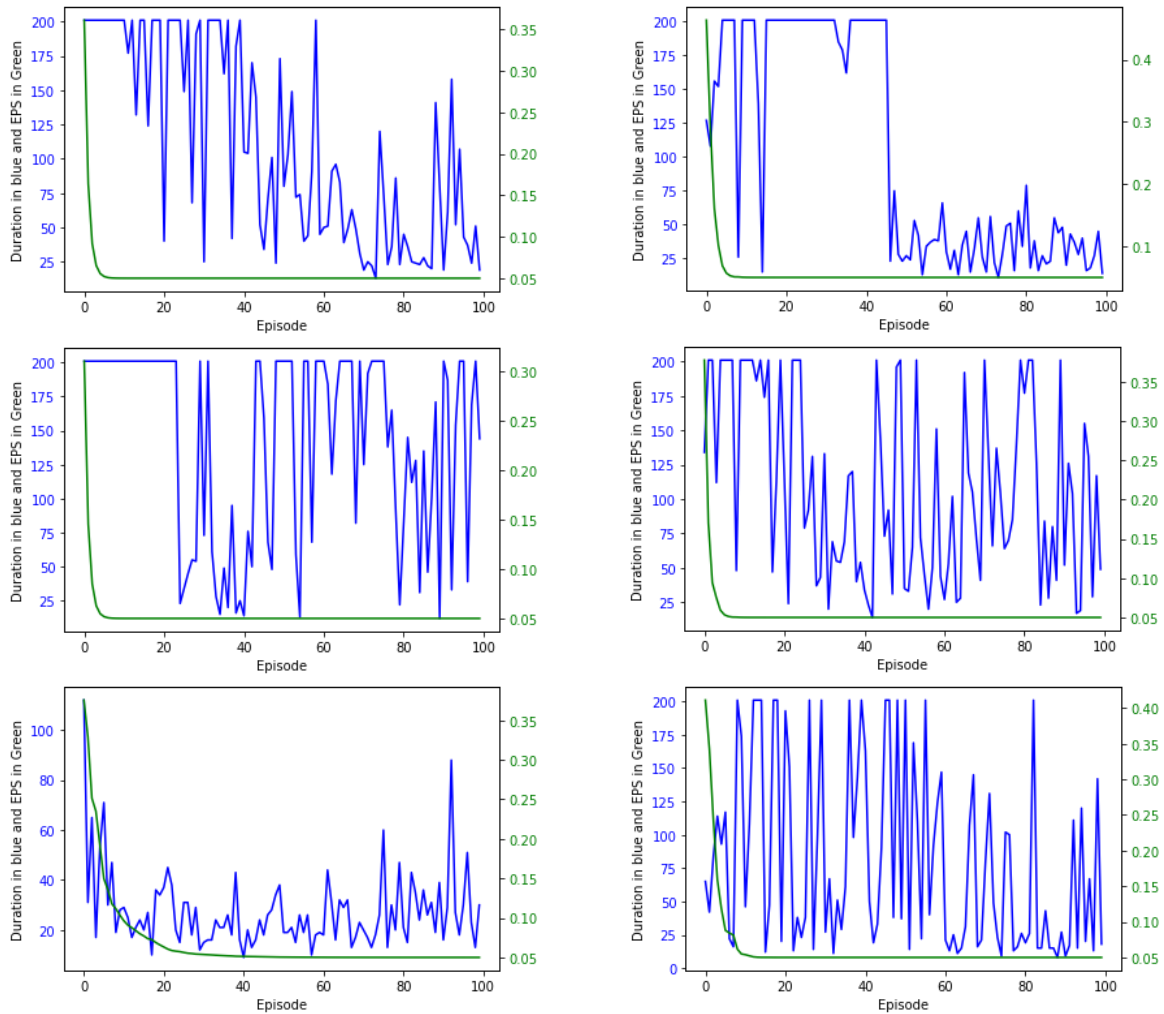


Figure 12. Episode Duration and EPS threshold for Method 3 and 4 on the 10 generated maze

The mean durations recorded for mazes 1 to 10 are:

130.7, 110.7, 159.82, 115.82, 105.46, 109.0, 121.74, 170.8, 158.18, 112.96

Based on those values, the mazes were re-ordered by increasing difficulty which led to the following trainings.



