The Effects of Intrinsic Motivation Signals on Reinforcement Learning Strategies

Effekte von intrinsischer Motivation auf Reinforcement Learning Strategien
Bachelor-Thesis von Yannik Frisch aus Schwalmstadt
Tag der Einreichung:

1. Gutachten: Prof. Dr. Jan Peters
2. Gutachten: Dr. Elmar Rückert
3. Gutachten: Svenja Stark

TECHNISCHE
UNIVERSITÄT
DARMSTADT

The Effects of Intrinsic Motivation Signals on Reinforcement Learning Strategies
Effekte von intrinsischer Motivation auf Reinforcement Learning Strategien

Vorgelegte Bachelor-Thesis von Yannik Frisch aus Schwalmstadt

1. Gutachten: Prof. Dr. Jan Peters
2. Gutachten: Dr. Elmar Rückert
3. Gutachten: Svenja Stark

Tag der Einreichung:

Erklärung zur Bachelor-Thesis gemäß §38 Abs.2 APB der TU Darmstadt

Hiermit versichere ich, die vorliegende Bachelor-Thesis ohne Hilfe Dritter nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die aus Quellen entnommen wurden, sind als solche kenntlich gemacht worden. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Mir ist bekann, dass im Falle eines Plagiats (§38 Abs.2 APB) ein Täuschungsversuch vorliegt, der dazu führt, dass die Arbeit mit 5.0 bewertet und damit ein Prüfungsversuch verbraucht wird. Abschlussarbeiten dürfen nur einmal wiederholt werden.

Bei der abgegebenen Thesis stimmen die schriftliche und die zur Archivierung eingereichte elektronische Fassung überein.

Darmstadt, den 8. Dezember 2017

____

(Yannik Frisch)

Abstract

Using neurobiological and psychological models in robotics and machine learning was of growing interest in the last years. Whereas common algorithms in the reinforcement learning framework tend to get stuck in local maxima while exploring the environment, intrinsic motivation modules can be used to extend these algorithms and push the reinforcement learning agent out of its equilibrium, similar to a human who gets bored of a task he fulfills many times or makes no progress while trying to fulfill it. This thesis gives an overview of models of intrinsic motivation founded on neurobiology and psychology, before presenting a computational view of extending algorithms of the reinforcement learning framework with intrinsic motivation models. Several existing theoretical models and related work are presented, achieving a better performance than classic algorithms, regarding the exploration/exploitation trade-off and driving the autonomous learning of an agent. Three of these models, maximizing incompetence motivation (IM), maximizing competence motivation (CM) and competence progress motivation (CPM), are implemented, in which the authors define competence by the number of primitive actions an agent needs to reach a terminal state and add a negative intrinsic reward for reaching this terminal state, which increases or decreases proportionally to the competence of the agent. The models are evaluated on four simulated scenarios and compared with the performance of the classic reinforcement learning algorithm SARSA and a time-decreasing-epsilon (TDE) modification of it. Using CM achieves at best a faster convergence towards the same terminal state as SARSA, whereas using IM and CPM results in an agent being pushed out of its equilibrium of local maxima and leads to more exploration, while still maximizing the expected external reward. An agent using these models is able to learn skills which an agent using classic SARSA would never explore. The presented related work and the implemented models show that using models for intrinsic motivation together with reinforcement learning algorithms results in well-performing behavior for tasks, on which classic algorithms would fail or get stuck in local maxima, and so provide a useful base for future work and research to build up a fully autonomous learning system.

Zusammenfassung

Die Verwendung von neurobiologischen und psychologischen Modellen in Robotik und Machine Learning Frameworks hat in den letzten Jahren eine wachsende Beliebtheit erfahren. Während bekannte Algorithmen aus dem Reinforcement Learning Framework dazu tendieren, in lokalen Maxima stecken zu bleiben, können Module zur intrinsischen Motivation diese Algorithmen erweitern und für eine Bewegung des Reinforcement Learning Agenten aus seinem Stillstand im lokalen Maximum heraus sorgen. Dieses Verhalten ist ähnlich einem Menschen, der von einer sich ständig wiederholenden Aufgabe, oder einer Aufgabe bei der er keinen Fortschritt macht, gelanweilt wird. Diese Thesis präsentiert zunächst einige psychologisch und neurobiologisch fundierte Modelle zur Definition von intrinsischer Motivation. Anschließend wird diese von einer computationalen Seite betrachtet, mit Blick auf das Reinforcement Learning Framework. Es werden mehrere bereits existierende theoretische Modelle und praktische Anwendungen in verangener Arbeit zur Verwendung von intrinsischer Motivation vorgestellt, die es ermöglichen, bessere Leistungen im Hinblick auf den exploration/exploitation trade-off und das autonome Lernen eines Agenten zu erzielen. Drei dieser Modelle, Maximierung von Inkompetenz Motivation (IM), Maximierung von Kompetenz Motivation (CM) und Kompetenz-Fortschritt Motivation (CPM) werden anschließend implementiert, wobei die Authoren Kompetenz über die Anzahl benötigter Zeitschritte oder Aktionen definieren, die ein Agent benötigt um ein Ziel zu erreichen. Eine negative intrinsische Belohnung, welche proportional zur Kompetenz des Agenten wächst oder sinkt, wird auf die externe Belohnung addiert. Die Modelle werden auf vier simulierten Szenarien evaluiert und mit der Leistung des klassischen Reinforcement Learning Verfahrens SARSA, sowie einer Abwandlung mit einem über die Zeit sinkenden Anteil an zufälliger exploration (TDE), verglichen. Während die Verwendung von CM im bestenfalls eine schnellere Konvergenz zum selben Ziel wie SARSA ermöglicht, wird ein Agent mit IM oder CPM aus seinem Stillstand im lokalen Maximum bewegt, was zu einem erhöhten Anteil an exploration führt, wobei trotzdem die erwartete externe Belohnung maximiert wird. Ein Agent mit diesen Modulen ist in der Lage Aufgaben zu lernen und Belohnungen zu erhalten, die ein klassischer SARSA Agent nie entdecken würde. Die präsentierten Arbeiten und eigens entwickelten Modelle zeigen dass die Verwendung von intrinsischer Motivation für Reinforcement Learning Strategien in der Lage ist, Aufgaben zu meistern, an denen klassische Verfahren scheitern würden oder bei denen sie in lokalen Maxima stecken bleiben würden. Diese Modelle bieten somit eine nützliche Grundlage für zukúntige Arbeiten mit dem Ziel ein voll-autonomes lernendes System zu entwickeln.

Acknowledgments

I would like to thank my family and all my friends for supporting me.
Thanks to Svenja Stark, Dr. Elmar Rückert and Prof. Dr. Jan Peters for supervising my thesis.

Contents

Figures

List of Figures

1 Introduction

In the last century, a lot of scientific effort has been put into trying to understand human behavior. One of the oldest psychological questions is why humans do, or do not do, specific things. Researchers in psychological science explain this with the concept of motivation and its corresponding scientific area motivational psychology, which is strongly connected to the area of emotional psychology.

Motivation can be defined, as done by Rheinberg, as the "activational force of the current life situation towards a positive goal state"[1]. Such short definitions are often kept very tight and need to be extended for special cases. For the one above the positive goal state might as well be the reduction or avoidance of negative states. The concept of an activity's attraction being only dependent on the rewarding goal state they are aiming for, as explained by Heckhausen in [2] and Vroom in [3], can be used to explain some but not all human behavior.

Humans sometimes focus their activities only on the reward they are about to receive for them. Imagine an athlete, who trains a lot (activity) to join a well paid and famous team to earn a lot of money (reward). Now, as he tries to stay fit for the summer, he has to train during the winter as well. No one will say he is excited by rejecting the very tasty but unhealthy Christmas meal or declining a friends invitation for a party on a competition day. In case the expected reward will be high enough, we sometimes even do so-called aversive activities, which we would really avoid otherwise. The athlete might, for example, have to walk through a snowstorm in order to reach his training place during the winter season to stay fit for the incoming summer season.

On the other hand, some activities bring pleasure to humans by themselves, without expecting a rewarding goal state. Rheinberg showed that students, who took a protocol of their day every 10 minutes, spend on average 46% of the time they are awake on such activities, like watching a movie or playing video games [4]. Just like in the proverb "A showcase, not just a rally", the reward is not achieved by a goal state after performing the activity, but by the activity itself. We as humans tend to perform these activities as long as we are rewarded while doing it, instead of ending them as fast as we can and be rewarded afterwards. This is obvious for activities close to the body, like eating sweets or sexual activity, but can also be shown for other activities. Sometimes humans perform these activities even when they are extremely aversive, like smoking or eating unhealthy food.

Until now, we have only talked about so-called inhomogeneous signed constellations, where the activity is sensed in a positive way and the reward is not, or the activity is negative and the reward is sensed positive. Of course homogeneous constellations exist as well. Going back to our athlete, he might just enjoy the training a lot, and get paid for it, too. For these constellations, humans do not tend to stay in the first reached goal state, they actively seek for new goal states which acquire similar forms of activities to reach them: Our athlete will not stop after training once. He will continue to do so and become fitter and fitter, while he also really enjoys the training process itself. The following section will give an overview of the psychological definitions of the concept of intrinsic motivation, which is responsible for him to do so.

1.1 Intrinsic Motivation - Psychological View

Written by Ryan and Deci, "Intrinsic motivation is defined as the doing of an activity for its inherent satisfaction rather than for some separable consequence. When intrinsically motivated, a person is moved to act for the fun or challenge entailed rather than because of external products, pressures, or rewards" [5]. This behavior is very often observed for young children, as they constantly try to explore the world, touch, bite or throw things, or shout at new objects they find, but it can somehow still be observed for grown adults, when they play cards or soccer with friends, watch a good TV-show or lay on the beach and enjoy the sun. The definition was given in contrast to the one of "extrinsic motivation" which is described in [5] as "a construct that pertains whenever an activity is done in order to attain some separable outcome. Extrinsic motivation thus contrasts with intrinsic motivation, which refers to doing an activity simply for the enjoyment of the activity itself, rather than its instrumental value."

The aspect that differentiates the two types of motivation is instrumentalisation. This is an important difference to another differentiation, which is used in a lot of psychological but also computational literature: The one of internal and external motivation. These concepts only differentiate between the location of the system that gives the observed reward for an activity. In our example, the athlete would be driven by an internal motivation to do sports if he trains a lot to become fitter and healthier. The rewarding system is his own body. On the other hand, he would be externally motivated if he only exercises to earn money, where the reward producing system would be located outside his body. Lepper showed that the first type of motivation is often resulting in a stronger drive for the activity, than external motivation [6].

It should be clear, that mixing intrinsic with internal and extrinsic with external is wrong and can be confusing, because

extrinsic motivations can be internal and vise versa. Ryan and Deci have shown the existence of different kinds of instrumentalization that can be classified as self-determined to a certain grade [5]. This can become more clear with some simple examples:
The athlete that is exercising is currently very disappointed with the situation and only does the sports to not get kicked out of the team. The drive for the action is now produced by an external system, as he currently does not enjoy the situation and only exercises to avoid possible sanctions, he is extrinsically and externally motivated. He could also be training hard because he aims to join a more famous team and receive an even better paying. In this case, the driving force is located inside the athlete, but the training is again not done for its own sake, but for the possible rise into the new team. His behavior is internally and extrinsically motivated. Third, it is possible that our athlete is only exercising for the fun of it, to become fitter and maybe even learn some new training methods for the future. In this case he is intrinsically and internally motivated.
We also need to be aware of possible intersections or overlappings of the given definitions. The athlete might partly be extrinsically motivated by earning a lot of money, and partly intrinsically by the fun of doing sports and learning new exercising steps. We could also imagine him being intrinsically motivated to go to a gym to train, but he needs to ride his bike to get there. In case he does not like the activity of riding the bike itself, he is extrinsically and internally motivated for this behavior, which spins out of the intrinsically motivated behavior of exercising at the gym [7].
Despite these definitions, earlier research has already led to some different theories about which activities become intrinsically motivating for humans. It is important to know, that these definitions are inconsistent concerning the people and time: One activity might be completely irrelevant and uninteresting for a certain person, while it is very interesting for another person. Second, the activity might be interesting for this person right now, but not in the future. These theories can be grouped more or less into four groups, which are defined below, analogue to Oudeyer and Kaplan in [7], in a chronological order.

1.1.1 Drives to Manipulate, Drives to Explore

These theories all build on the theory of drives, as described by Hull in 1943 [8]. Drives emerge from deficits in a specific region of a humans needs, for example the need for food or social integrity. The resulting drives will then push the human towards actions which reduce this needs, like eating or meeting up with some friends. In the 1950s, psychologists started to do research about the intrinsic motivation based on this theory of drives. For example, Montgomery described a drive for exploration in 1952 [9] and Harlow a drive to manipulate the humans environment and other humans in 1950 [10]. The theory of drives was mainly criticized by White in 1959, who argued that intrinsic motivations are not homeostatic and not the result of a deficit inside the humans body [11].

1.1.2 Reduction of Cognitive Dissonance

A different concept, the theory of cognitive dissonance, explained by Festinger in 1957 [12], proposed organisms being motivated to reduce the dissonance, resulting of the incompatibility between internal cognitive structures and the situations currently perceived. Based on this, Kagan stated in 1972 that an important motivation of humans is to reduce the uncertainty in the sense of the “incompatibility between (two or more) cognitive structures, between cognitive structure and experience, or between structures and behavior” [13]. The main critic on this theories was based on research that shows humans also tend to increase the uncertainty with a lot of their behavior, which holds especially for the phenomena of sensation seeking. This theory describes humans who tend to get themselves into new and very exciting situations, which are often also more or less dangerous [14]. Most of the time, we seem to look for some form of optimality between completely uncertain and completely certain situations [15].

1.1.3 Optimal Incongruity

This idea was mainly developed by Hunt in 1965, who wrote that children and adults reach out for the optimal incongruity of their body system [15]. For children as information-processing systems, stimuli with a discrepancy between the perceived and the standard level became very interesting. Dember and Earl described the incongruity or discrepancy in intrinsically motivated behavior as the difference between a person’s expectation and the properties of the stimulus [16]. Similar to that, Berlyne proposed a notion in 1960 that defines the most rewarding situations as those with an intermediate level of novelty between already familiar and completely new situations [17].

1.1.4 Motivation for Effectance, Personal Causation, Competence and Self-Determination

Some researchers disagreed with the notion of optimal incongruity and preferred the concept of challenge. They stated the motivations for effectance [11], personal causation [18], competence and self-determination [19] as the driving forces in the human body. In general, they describe the degree of control people can have on other people, external objects and themselves, as what is motivating them. Analogously to this, the concept of optimal challenge arose like in the theory of flow by Csikszentmihalyi [20]. This describes that if we have a scale for the difficulty of an activity and the skill or the time that is required to perform it, humans only enjoy activities and get only motivated for them if they are inside the flow area. Otherwise, they would get bored if the activity is too easy or get frustrated if the activity is too difficult. Figure 1.1 does show a simple visualization for this theory.

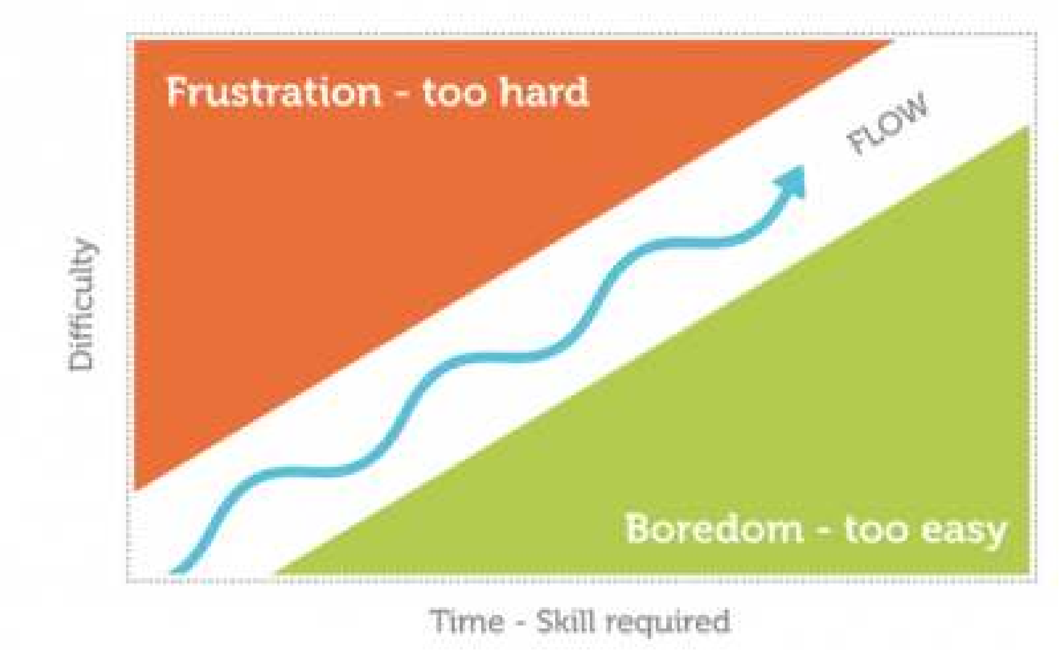


Figure 1.1.: Theory of Flow

Only activities inside the "flow" area are interesting and motivating and enable a focused attention on them. This picture ist taken from [21].

While those psychological definitions remain rather difficult, computer scientists try to find a clearer definition to use for their learning algorithms and robotic systems. A computational view is shown in the next section of this chapter.

1.2 Intrinsic Motivation - Computational View

We first give an overview of the reinforcement learning framework, which is used for autonomous development in robotics and machine learning systems, in order to self-explore an environment and maximize the observed reward while exploring. Secondly, we present a typology, given by Oudeyer and Kaplan [7], which gives several distinctions between motivational systems for these robotic and machine learning systems, which are based on the psychological definitions we have shown in the previous section.

1.2.1 Reinforcement Learning

The machine learning field of computational science deals with learning from data supported by personal computers, to do the math. These learning problems can be differentiated into supervised and unsupervised learning. Supervised learning problems deal with computing a known trait of the data, for example a probability distribution or a regression function. They need a supervisor or discriminant function to accept or decline the currently learned theory about the data. Unlike this, unsupervised learning algorithms try to find patterns in the data without being guided. Clustering is one well known example for this.

Beyond these types of algorithms there is a third type of learning problem, called reinforcement learning (RL), dealing with the problem of a learning agent interacting with its environment to achieve a goal, where the probabilities to successfully interact and the possible rewards are unknown. As described by Sutton and Barto, reinforcement learning is a computational approach to learn from interaction [22]. The learning agent discovers, which actions yield the most

reward, by trying them. It then learns from its own experience.
One of the main challenges in this field of machine learning is the trade-off between exploration and exploitation. The agent will find actions which are more rewarding than others for a defined state. The trade-off is between exploiting what it already knows in order to obtain reward, and explore the environment in order to make better action selections in the future [22]. We will see about ways to solve this trade-off in later sections.
The RL framework is used to solve a fully described Markov decision process(MDP) [23]. This control process models decision making, where the process, or learning agent, is in a state s, at each time step. The agent chooses an action a, moving it into the new state s' and observing a reward r for this move. The probability of successfully moving from state s to state s' by taking action a is described by the state transition function $P(s,a,s')$. These transitions need to yield the Markov property [24], which limits the conditionally dependency of the next state s' only by the current state s and the chosen action a.
To sum it up, the MDP can be described as a 5-tuple: $(S,A,P(),R(),\gamma)$, where S is the finite set of all possible states and A is the finite set of all actions. $P(s,a,s')$ is the transition model, defining the probability for action a in state s at time-step t leading to state s' at time-step $t+1$. $R(s,a,s')$ describes the immediate reward being received by taking action a in state s leading to state s'. Finally, $\gamma \in [0,1]$ is called the discount factor, which sets the importance of future rewards over present rewards.

The Policy

The solution for a MDP is a policy $\pi : S \rightarrow A$, which defines the action to choose for every state, in order to maximize the expected reward over time. One method for this is an epsilon-greedy policy, where the agent chooses the most rewarding action with probability $1-\epsilon$, or a randomly chosen action otherwise. This is a an approach to deal with the exploration-exploitation trade-off, where $\epsilon \in [0,..,1]$ is the tuning parameter, which might be fixed or adaptive based on some heuristics [25].

State-Value-Function

In temporal-difference reinforcement learning, which we focus on in this thesis, the agent uses a function V to express the value of a given state s_t. To improve his strategy π, he tries to come as close to the optimal solution V_π as possible. He does this by choosing an action in every iteration according to his strategy π, observe the reward r and adapt the function V according to

$$V(s_t) \leftarrow V(s_t) + \alpha(r_{t+1} + \gamma V(s_{t+1}) - V(s_t))$$

where α is called the learning rate and γ is the discount-factor.
The learning rate gives the strength of adaption of the current function V for every iteration. It determines to what extent old information will get overwritten by the newly acquired information. The discount factor is used to weight future rewards. A smaller value for γ makes the agent very shortsighted, while a bigger value makes him more sensitive for future rewards and strive for long-term high rewards. We will now show two different algorithms for this approach of reinforcement learning, both with individual benefits and shortcomings.

Q-Learning: Off-Policy

In Q-Learning, the agent values the current action a, not the state s. The function V becomes the function $Q(s,a)$ and is called action-value or state-action function [26]

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

This algorithm is a so-called off-policy algorithm, because it does differ from the current strategy π, while choosing a'. Algorithm 1 does show the pseudo-code for this algorithm.

```
Data: $S, A, \alpha, \gamma$
Initialize $Q(s,a)$ arbitrarily $\forall s \in S, a \in A$
for each learning episode do
    Initialize $s$
    while $s$ not terminal do
        Choose $a$ from $s$ using policy derived from $Q$ (e.g. $\epsilon$-greedy)
        Take action $a$, observe reward $r$ and next state $s'$
        $Q(s,a) \leftarrow Q(s,a) + \alpha(r + \gamma \max_{a'} Q(s',a') - Q(s,a))$
        $s \leftarrow s'$
```

Algorithm 1: Q-Learning Pseudo-Code

SARSA: On-Policy

The second algorithm is called State-Action-Reward-State-Action(SARSA) and does also learn a state-action function Q. But unlike Q-Learning, the algorithm does relate on the agents strategy π and is a so-called on-policy algorithm [27]

$$Q(s,a) \leftarrow Q(s,a) + \alpha(r + \gamma Q(s',a') - Q(s,a))$$

Where a and a' are chosen according to the policy π. Algorithm 2 does show the pseudo-code for the SARSA algorithm:

```
Data: $S, A, \alpha, \gamma$
Initialize $Q(s,a)$ arbitrarily $\forall s \in S, a \in A$ and $Q(s_{terminal}, \cdot) = 0$
for each learning episode do
    Initialize $s$
    Choose $a$ from $s$ using policy derived from $Q$ (e.g. $\epsilon$-greedy)
    while $s$ not terminal do
        Take action $a$, observe reward $r$ and next state $s'$
        Choose next action $a'$ from $s'$ using policy derived from $Q$ (e.g. $\epsilon$-greedy)
        $Q(s,a) \leftarrow Q(s,a) + \alpha(r + \gamma Q(s',a') - Q(s,a))$
        $s \leftarrow s'$
        $a \leftarrow a'$
```

Algorithm 2: SARSA Pseudo-Code

The on-policy SARSA algorithm enables learning relatively to the policy the agent follows, while the off-policy Q-learning learns relatively to the greedy policy. Under some common conditions, they both converge to the real value function [22]. Q-Learning will converge a little slower, but enables the possibility to continue the learning process while changing policies [27]. Though, SARSA being an on-policy method, takes the current policy the agent is following into account. It incorporates this into the update step of the state-action values, while Q-Learning simply assumes that the agent follows an optimal policy. In realistic applications, SARSA will take into account, that the agent sometimes performs random actions, which might end in a very bad situation for it, like falling off a cliff, and adapt the policy according to it. We will use the SARSA Algorithm in the experiments in section 3.

Rewards in a computational framework, like the one of reinforcement learning, are typically only a numerical value and are different from the definitions used in the psychology literature. The theory of operant conditioning by Skinner uses money, food, high grades at school, or similar things, to define rewards [28]. But, as we dig deeper into the literature of neuro-psychology and biologics, we can see that these rewards are often only rewarding because they result in the a release of several neurotransmitters, like dopamine [29], so the two meanings overlap somehow and open the door for some very interesting computational approaches for (re)defining the term motivation for computational systems, bringing it closer to the psychological concept of motivation and possibly guide life-time learning, similar to humans. It is a common approach to implement a module in a system that supervises the value of one or more internal variables representing the motivation of the system and sending signals, like sensorimotor values, to the other parts of it. Examples for this can be found in [30], [31], [32] and [33].

Being only a numerical value, several motivational variables can be used together in a single architecture [34]. This gives credit to biological systems being pushed in certain directions by multiple overlapping motivational factors. These (possibly conflicting) motivational factors can either be compared directly because of their numerical characteristics, or can be associated with a numerical weight. Such a system has been explained by Konidaris and Barto [33]. The computational lecture gives several distinctions for such motivational systems, which could possibly create the reward for machine learning algorithms and we will learn about these distinctions in the following subsections, analogously to Oudeyer and Kaplan [7], based on psychological distinctions, and including a definition of intrinsic motivation.

1.2.2 Internal and External

The distinction between internal and external motivations only defines the functional location of the mechanism that computes or generates the reward. In case the reward is produces by a system located outside the learning agent, it is called external. An example for this is a classic hand-crafted reward for the agent, designed by a human. On the other side, a reward computed by a system located inside the agent would be called internal. Imagine an agent being motivated to keep his internal energy level as satiated as possible.

1.2.3 Intrinsic and Extrinsic

Even though we have seen a broad definition for those types of motivation in psychology literature earlier in this chapter, they are actually really vague for computer scientists and do not lead to an unique computational model. This is a result of two conceptual problems, which is first the broad range of psychological definitions for terms like fun or challenge, which are crucial for defining intrinsic motivation, and secondly the ambiguity of distinctions between intrinsic and extrinsic. We have already shown the existence of motivations, which are both. We have also shown that mixing this distinction and the one between internal and external is wrong. The best idea seems not to be an unique definition, but rather a wider range of different motivational systems to define intrinsic motivation in a computational way. This corespondents well to human behavior, which is not the result of a single motivational system, but many different systems [35]. A typology of computational approaches will be shown in the next chapter of this thesis.