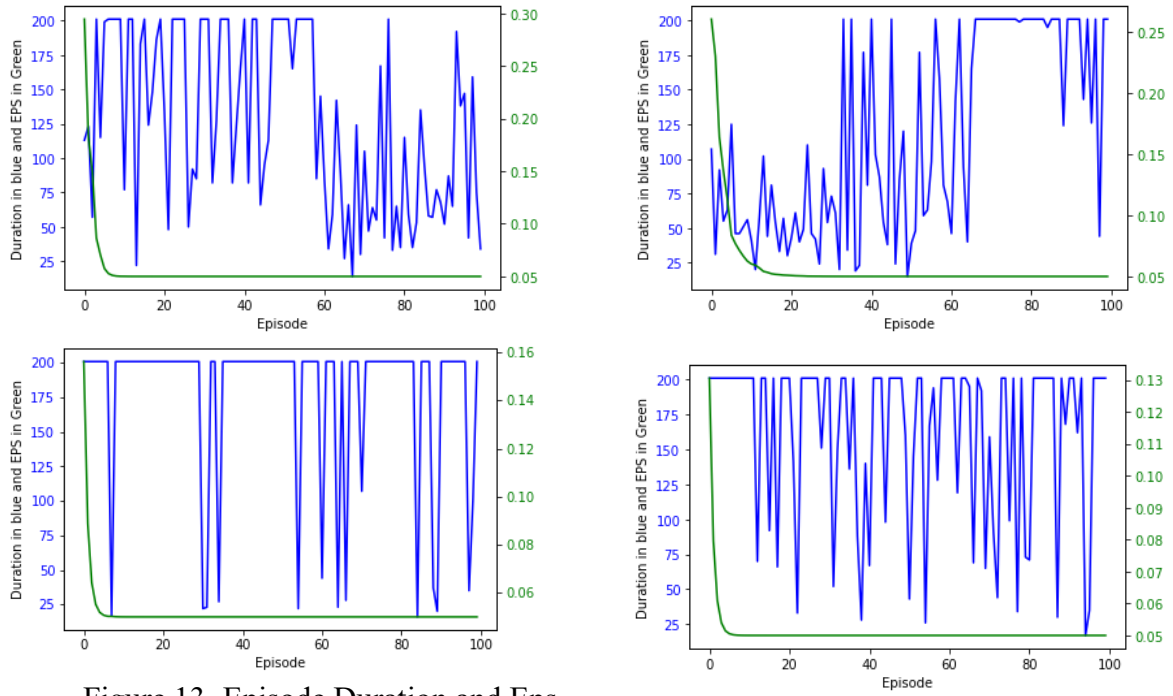


Figure 13. Episode Duration and Eps

threshold for Method 3 and 4 on the 10 generated training on them successively

Method 3 didn't solve the hard maze.

By retraining for Method 4 on the hard maze using the previous computed weights on method 3, we obtained the following figure.

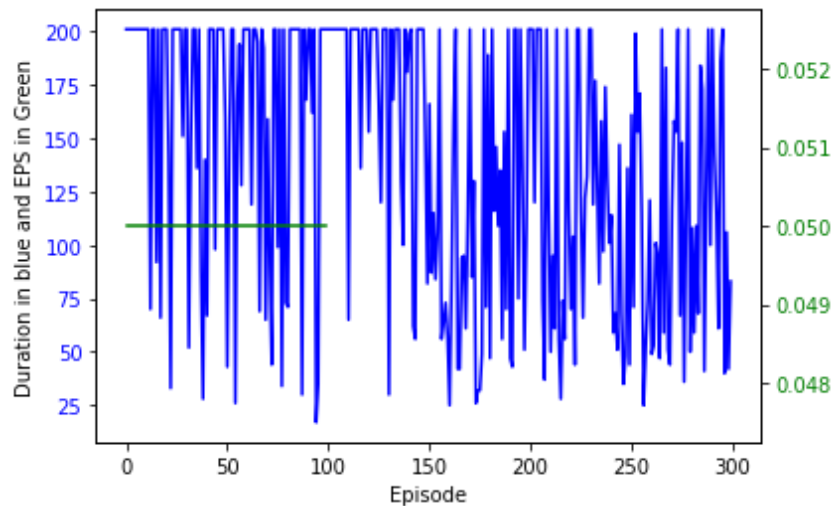


Figure 14. Episode Duration and Eps threshold for Method 4 when retraining on the hard maze

4.2. Associated Q-Values for each methods in the different situations

The following table shows the results observed in those setups.