1.2.4 Homeostatic and Heterostatic

Most types of motivation found in robots, and probably the animal kingdom as well, are homeostatic motivational systems trying to achieve a comfort level of satiation of the motivational variables [31]. On the other side, “heterostatic motivation systems constantly try to disturb this comfort zone”[7]. Homeostatic motivations are systems, which try to compensate the effect of perturbations, internal or external, and heterostatic motivations are systems that try to (self-)perturbate the organism out of its equilibrium [7].

1.2.5 Fixed and Adaptive

A fixed motivational system will always provide the same reward for the same state or sensorimotor input. This reward does not change through the agents lifetime, while an adaptive motivational system will not necessarily provide the same reward for the same state, every time it is reached. An example for a fixed motivational system is an agent which gets rewarded every time it observes novel situations, while the system would be adaptive, in case the agent is able to remember already seen novel situations. Given these distinctions, we will explore the already existing models for intrinsic motivated systems in the next chapter, before we get to the actual experiment, where we will see the advantage of using such models over using classic reinforcement learning.

2 Related Work

Former work by Barto, Singh and Lewis, based on an evolutionary perspective, showed the difficulties of distinctions between extrinsic and intrinsic rewards in computational frameworks. They elucidated the need to define suitable reward functions to realize intrinsic motivation in [36] and suggested to define intrinsic signals in RL as “primary reward signals, hard wired from the start of the agent’s life” [37]. This is often realized by explicitly modeling the internal state of an agent, for example in [38]. A range of models of cognitive architectures including an intrinsic motivation system does already exist in the literature. In this chapter, we provide an overview of a topology of computational approaches of intrinsic motivation, analogue to Oudeyer and Kaplan [7], but focus on competence-based approaches, as they have not been studied much yet. We furthermore present already existing models for these approaches.

2.1 Knowledge-Based Models for Reinforcement Learning

Knowledge-based models are based on an agent being able to gain knowledge and make predictions about the world it explores. This can be applied to passive interaction where the agent just observes its environment, as well as to active interaction where the agent performs actions in it and observes the results. Singh, Barto and Chentanez built up such a model in [39]. They used an agent which gets intrinsically rewarded by events in a grid-world environment based on the concept of salience [40], which are visual or auditive signals, that the agent does not expect in his environment. The learning was guided by an extrinsic reward for completing multiple tasks with increasing difficulty levels, defined by an increasing amount of sub-tasks needed to perform the task. The agent was able to learn these tasks faster than using extrinsic rewards only. Stout, Konidaris and Barto gave another grid-world example for this salience-based intrinsic motivation in [41], where they present a method that is more adaptable to real robotic applications, using the option theory framework [42] and layered learning [43].

2.2 Competence-Based Models for Reinforcement Learning

The next major approach for computational models of intrinsic motivation is the one of competence-based models. It is based on an agent being able to build up a know-how module and measure it’s competence on achieving self-determined goals or tasks.

This approach is directly inspired by several psychological models, for example the theory of effectance [11], the theory of personal causation [18], the theory of competence and self-determination [19] and the theory of flow [20].

The agent needs to be able to measure the difficulty of a task, as well as its performance on it. A task can be any resulting state in the RL framework or any sensorimotor input. The know-how module $KH(t_g)$ is responsible for planing actions in order to reach a goal g_k, which is self-determined by the agent, for a given time-step t_g. The next module is a motivation module which will reward the agent based on the performance of $KH(t_g)$. After reaching a goal state or a timeout, the performance is measured by comparing the reached state $g(t_g)$ with the initial goal $g^*(t_g)$.

$$l(g_k, t_g) = ||g^*(t_g) - g(t_g)||$$

This represents the level of (mis-)achievement for a given goal g_k and time-step t_g, and will be the basis for computing an intrinsic and internal reward and by this defining the level of interestingness of this goal. The third and last module is responsible for choosing goals which will provide the maximal rewards. This can be done with classic RL approaches. As we will focus on competence-based approaches in the experiments, we will give a more detailed overview about the sub-approaches of this section, based on [7].

2.2.1 Maximizing Incompetence Motivation (IM)

The IM approach does push the agent towards goals for which it performs worst. This is equivalent to an approach motivating for maximally difficult challenges. The resulting computational equation would be:

$$r(SM(\rightarrow t), g_k, t_g) = C\langle l(g_k^{\sigma_g}, t_g)\rangle$$

where $\langle l(g_k^{\sigma_g}, t_g)\rangle$ represents the mean performances trying to reach goals $g_k^{\sigma_g}$ over a fixed amount τ of episodes.
$\sigma_g > (g_k, g_k^{\sigma_g})$ denotes a distance function, giving similar goals the same level of interestingness.

2.2.2 Competence Progress Motivation (CPM)

To stick with psychological models of optimal challenge and flow[20], the difficulty of a goal can be modeled by computing the mean performance in trying to achieve this goal. This can be expressed by the competence progress being experienced by the agent, as it repeatedly tries to achieve it. The resulting equation is:

$$r(SM(\rightarrow t), g_k, t_g) = C(\langle l(g_k^{\sigma_g}, t_g - \theta)\rangle - \langle l(g_k^{\sigma_g}, t_g - \theta)\rangle)$$

where $\langle l(g_k^{\sigma_g}, t_g - \theta)\rangle$ denotes the mean performance in trying to reach goal g_k between episodes $t_g - \theta - \tau$ and $t_g - \theta$. An example for competence progress motivation is given by Stout and Barto in [44]. They use this concept to decide, which skill to learn and improve on at a given moment, drive continued learning behavior and gain a broad competence of skills. An internal motivation system takes the place of the external signals and chooses which skill to pursue, motivated by improvements in competence. The skills are defined as in the options formalism [42], and the agent is given a set of desired skills, expressed as subgoal states for which it must learn policies to achieve. The subgoals are hand-selected. They “desire to achieve an optimal exploration for skill learning, rather than an optimal balance between exploration and exploitation, as in traditional RL” [44] and achieve to outperform a naive agent, focusing on learning a skill for which progress can be made while ignoring those skills that are already learned or are at the moment to difficult.

2.2.3 Maximizing Competence Motivation (CM)

The CM approach would push the agent towards activities it already performs well on. The equation

$$r(SM(\rightarrow t), g_k, t_g) = \frac{C}{\langle l(g_k^{\sigma_g}, t_g)^{R_n(g_k)}\rangle}$$

does denote this. R_n expresses the region of the goal space that g_k falls in. This can be defined, for example, by a simple threshold σ_g for the distance from g_k:

$$R_n(g_k) = \{g_l | dist(g_l, g_k) < \sigma_g\}$$

2.3 Morphological Models for Reinforcement Learning

Morphological models are not based on measures between a cognitive learning system and the incoming sensorimotor values like the approaches shown above. This third approach is based only on the mathematical or morphological properties of the flow of these sensorimotor input values. Calculating the conditional entropy for two inputs is an example for this.

While Oudeyer and Kaplan present these model-types as reward-producing systems for the RL framework, they simultaneously suggest to combine several motivational systems in a learning agent, especially in a real robot. Examples for this can be found in the work of Stout, Konidaris and Barto in [41] where the authors define a framework for developmental robot learning which consists of using intra-option learning methods [42] for hierarchical RL [45], intrinsic motivation for task exploration and layered learning [43] to build an approach adaptable to real robotic applications. Gabriel, Akrour, Peters and Neumann advocated an application of RL to robotics in [46], using intrinsic motivation signals based of the entropy of the outcomes of the current policy, and the concept of empowerment, which is to maximize the entropy of the future. Their work was evaluated on a planar reaching task and a simulated robot table tennis task. Their algorithm is able to learn a diverse set of behaviors within the area of interest of a given task. Santucci, Baldassare and Mirolly built up the "Goal-Discovering Robotic Architecture for Intrinsic-Motivation (GRAIL)" in [47] which exploits the power of goals and competence-based intrinsic motivation to autonomously explore the world and learn different skills that allow the robot to modify the environment which was implemented in a simulated iCub robot and tested on four different experimental scenarios with reaching tasks. Baranes and Oudeyer presented "Robust Intelligent Adaptive Curiosity (RIAC)" in [48], an intrinsically motivated active learning algorithm, particularly suited for learning forward models in unprepared sensorimotor spaces and allowing a robot to self-organize developmental trajectories of increasing complexity. They also introduced the "Self-Adaptive Goal Generation Algorithm (SAGG)" in [49], an intrinsically motivated goal exploration mechanism for motor learning of inverse models. The combined these two approaches to the "“Self-Adaptive Goal Generation Robust Intelligent Adaptive Curiosity (SAGG-RIAC)" in [50], evaluated in three different robotic setups, and achieved statistically significantly superior performance.

While many of these approaches focus on solving the optimal exploration problem, in which the objective is to learn how to maximize return without necessarily accumulating high reward in the process [38], we focus on solving the trade-off between exploration and exploitation in an optimal way, which defines the optimal learning problem [38]. The next chapter describes our approaches of using intrinsic motivation for reinforcement learning to solve MDPs, before we evaluate our models in comparison with the classic RL algorithm SARSA.

3 Experiments and Results

This chapter will first give an overview of the observed behavior using SARSA to solve an MDP. We then present four limitations of this classic approach. We implement four scenarios exploiting these limitations and evaluate the performance of SARSA on these scenarios. Second, we present our own interpretation of competence-based reinforcement learning in form of three approaches, implemented to extend the classic SARSA algorithm, and also evaluated on the four scenarios. We evaluate our models in comparison with the performance of the classic SARSA algorithm, as well as in comparison with another approach called time-decreasing epsilon (TDE). To evaluate the performances we use heat-maps to track the agent's position on the grid-world, classic graphs to track numerical values, like the cumulative reward, and we visualize the agent's current policy for the grid-world domain using a grid-world with arrows and letters representing the action with highest state-action-function value.

3.1 Experimental Setups

The experiments in this chapter are evaluated on simple two-dimensional grid-worlds and hold the Markov assumption [24]. Figure 3.1 shows the grid-world for the first part of our experiment. The agent receives a reward of +10 for reaching the target position. It receives a reward of −1 for trying to move out of the grid-world or against a wall. In both cases the agent would not move to a different state.

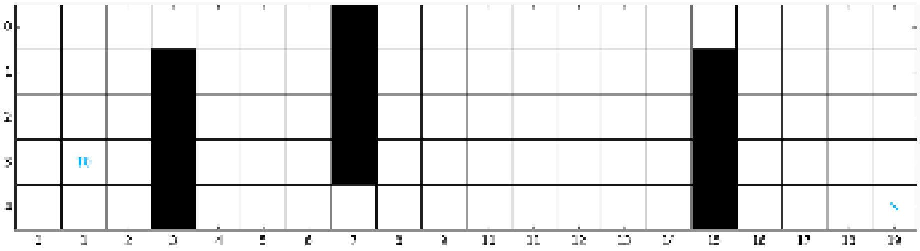


Figure 3.1.: The First Grid-World

This two-dimensional world is containing a reachable area colored in white, providing zero reward, and some walls colored in black, providing a reward of −1, when the agent tries to move onto their position. It is furthermore restricted by its own borders, also providing a reward of $r = -1$. The position of the target, which provides a reward of +10 for the RL algorithm, is denoted with $T0$, while the agent's starting position is denoted with S

The RL agent's state is representing its two-dimensional position $[x, y] : x \in [0, 5), y \in [0, 20)$ on this grid-world. It is capable of 9 king-movement actions, which are explained in Figure 3.2. We evaluate the performance of SARSA on this simple grid-world in the next section of this thesis.

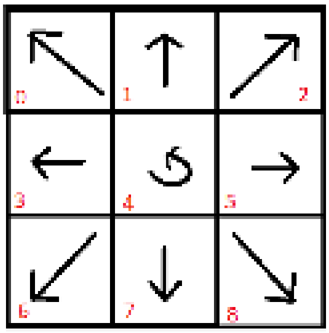


Figure 3.2.: The King-Movement Actions

The king-movement actions represent actions in 8 cardinal directions and one "do nothing/wait" action.

3.2 The Performance of SARSA

At first, we evaluate the classical RL algorithm SARSA on our first grid-world shown in Figure 3.1. To get rid of spikes by random exploitation and reduce memory usage, we limit the horizon to 5000 possible steps per episode. An episode otherwise terminates as soon as the agent reaches the terminal target position. The actions are drawn from an epsilon-greed policy choosing a random action with probability ϵ and an action with maximum state-action-function value otherwise.

```
Data: $\epsilon \in [0, ..., 1]$, $s \in S$, $Q(s, a)$
choose random number $r_n \in [0, ..., 1]$
if $r_n \leq \epsilon$ then
    return random action $a_{random}$
else
    return action $a := max_a Q(s, a)$
```

Algorithm 3: Epsilon-greedy Policy Pseudo-Code

We first evaluate the influence of the SARSA algorithm's parameters on the performance on our grid-world. We evaluate ϵ in a range of [0.0001, 0.001, 0.01, 0.05, 0.1, 0.5], the learning rate α in a range of [0.0, 0.25, 0.5, 0.75, 1.0] and the discount factor γ in the range of [0.0, 0.25, 0.5, 0.75, 1.0] as well. All sets of this parameters are averaged over 20 evaluations a 100 episodes of learning. The evaluation results, shown in Figure A.1, give us the best performance for $\epsilon = 0.0001$ while α and γ seem to not influence the result, unless they are set higher than zero. We expect the algorithm to build up the agents state-action-function towards an optimal path through the grid-world for reaching the rewarding target position, respectively the terminal state, and we expect the resulting state-action-function to define actions that lead to this position with minimal loss. The amount of time-steps needed to reach the terminal state and the cumulative reward for each episode of learning are used as an indicator for the agent's skill, learning process and performance of the algorithm.

Figure 3.3 shows the results of running one SARSA evaluation on our grid-world with the parameters set to $\epsilon = 0.0001$ where we take the best of our evaluation above, and set α and γ to 0.5 as the mean value of their value range. More details for this evaluation can be found in Figure A.2. The next section gives an overview of some scenarios where we expect the classic SARSA algorithm to fail. We then evaluate SARSA on these scenarios.

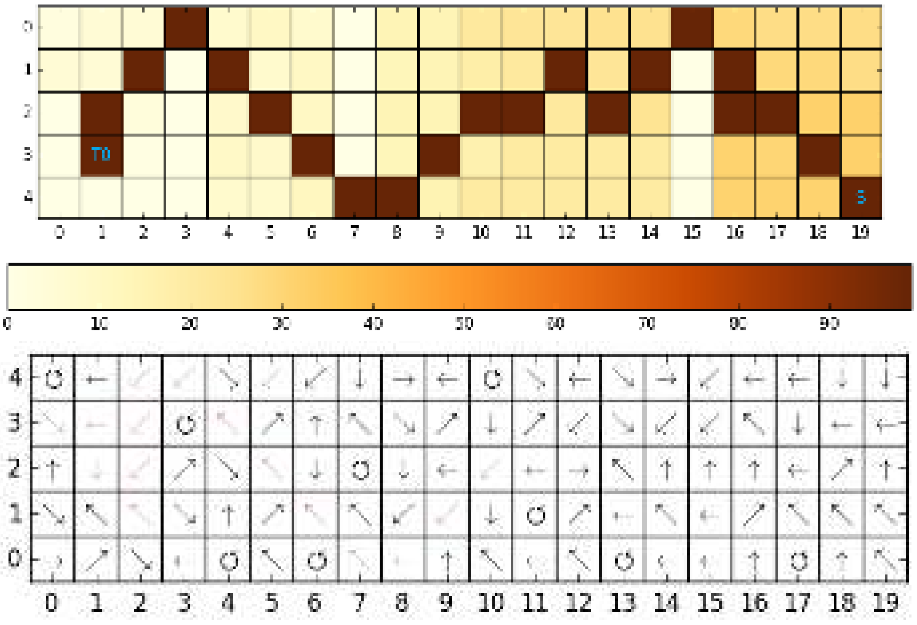


Figure 3.3.: SARSA Evaluation on First Grid-World

One Evaluation of SARSA on the first grid-world with the best set of parameters found in the previous parameter evaluations: $\epsilon = 0.0001, \alpha = 0.5, \gamma = 0.5$. The upper figure shows the heat-map of the agent's position over episodes: The darker the colors are, the later the episode the state has been visited last. E.g. a white colored state has been visited last in episode 0 or never, while a dark-brown colored state has been visited last in episode 90+. The lower figure shows an arrow for the action with highest state-action-function value for every state. Actions with a state-action-function value higher than 0.01 are displayed in red. One can see the agent is following a clear path through the grid-world, resulting from his state-action-function, leading him towards the goal state.

3.3 Limitations of SARSA

Several limitations of SARSA and different approaches to overcome these limitations can be found in the literature. SARSA tends to get stuck in local maxima, and not to be flexible enough for changing environments. Furthermore, the nature of using a state-action-function and the epsilon-greedy policy prevents the agent from observing negative rewards even with a much higher reward in sight. We design four scenarios exploiting these limitations and evaluate the performance of classic SARSA on these scenarios.

3.3.1 The 1st Scenario: Multiple goals

SARSA has troubles with multiple goals, when one goal is easy to achieve, while the other one is hard to reach, whether it provides a much higher reward, or not. Such a scenario is shown in Figure 3.4.

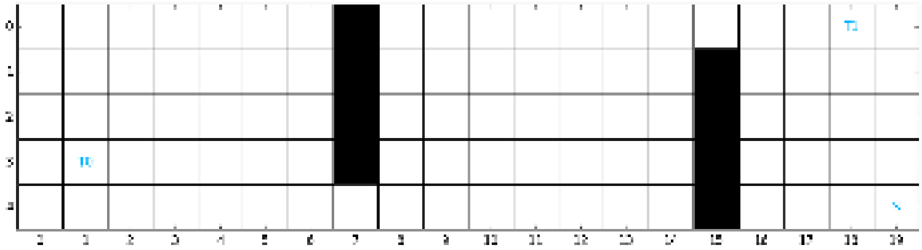


Figure 3.4.: Scenario 1: Two Terminal States

The target $T0$ provides a reward of $r = +100$, while target $T1$ gives a reward of $r = +10$. Apart from that apply the same definitions as in Figure 3.1.

Evaluating the SARSA parameters on this grid-world (see Figure A.3) indicates that they do not seem to have much influence unless ϵ is set very high (≥ 0.1) which increases the time needed to terminate but does not result in a higher reward. α and γ are seemingly not influencing the performance unless they are set to 0.0 which prevents the agent from useful learning. The results of one evaluation with the parameters set to $\epsilon = 0.01$, to still have some random exploration, $\alpha = 0.5$ and $\gamma = 0.5$, as the mean value of their range, are shown in Figure 3.5. More details can be found in Figure A.4.

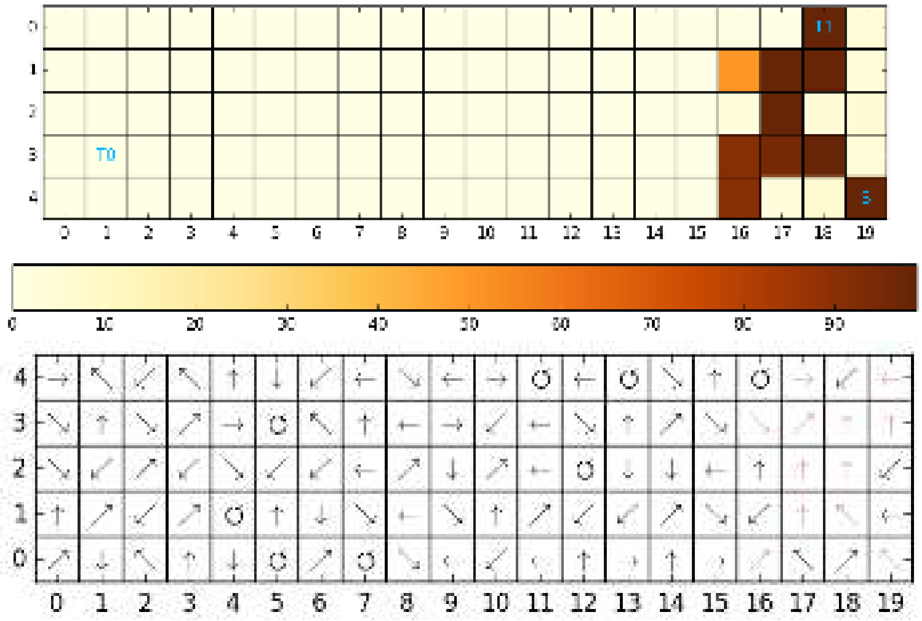


Figure 3.5.: SARSA Evaluation on Scenario 1

Evaluation of SARSA on the first scenario with the parameters set to $\epsilon = 0.01$, $\alpha = 0.5$ and $\gamma = 0.5$. The same definitions apply as in Figure 3.3. One can see, the agent is not learning to move towards $T0$, providing the much higher reward. Independent of the parameters, the agent always learns a state-action-function pushing him towards $T1$.

3.3.2 The 2nd scenario: Overcoming negative rewards to reach the target position

The next scenario is a world with a highly positive rewarding target, reachable only by overcoming multiple smaller negative targets. We expect an agent following the SARSA algorithm to get "scared" of the negative rewards and not explore enough to reach the target and build up a useful policy. This effect might be even stronger, if a second, but smaller, positive reward is positioned closer to the agent without the need to pass negative rewards (see previous subsection). A grid-world implementing this scenario is displayed in Figure 3.6.

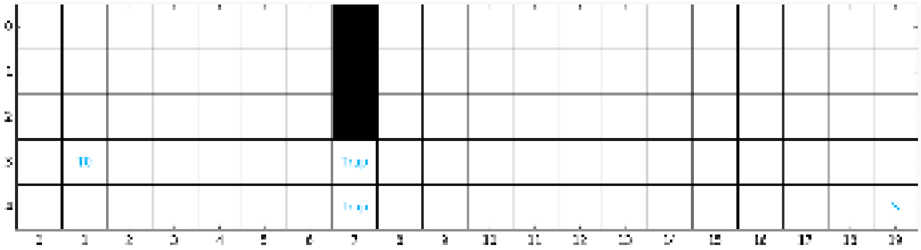


Figure 3.6.: Scenario 2: Traps

Same definitions as in Figure 3.1. The target T0 provides a reward of $r = +100$. The traps give a reward of $r = -1$, just like the walls, but can be passed.

We now extend the horizon to 200 episodes of learning per evaluation because SARSA often already performs very bad below 100 episodes of learning for this task. Evaluating the SARSA parameters (see Figure A.5 for some insights), indicates the use of a high γ, while using a very high rate of random exploration ($\epsilon \geq 0.05$). Figure 3.7 displays the results of running one evaluation of SARSA with the parameters set to $\epsilon = 0.1, \alpha = 0.5, \gamma = 1.0$. The results of evaluating the SARSA parameters can be found in Figure A.5. More details for this evaluation can be found in A.6. The agent is able to build a useful policy pushing it towards the target, but requires a high percentage of random exploration to do so, which often results in receiving negative reward by the borders and wall of the grid-world.

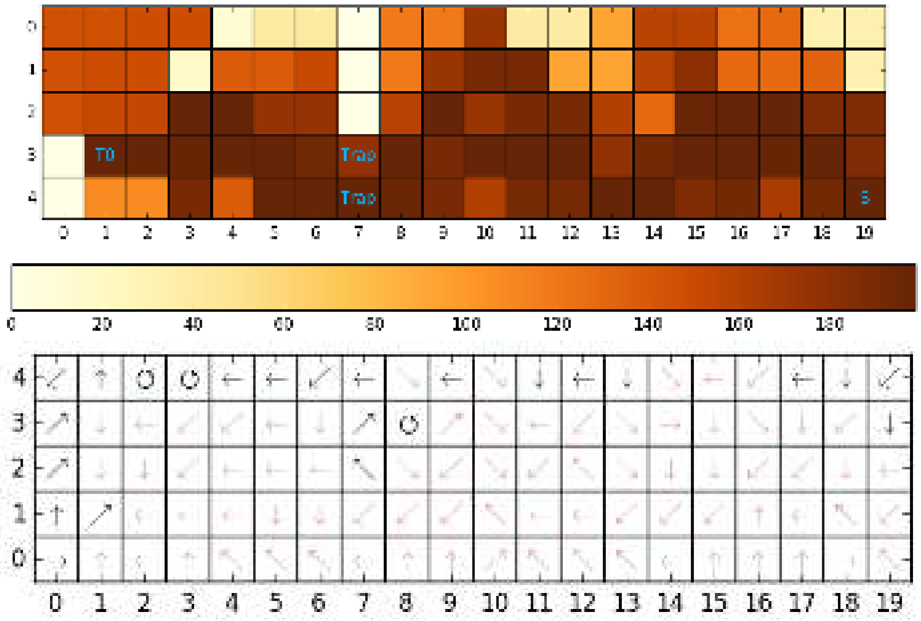


Figure 3.7.: SARSA Evaluation on Scenario 2

One evaluation of SARSA on the second scenario with the parameters set to $\epsilon = 0.1, \alpha = 0.5, \gamma = 0.9$ and 200 learning episodes. The same definitions as in Figure 3.3 apply. One can see the high amount of random exploration in the upper plot, but the lower plot shows the agent being able to build a strong policy leading it towards the target.

3.3.3 The 3rd Scenario: Changing Terminal States / Changing Environment

Thirdly, we think SARSA will perform badly on a changing grid-world, for example with targets that are repositioned after being visited a couple of times. To perform well, the agent needs to throw away an already existing policy and come up with a new one really fast. We expect it not to be that flexible and perform badly on this scenario. Again, the performance of SARSA is evaluated on the grid-world shown in Figure 3.1, this time the target gets repositioned randomly after being visited 10 times. The results of evaluating SARSA's parameters is shown in Figure A.7, and Figure 3.8 displays the result of one evaluation with the parameters set to $\epsilon = 0.01, \alpha = 1.0, \gamma = 0.0$. More details for this evaluation can be found in Figure A.8. One can see that the agent is able to reposition the target 19 times, which is the theoretical optimum.