	Up	Right	Down	Left
Method 1	0.2443	0.2638	0.2436	0.2435
Method 2	0.2443	0.2638	0.2436	0.2435
Method 3	0.6139	0.6148	0.6111	0.6144
Method 4	0.6560	0.6559	0.6620	0.6545

Table 1. Q-values recorded on situation 1.

	Up	Right	Down	Left
Method 1	0.2443	0.2638	0.2436	0.2435
Method 2	0.2443	0.2638	0.2436	0.2435
Method 3	0.6139	0.6148	0.6111	0.6144
Method 4	0.6560	0.6559	0.6620	0.6545

Table 2. Q-values recorded on situation 2.

	Up	Right	Down	Left
Method 1	0.2443	0.2638	0.2436	0.2435
Method 2	0.2443	0.2638	0.2436	0.2435
Method 3	0.6139	0.6148	0.6111	0.6144
Method 4	0.6560	0.6559	0.6620	0.6545

Table 3. Q-values recorded on situation 3.

5. Discussion of the results

As it can be observed in the Tables 1 to 3, the first Method didn't provide any solution for the maze. In fact, no matter the situation the maze was in, the agent didn't find any interest in adapting its behavior to the maze. However, the training as observed in Figure 10 showed that the maze was sometimes solved in less than 10 steps, but these solutions weren't stabilized by the learning agent.

For Method 2, It can be clearly seen that some mazes showed much greater complexity than others. In fact, the 9th maze wasn't ever optimized as it was highly complex (note that each generated maze was checked to be doable, so this can't be explained by a maze not doable). Training on many UED generated mazes didn't prove to change the Q-values associated with each of the situations we were testing. This can be explained by the fact that the difficulty of the generated mazes was not near the difficulty of the Hard Maze, hence not bringing any advantage to the agent maze whom discovered the Hard Maze while playing it for the first time.

For both Method 3 and 4, we realize that ordering the mazes by levels of complexity and retraining on them successively didn't show any improvement to how the hardest mazes were solved. This can probably be caused by the fact that we only generated 10 mazes. In fact, since the complexity of the mazes can be on many different levels, 10 different levels don't assure that the complexity will be of proportional difficulty from one to the other.

For Method 3, we can see in the Tables 1 to 3 that the Q-Values have been affected and now display greater values. It seems like training on increasingly complex mazes has permitted the agent to learn that any situation displayed could show reward on the long term.

For Method 4, Tables 1-3 show us that the Q-values have once again increased but don't seem to adapt to the situation the maze is in.

Furthermore, Figure 14 shows that the training epochs on the Hard maze were complicated but the agent still managed to finish the maze many times

Overall, the “depression” score, ie how low the Q-values are at the beginning of the maze seem to have been affected by the UED Maze implementation.

However, this implementation didn’t seem to show any interesting value of the “anxiety” score, ie. Q-Values that should be observed for actions leading in “traps”.

7. Conclusion

This report aimed to show similarity that exist between human behavior and observations in multi-agent setting.

This method showed inefficient in our problem setup to affect the “anxiety” score of our agents. However, the Q-values for our method showed increasing values compared to the simpler method. Hence showing that proportionally, the agent computed a greater interest to try to find solutions in the hard maze than by just training on it.

To further study this topic, it would be interesting to investigate why the agent didn’t stabilize in most scenarios. By using more powerful computation resources, increasing the number of training steps, the length of each trial and the size of the convolutional network, we might be able to address this issue.

Furthermore, the aim of this project was to create an autonomous maze builder, which would put walls wherever it wanted.

Because of computation limitations, implementing such a method would have asked for much more resources. However, this would be the proper UED implementation that could lead to interesting results.

8. Bibliography

Cournot A. A.

1838. *The Recherches*

Edgeworth, F.

1881. *Mathematical Psychics*

J. von Neumann and Fréchet M.,

1953. *Communication on the Borel Note*. *Econometrica*

Jhon F. Nash Jr.

1950. *Equilibrium points in n-person games*

Bicchieri C.

2004, *Rationality and Game Theory*, The Oxford Handbook of Rationality

Y. Wen , Y. Yang, J. Wang

2020, *Modelling Bounded Rationality in Multi-Agent Interactions by Generalized*

Recursive Reasoning, University College London, Huawei Research & Development

U.K.

Coelho Prates R.

2014, *Power in Game Theory*, Universidade Federal do Paraná

J. Z. Leibo, E. Hughes, M. Lanctot and T. Graepel

2019, *Autocurricula and the Emergence of Innovation from Social Interaction: A*

Manifesto for Multi-Agent Intelligence Research, Deepmind

Y. Wen, H. Chen, Y. Yang, Z. Tian, M. Li, X. Chen, J. Wang

2021. *Multi-Agent Trust Region Learning*, ICLR

C. F. Camerer, T. H. HO JUIN-KUAN CHONG

2004. *A cognitive hierarchy model of games*, The Quarterly Journal of Economics,

Volume 119, Issue 3.

Y. Wen, Y. Yang, R. Luo, J. Wang, W. Pan

2019, *Probabilistic Recursive Reasoning for Multi-Agent Reinforcement Learning*,

ICLR

P. Dayana and Y. Nivb

Reinforcement learning: The Good, The Bad and The Ugly, 2008

V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, M. Riedmiller

2013. *Playing Atari with Deep Reinforcement Learning*

- J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov
2017. *Proximal Policy Optimization Algorithms*
- F. C.A. Groen, M. T.J. Spaan, J. R. Kok, and G. Pavlin
2005. *Real world multi-agent systems: information sharing, coordination and planning*. Logic, Language, and Computation, 6th International Tbilisi Symposium on Logic, Language, and Computation, TbiLLC 2005
- Y. Wen,
2020. *Modeling Mutual Influence in Multi-Agent Reinforcement Learning*, UCL
- J. Devlin, M.-W. Chang, K. Lee, K. Toutanova
2019, *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*
- Jones, H. E., and Batalla, M.
1944. *Transfer in children's maze learning*, Journal of Educational Psychology, 35(8), 474–483.
- Q. J M Huy , D. Renz
2017. A Formal Valuation Framework for Emotions and Their Control
- V. Shuman, D. Sander, and K. R. Scherer
2013. *Levels of Valence*
- D. Nettle and M. Bateson
2012. *The evolutionary origins of mood and its disorders*
- Unknown Author
2019. *The Science Of Emotion: Exploring The Basics Of Emotional Psychology*. UWA
- M. Chevalier-Boisvert, D. Bahdanau, S. Lahlou, L. Willems, C. Saharia, T. Huu Nguyen, Y. Bengio
2019, *BabyAI: A Platform to Study the Sample Efficiency of Grounded Language Learning*, ICLR 2019
- H. Keren, C. Zheng, D. C Jangraw, K. Chang, A. Vitale, R. B Rutledge, F .Pereira, D. M Nielson, A. Stringaris
2021. *The temporal representation of experience in subjective mood*, eLife
- M P Page and D Norris
1998, *The primacy model: a new model of immediate serial recall*
- DJ Strouse, K. R. McKee, M. Botvinick, E. Hughes, R. Everett
2021. *Collaborating with Humans without Human Data*, NeurIPS 2021

T. M. Moerland, J. Broekens, C. M. Jonker

2017. *Emotion in Reinforcement Learning Agents and Robots: A Survey*

J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, P. Abbeel

2017. *Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World*

M. Dennis, N. Jaques, E. Vinitzky, A. Bayen, S. Russell, A. Critch, S. Levine

2020. *Emergent Complexity and Zero-shot Transfer via Unsupervised Environment Design*

T.-W. Ke, M. Maire, S. X. Yu

2016. *Multigrid Neural Architectures*

M. Jiang, M. Dennis, J. Parker-Holder, J. Foerster, E. Grefenstette, T. Rocktäschel

2021. *Replay-Guided Adversarial Environment Design*, NeurIPS 2021

M. Jiang, E. Grefenstette, T. Rocktäschel

2020. *Prioritized Level Replay*