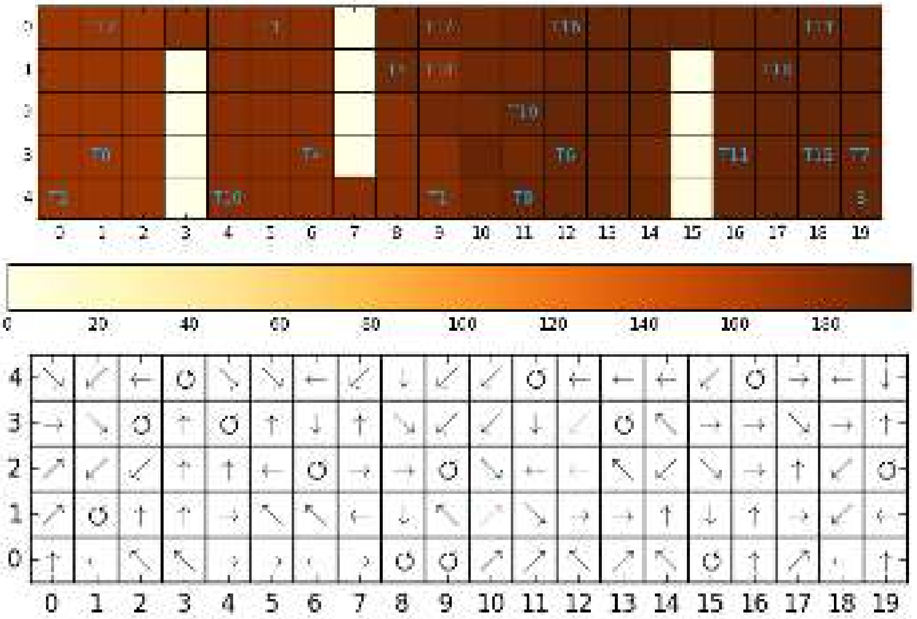


Figure 3.8.: SARSA Evaluation on Scenario 3

One evaluation of SARSA on the scenario 3 with $\epsilon = 0.01, \alpha = 1.0, \gamma = 0.0$ and 200 learning episodes. Same definitions as in Figure 3.3 apply. The target positions are relocated randomly after being visited 5 times. The upper plot shows, the target is repositioned 19 times. The agent only learns a very shortsighted state-action-function, which is, paired with a decent amount of random exploration ($\epsilon = 0.01$), required to perform well on this scenario.

3.3.4 The 4th scenario: Multiple Goals and Hierarchies of Actions

The most common methods to solve MDPs, Q-Learning and SARSA, face performance problems for complex actions and terminal states which may require a hierarchy of actions to reach them. These methods get outperformed significantly by SMDP methods [42], which are able to build up a hierarchy of actions, and therefore allow their agents to deal with more difficult and complex scenarios which are often build to be more realistic than simple grid-worlds with corresponding reward-matrices depending on the agents position. One attribute of these scenarios can be multiple goals with the same or even no (immediate) reward. We use a grid-world with more complex actions and terminal conditions in section 3.4 to evaluate SARSA on such a scenario. However, SMDP methods are out of the scope of our experiments.

3.3.5 Time-Decreasing Epsilon (TDE)

One approach to deal with the exploration/exploitation trade-off and extend the classic SARSA algorithm is to use an epsilon which decreases over time. We use this approach to compare it with the performance of classic SARSA and our approaches for intrinsic motivation which are explained in the next section. For our experiments we start off with a high amount of exploration $\epsilon = 0.5$ for the first fourth of the learning episodes and then decrease ϵ exponentially. The trend of epsilon is shown in Figure A.9.

While we are not able to achieve a better performance on scenario 1 with this approach, and an even worse performance on scenario 3, TDE is able to outdo the classic SARSA approach in scenario 2. Evaluating the parameters (Figure A.10) gives the best performance for $\alpha = 0.5$ and $\gamma = 1.0$. The results of one evaluation are shown in Figure 3.9, more details can be found in Figure A.11. The TDE approach outperforms SARSA and results in a more stable learning due to the reduced amount of random exploration in later episodes.

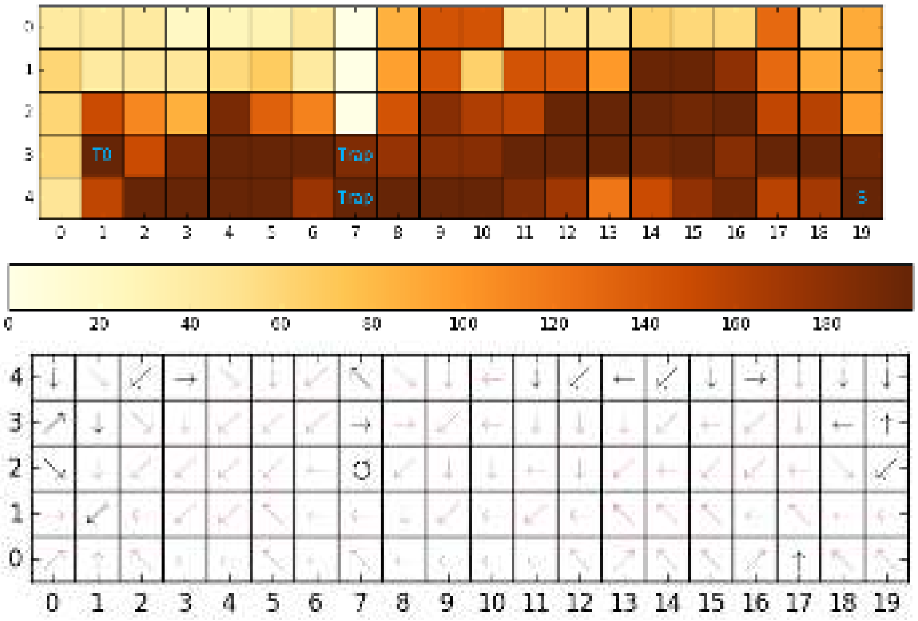


Figure 3.9.: TDE Evaluation on Scenario 2

One evaluation of TDE on scenario 2 with $\alpha = 0.5$, $\gamma = 1.0$ and 200 learning episodes. Same definitions as in Figure 3.3 apply. The upper plot shows a less amount of random exploration, while the lower plot shows that the agent is able to build a strong state-action-function, just like with the classic SARSA algorithm.

3.4 Models for Intrinsic Motivation Extended SARSA

In this section we present our models for intrinsic motivation to improve learning and evaluate their performance on the grid-worlds shown in the previous section. We use notations similar to Oudeyer and Kaplan [7]. The RL agent receives an intrinsic reward r_i when reaching a terminal state which is added onto the reward observed by interacting with the environment, denoted with r_e. The sum $r = r_e + r_i$ is used to update the state-action-function in the SARSA algorithm (see chapter 1). The intrinsic reward r_i always depends on the level of misachievement for a reached terminal state which is defined by $l_g = ||p_g^* - p_g||$, where p_g^* is the optimal performance for a given terminal state g. For our approaches we decide to define the performance by the number of time-steps / primitive actions needed to reach the goal state / terminal state. Therefore the optimal performance p_g^* is given by the minimum of time-steps / actions needed to reach it. These are already defined for each terminal-state $g \in S$ and the agent does not need to discover them. p_g denotes the agent's current performance in the last learning-episode while trying to reach the terminal-state g. One could think of other possible definitions for (optimal) performance, e.g. the accumulated sum of observed rewards while trying to reach a goal state.

We now present three different approaches to map this level of misachievement l_g into an intrinsic reward r_i, based on psychological definitions for motivation, previous work in this research section and our own interpretations. We evaluate our models on the scenario 1-3 shown above with 200 episodes of learning, as they sometimes do not converge before episode 80.

3.4.1 Incompetence Motivation (IM)

To drive the agent towards terminal states with a high level of misachievement l_g, the intrinsic reward r_i is defined by

```
Data: $l_g$, $c$
if $l_g = 0$ then
    $r_i = c$
else
    $r_i = c/l_g$
```

Algorithm 4: IM Model Pseudo-Code

where c is a negative constant. We choose $c = -100$ which is not changed throughout our experiments. This intrinsic reward r_i pushes the agent away from terminal states with a small level of misachievement l_g, which is our interpretation of incompetence motivation.

Evaluating our IM approach on scenario 1 (Figure 3.4), we observe the results displayed in Figure A.12 which indicate the best results for a set of $\epsilon = 0.01, \alpha = 0.75, \gamma = 1.0$. The visit history and resulting state-action-function of one evaluation of our IM approach with these parameters are shown in Figure 3.10, more details can be found in A.13. The negative intrinsic reward of the terminal state $T0$ grows, as the agent gets better in reaching it, resulting in a state-action-function guiding it away from $T1$ and towards $T0$, providing the higher external reward, at about 50 episodes of learning.

For our second scenario with traps (Figure 3.6), we find a best set of $\epsilon = 0.1, \alpha = 0.5, \gamma = 1.0$. The results of one evaluation with these parameter settings are displayed in Figure 3.11, while more details can be found in Figure A.15. The results of the SARSA parameter evaluations can be found in Figure A.14. The behavior does not differ much from the one using SARSA. The agent is able to learn reaching the terminal state with a high amount of random movement, in which it also risks observing a high amount of negative rewards.

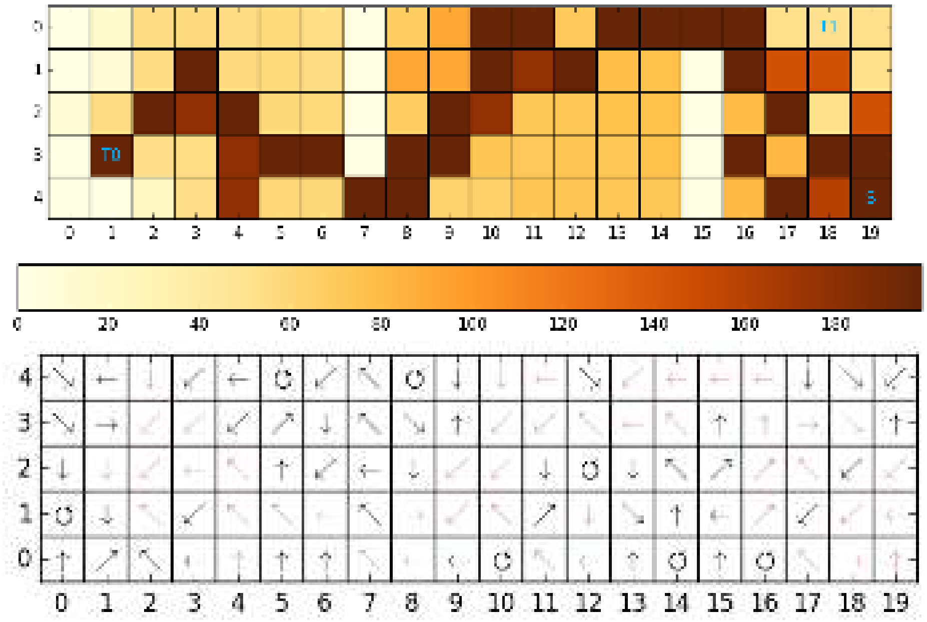


Figure 3.10.: IM Model Evaluation on Scenario 1

Evaluation a 200 learning episodes of our IM approach on the first scenario with two targets. Same definitions as in Figure 3.3 apply. Parameters are set to $\epsilon = 0.01, \alpha = 0.75, \gamma = 1.0$. The agent now learns a state-action-function pushing it towards $T0$, providing the higher external reward.

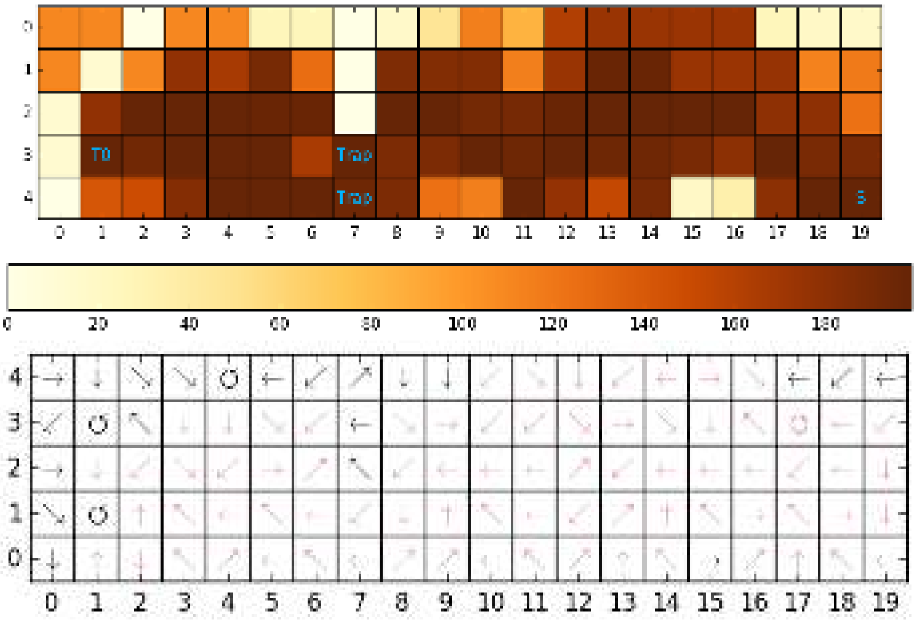


Figure 3.11.: IM Model Evaluation on Scenario 2

Evaluation a 200 learning episodes of our IM approach on the second scenario with traps. Same definitions as in Figure 3.3 apply. Parameters are set to $\epsilon = 0.1, \alpha = 0.5, \gamma = 1.0$. The behavior is very similar to the one of classic SARSA, requiring a high amount of random exploration, which results in a strong state-action-function, but also in a high amount of observed negative rewards.

Thirdly, we evaluate the SARSA parameters for our IM approach on the scenario 3 (Figure 3.1) with the terminal state repositioning randomly after being visited 5 times. The results are shown in Figure A.16. The evaluations give us a best set of $\epsilon = 0.01, \alpha = 1.0, \gamma = 0.0$ which gives us the results shown in Figure 3.12 for one evaluation. More details can again be found in Figure A.17. The performance is as good as with classic SARSA.

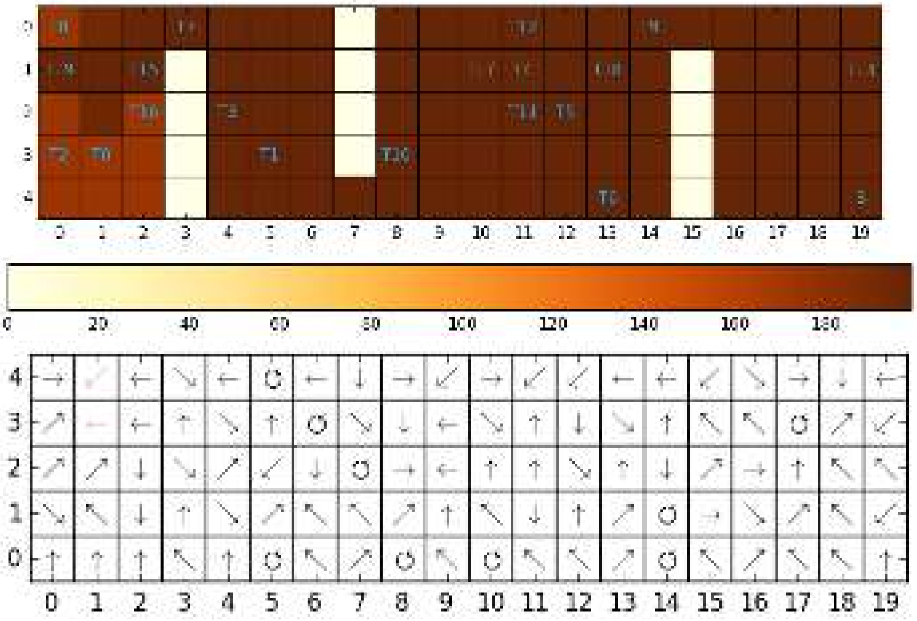


Figure 3.12.: IM Model Evaluation on Scenario 3

Evaluation a 200 learning episodes of our IM approach on the third scenario with randomly repositioned terminal states. Same definitions as in Figure 3.3 apply. Parameters are set to $\epsilon = 0.01, \alpha = 1.0, \gamma = 0.0$. The agent performs as good as with classic SARSA and manages to reposition the terminal state 19 times.

3.4.2 Competence Motivation (CM)

In contrast to IM, the intrinsic reward r_i for CM is defined by

```
Data: $l_g$, $c$
if $l_g = 0$ then
    $r_i = -c$
else
    $r_i = -c/l_g$
```

Algorithm 5: CM model pseudo-code

This method values terminal-states with a low level of misachievement even higher. Analogously to the IM model, we evaluate the CM approach on our three environments.

The results of evaluating the parameters for the first scenario with two goals are shown in Figure A.18. They give us a best set of $\epsilon = 0.01, \alpha = 0.5, \gamma = 0.25$ which gives the evaluation results shown in Figure 3.13, details are displayed in Figure A.19. As soon as the agent reaches the terminal state with increasing performance, the amount of exploration seems to decrease. So the agent learns to reach $T1$ with less variance and faster than with the classic SARSA approach.

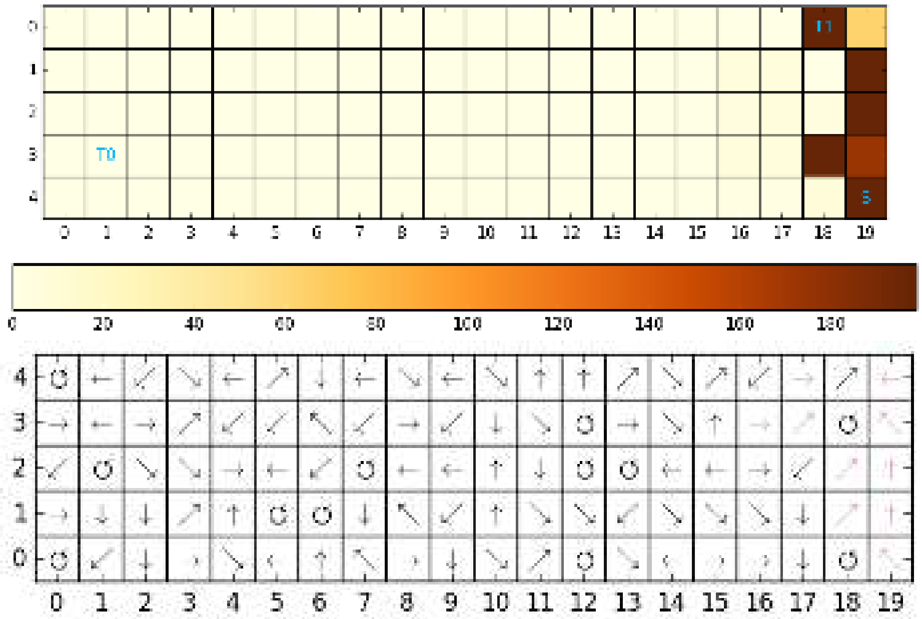


Figure 3.13.: CM Model Evaluation on Scenario 1

Evaluation a 200 learning episodes of our CM approach on the first scenario with two targets. Same definitions as in Figure 3.3 apply. Parameters are set to $\epsilon = 0.01, \alpha = 0.5, \gamma = 0.25$. The agent learns to reach $T1$, with very little variance on its path towards it.

The results of evaluating the SARSA parameters for the second scenario with traps are shown in Figure A.20. They give us a best set of $\epsilon = 0.1, \alpha = 0.5, \gamma = 1.0$ which gives the evaluation results shown in Figure 3.14, details are presented in Figure A.21. The CM approach is not able to perform better than SARSA, TDE or IM on this scenario.

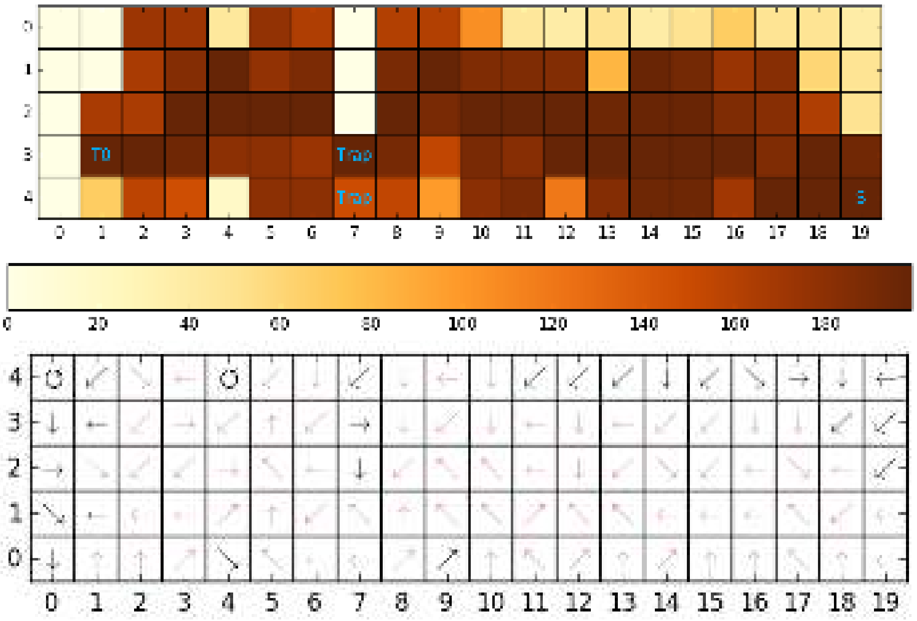


Figure 3.14.: CM Model Evaluation on Scenario 2

Evaluation a 200 learning episodes of our CM approach on the second scenario with traps. Same definitions as in Figure 3.3 apply. Parameters are set to $\epsilon = 0.1, \alpha = 0.5, \gamma = 1.0$. The behavior is very similar to the one of SARSA, TDE and IM, requiring a high amount of random exploration to reach the terminals state and learn strong state-action-function.

The results of evaluating the parameters for scenario 3 with relocating goals are shown in Figure A.22. They give us a best set of $\epsilon = 0.01, \alpha = 1.0, \gamma = 0.0$ which gives the evaluation results shown in Figure 3.15, details are displayed in Figure A.23. The agent performs as good as with SARSA and the IM approach.

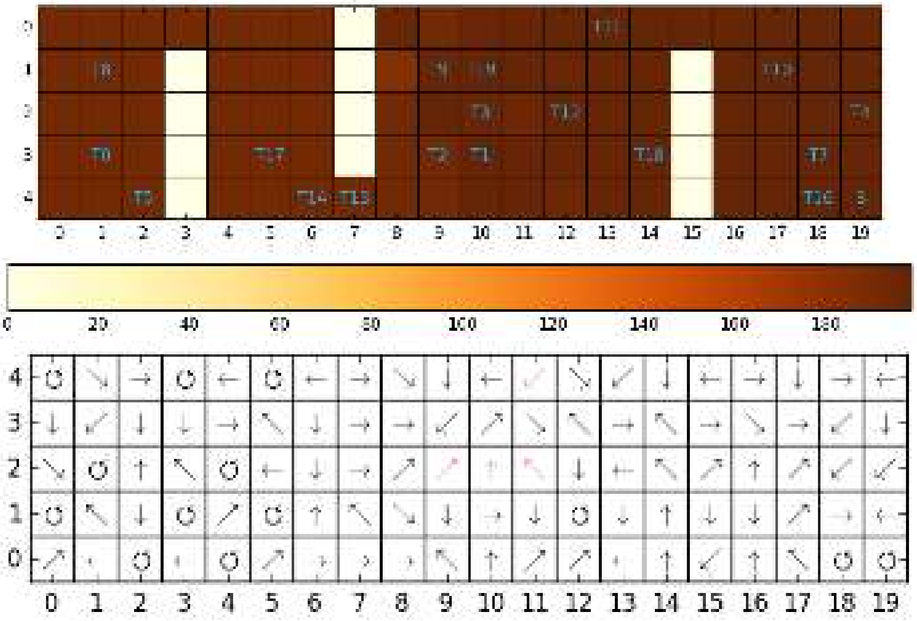


Figure 3.15.: CM Model Evaluation on Scenario 3

Evaluation a 200 learning episodes of our CM approach on the third scenario with randomly repositioning targets. Same definitions as in Figure 3.3 apply. Parameters are set to $\epsilon = 0.01, \alpha = 1.0, \gamma = 0.0$. The agent is again able to reposition the target 19 times and successfully learns a very shortsighted state-action-function, which is required to perform well on this scenario.

3.4.3 Competence Progress Motivation (CPM)

Thirdly, our approach for CPM defines r_i by

```
Data: $l_g, \overline{l_g}^{\theta}, c$
if $\overline{l_g}^{\theta} - l_g = 0$ then
    $r_i = c$
else
    $r_i = c / \overline{l_g}^{\theta} - l_g$
```

Algorithm 6: CPM model pseudo-code

where $\overline{l_g}^{\theta}$ denotes the mean level of misachievement for the last θ times trying to reach terminal state g. We choose $\theta = 10$ and stick with our choice for the rest of our experiments. Analogously to the IM and CM models, we evaluate the CPM approach on our three environments.

The results of evaluating the parameters for scenario 1 with two goals are shown in Figure A.24. They give us a best set of $\epsilon = 0.001, \alpha = 0.75, \gamma = 0.0$ which gives the evaluation results shown in Figure 3.16, details are displayed in Figure A.25. The details show that the agent learns to reach $T0$ and $T1$ very well but is interrupted by episodes of exploration when it makes no learning progress. This results in a relatively high amount of exploration.

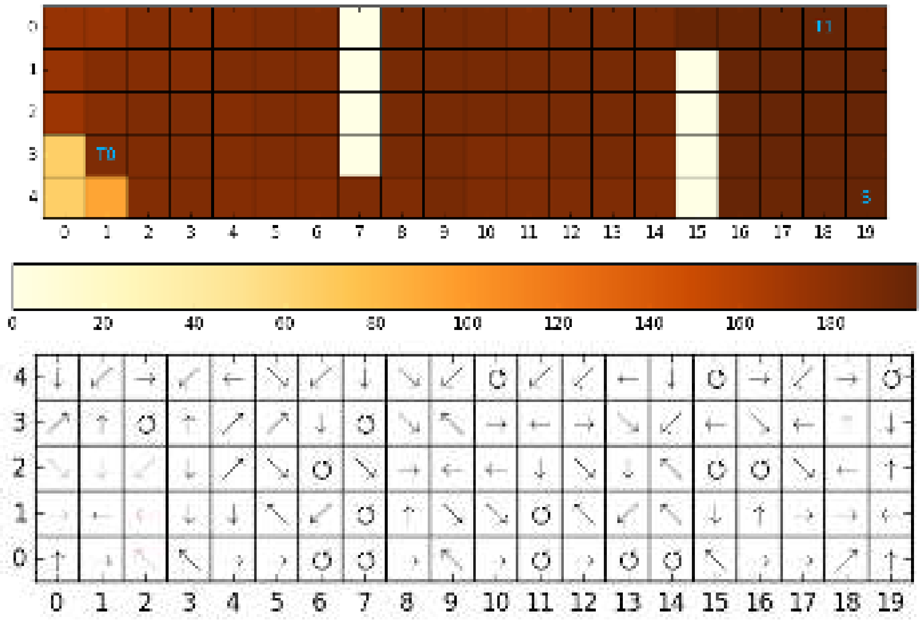


Figure 3.16.: CPM Model Evaluation on Scenario 1

Evaluation a 200 learning episodes of our CPM approach on the first scenario with two terminal states. Same definitions as in Figure 3.3 apply. The parameters are set to $\epsilon = 0.001, \alpha = 0.75, \gamma = 0.0$.

We can now see the agent exploring very much even in later episodes, but still building shortsighted state-action-functions for reaching $T0$ and $T1$.

With a set of $\epsilon = 0.001, \alpha = 0.75, \gamma = 0.5$ we observe the results shown in Figure 3.17 while more details can be found in Figure A.26. We can now see less exploration and a much stronger state-action-function pushing the agent towards $T0$.

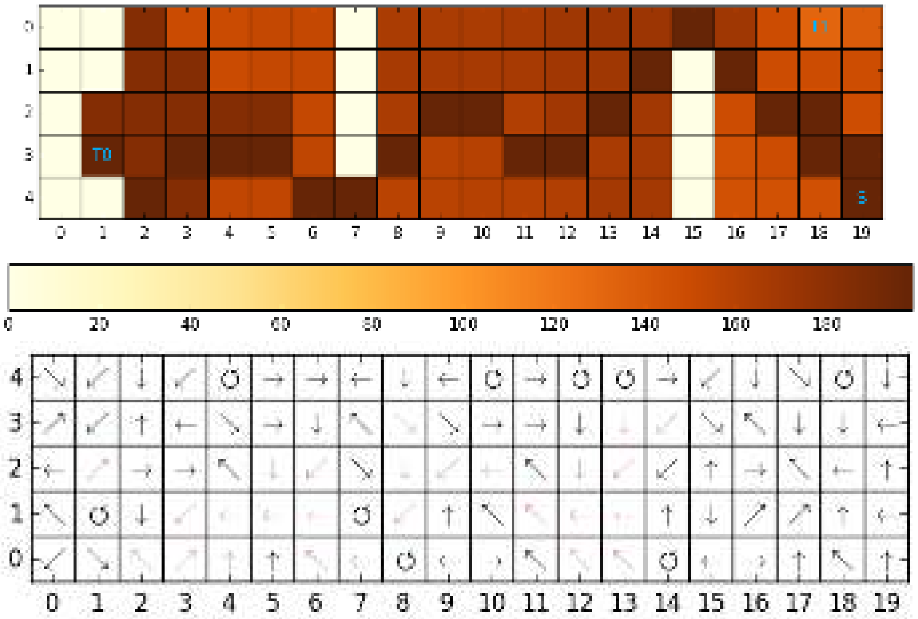


Figure 3.17.: CPM Model Evaluation 2 on Scenario 1

Another evaluation a 200 learning episodes of our CPM approach on the first scenario with two terminal states. Same definitions as in Figure 3.3 apply. Parameters are now set to $\epsilon = 0.001, \alpha = 0.75, \gamma = 0.5$. The upper plot now shows less random exploration and a clearer path through the grid-world. The lower plot shows the agent now learns a more farsighted state-action-function pushing it towards $T0$.

The results of evaluating the SARSA parameters for the scenario 2 with traps are shown in Figure A.27. They give us a best set of $\epsilon = 0.1, \alpha = 0.5, \gamma = 1.0$ which gives the evaluation results shown in Figure 3.18, details are displayed in Figure A.28. Again, the agent learns a state-action-function pushing it towards $T0$, but requiring a high amount of exploration. The behavior is very similar to the one in the previous approaches.

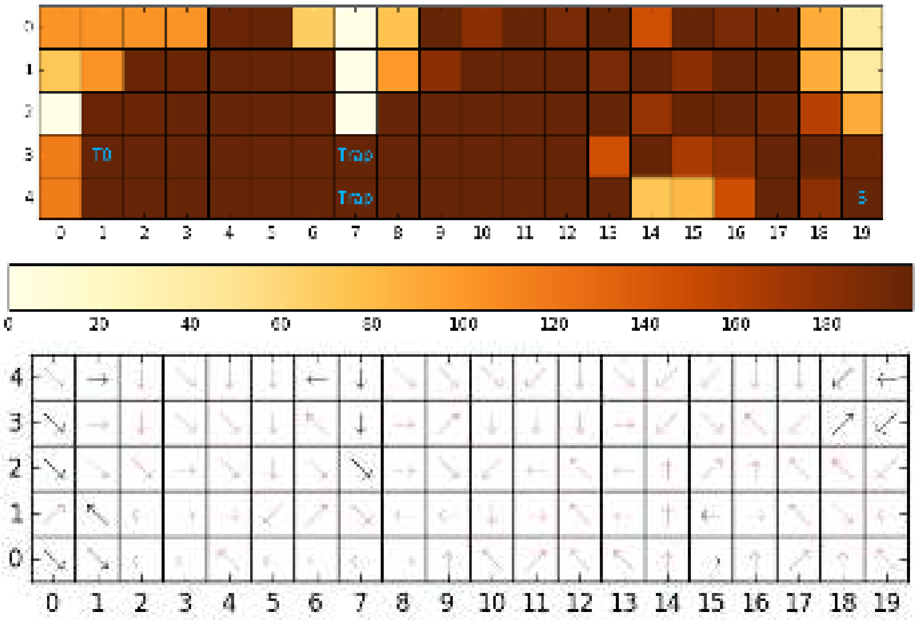


Figure 3.18.: CPM Model Evaluation on Scenario 2

Evaluation a 200 learning episodes of our CPM approach on second scenario with traps. Same definitions as in Figure 3.3 apply. Parameters are set to $\epsilon = 0.1, \alpha = 0.5, \gamma = 1.0$. The heat-map shows a high amount of random exploration, while the lower Figure displays a very farsighted state-action-function leading the agent towards $T0$. The behavior is very similar to the previous approaches.

The results of evaluating the parameters for scenario 3 with relocating goals are shown in Figure A.29. They give us a best set of $\epsilon = 0.01, \alpha = 1.0, \gamma = 0.0$ which gives the evaluation results shown in Figure 3.19, details are displayed in Figure A.30. The agent is again able to perform very well on it and reposition the target 19 times.

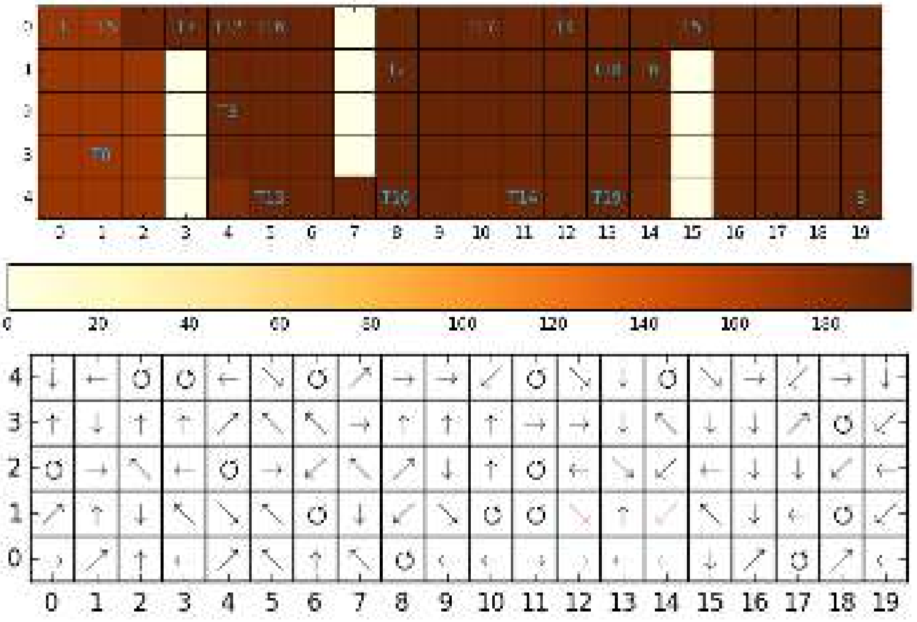


Figure 3.19.: CPM Model Evaluation on Scenario 3

Evaluation a 200 learning episodes of our CPM approach on the third scenario with targets being repositioned randomly after being visited 5 times. Parameters are set to $\epsilon = 0.01, \alpha = 1.0, \gamma = 0.0$. The agent is able to explore much, reposition the target 19 times and learn shortsighted state-action-functions, always pushing it towards the right target position.

3.5 Final Evaluations on the Market Domain

We now present another grid-world domain, the market, shown in Figure 3.20. This grid-world implements scenario 4 by using a five dimensional state, instead of 2 dimension, and the agent also needs to learn a hierarchy of actions to reach the terminal state, instead of just moving onto it. It also implements scenario 1, by using 3 different goal-states with increasing reward they provide, as well as an increasing amount of time-steps needed to reach them.

Positioned in the world are three fruits: An apple, an orange and a banana. In addition, three merchants: An apple merchant, an orange merchant and a banana merchant. The agent is now capable of 10 actions, the nine king-movement actions presented in Figure 3.2, and an action to pick up a fruit. It is now holding a five-dimensional state $[x, y, a, o, b] : x \in [0, ..., 4], y \in [0, ..., 19], a \in [0, 1], o \in [0, 1], b \in [0, 1]$ where the new three dimension track whether the agent picked up a fruit, or not.

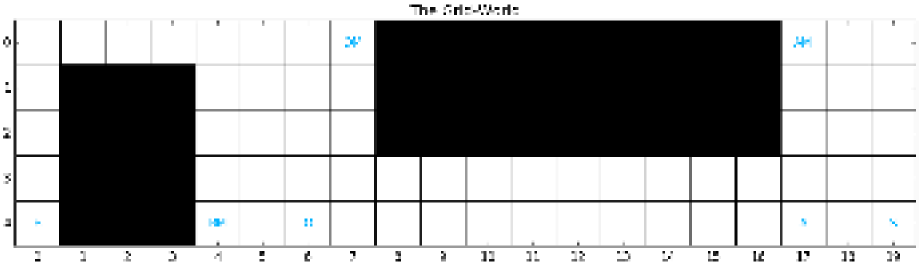


Figure 3.20.: Scenario 4: The Market Domain

This two-dimensional world is containing a reachable area colored in white and some walls colored in black. It is furthermore restricted by its own borders. The three fruits are denoted with A, O, B and their corresponding merchants with AM, OM, BM. Moving onto A, performing the pick action and moving to AM provides a reward of +10. Analogously +50 for moving O to OM and +100 for moving B to BM.

To reach a rewarding terminal state, the agent needs to pick up a fruit and move to the corresponding merchant. It receives a reward of +10 for bringing the apple A to the apple merchant AM, a reward of +50 for bringing the orange O to the orange merchant OM and a reward of +100 for picking up the banana B and delivering it to the banana merchant BM. We use this domain to evaluate the performance of SARSA, TDE and our models for intrinsic motivation on a more complex world with more complex actions required to reach a terminal state. To consider the higher complexity of this domain, we use 300 episodes of learning.

3.5.1 SARSA

We first evaluate the classic SARSA algorithm's parameters, to find the best possible performance of it on our grid world in Figure 3.20. We observe a set of $\epsilon = 0.05, \alpha = 0.75, \gamma = 0.75$. The results are displayed in Figure A.31, and the evaluation results can be found in Figure 3.21. More details can be seen in Figure A.32.

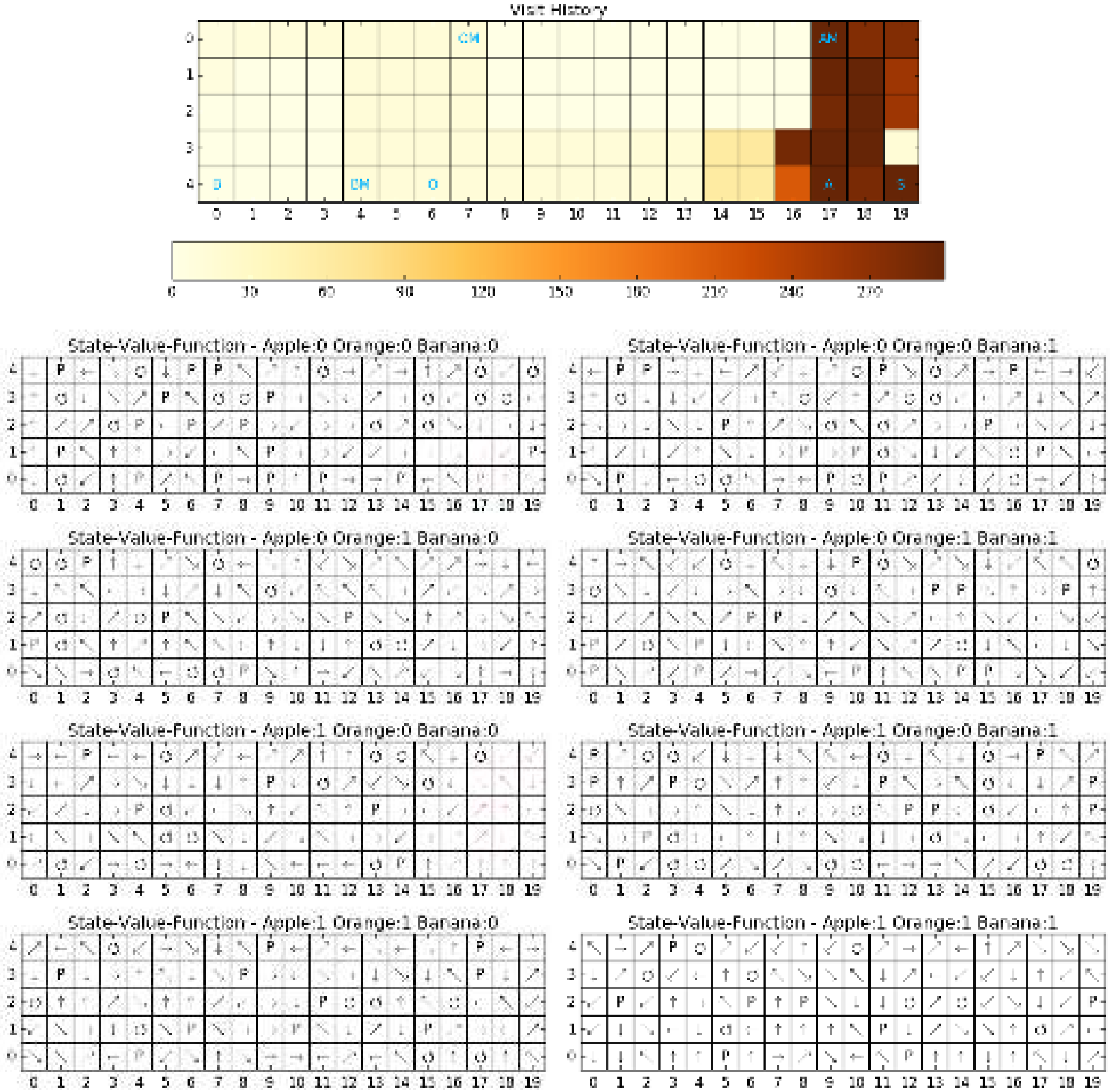
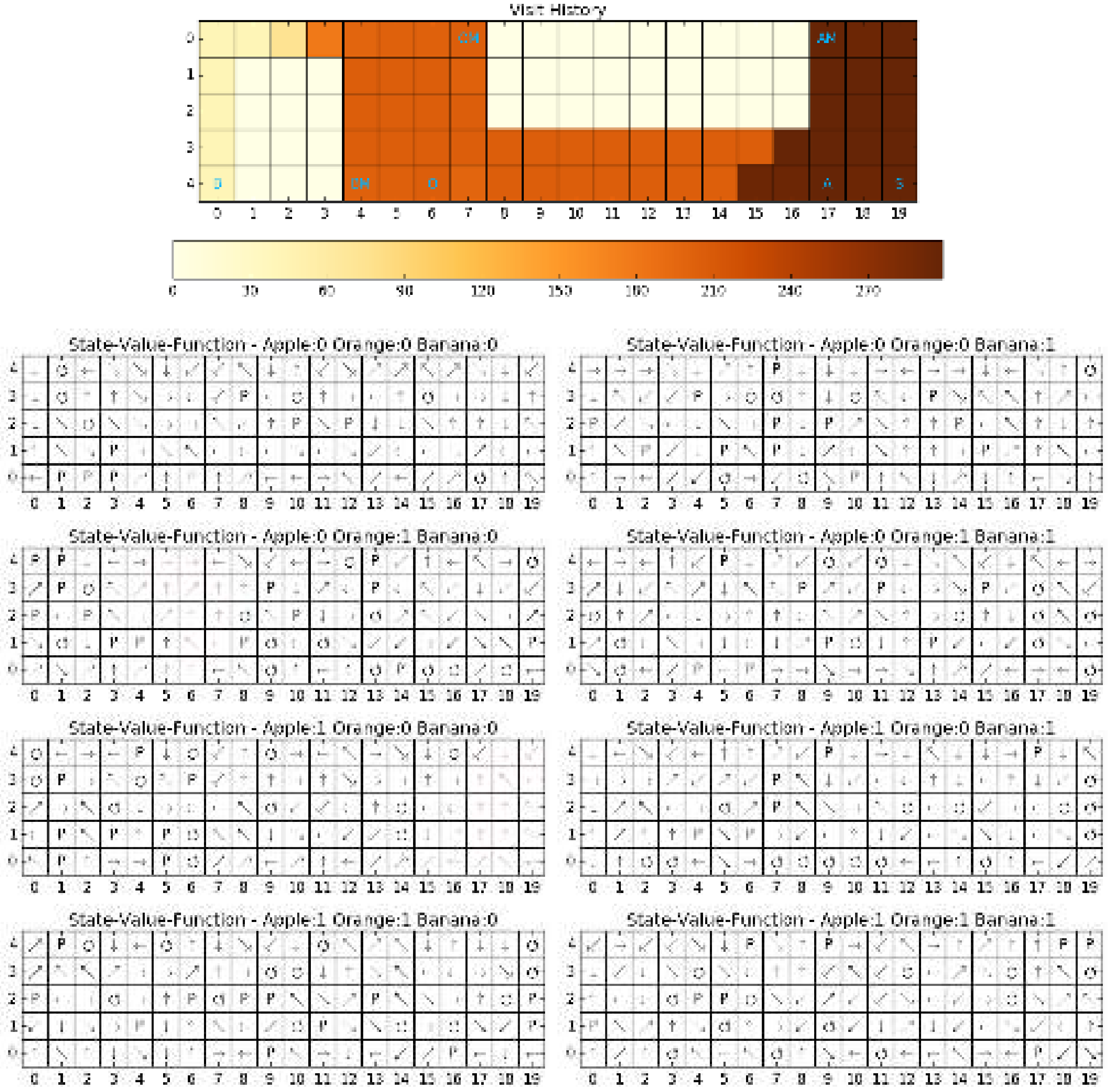


Figure 3.21.: SARSA Evaluation on Scenario 4

Evaluation a 300 learning episodes of the classic SARSA algorithm on the market domain. Parameters are set to $\epsilon = 0.05, \alpha = 0.75, \gamma = 0.75$. The heat-map shows the agent's position over the learning episodes, where state with brighter colors have been visited last in earlier episodes than states with darker colors. We now display the state-action-function for every possible combination of the third, fourth and fifth dimension of the agent's state, resulting in 8 plots, representing whether the agent picked up fruits or not. One can see the agent explores very little and converges towards picking up the apple (first row, left plot of state-action-function plots) and bringing it to the apple merchant (third row, left plot of state-value-function plots). The agent does not learn to pick up an orange or a banana, but learns a very shortsighted state-action-function to bring an orange to the orange merchant (second row, left plot of state-value function).

3.5.2 TDE

Evaluationg the TDE parameters on scenario 4, we find the best possible performance with a set of $\alpha = 0.5, \gamma = 0.75$. The results are displayed in Figure A.33, and the evaluation results can be found in Figure 3.22. More details can be seen in Figure A.34.



Visit History

State-Value-Function - Apple:0 Orange:0 Banana:0

State-Value-Function - Apple:0 Orange:0 Banana:1

State-Value-Function - Apple:0 Orange:1 Banana:0

State-Value-Function - Apple:0 Orange:1 Banana:1

State-Value-Function - Apple:1 Orange:0 Banana:0

State-Value-Function - Apple:1 Orange:0 Banana:1

State-Value-Function - Apple:1 Orange:1 Banana:0

State-Value-Function - Apple:1 Orange:1 Banana:1

Figure 3.22.: TDE Evaluation on Scenario 4

Evaluation a 300 learning episodes of the TDE approach on the market domain. Same definitions as in Figure 3.21 apply. Parameters are set to $\alpha = 0.5$ and $\gamma = 0.75$. The agent learns converges towards the apple merchant again, but due to the higher amount of exploration, he also learns a more farsighted state-action-function for bringing the orange to the orange merchant and even manages to learn very shortsighted state-value-functions for bringing a banana to the banana merchant.

3.5.3 Incompetence Motivation

Next, we evaluate our IM approach on the market domain. The results of evaluating the parameters can be found in Figure A.35. These evaluations give us the best performance for a set of parameters of $\epsilon = 0.001, \alpha = 0.75, \gamma = 0.25$. Results of one evaluation with these parameters can be found in Figure 3.23, more details in Figure A.36.

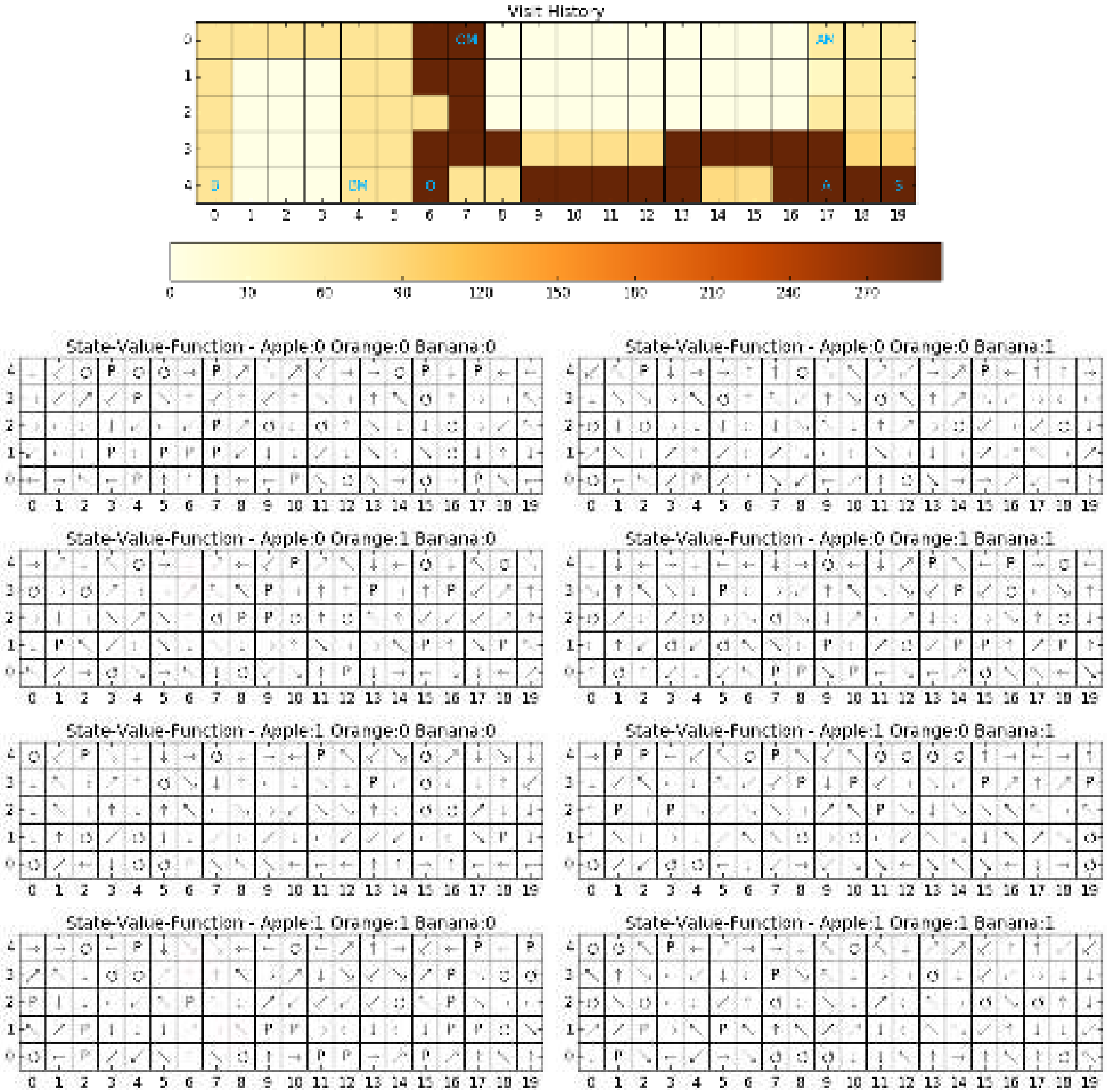


Figure 3.23.: IM Model Evaluation on Scenario 4

Evaluation a 300 learning episodes of our IM approach on the market domain. Same definitions as in Figure 3.21 apply. Parameters are set to $\epsilon = 0.001, \alpha = 0.75, \gamma = 0.25$. The heat-map is now showing the agent converges towards bringing the orange to the orange merchant. Even with a picked up apple (third row, left plot), he is pushed towards picking up an orange.

3.5.4 Competence Motivation

Now we evaluate the CM approach. Evaluating the SARSA parameters gives us the results shown in Figure A.37 and a best parameter set of $\epsilon = 0.01, \alpha = 0.25, \gamma = 0.5$. The results of running one evaluation with these parameter settings are displayed in Figure 3.24, more details in Figure A.38. The agent now explores even less than with using classic SARSA only, and really focuses on reaching the apple, picking it up and bringing it to the apple merchant.

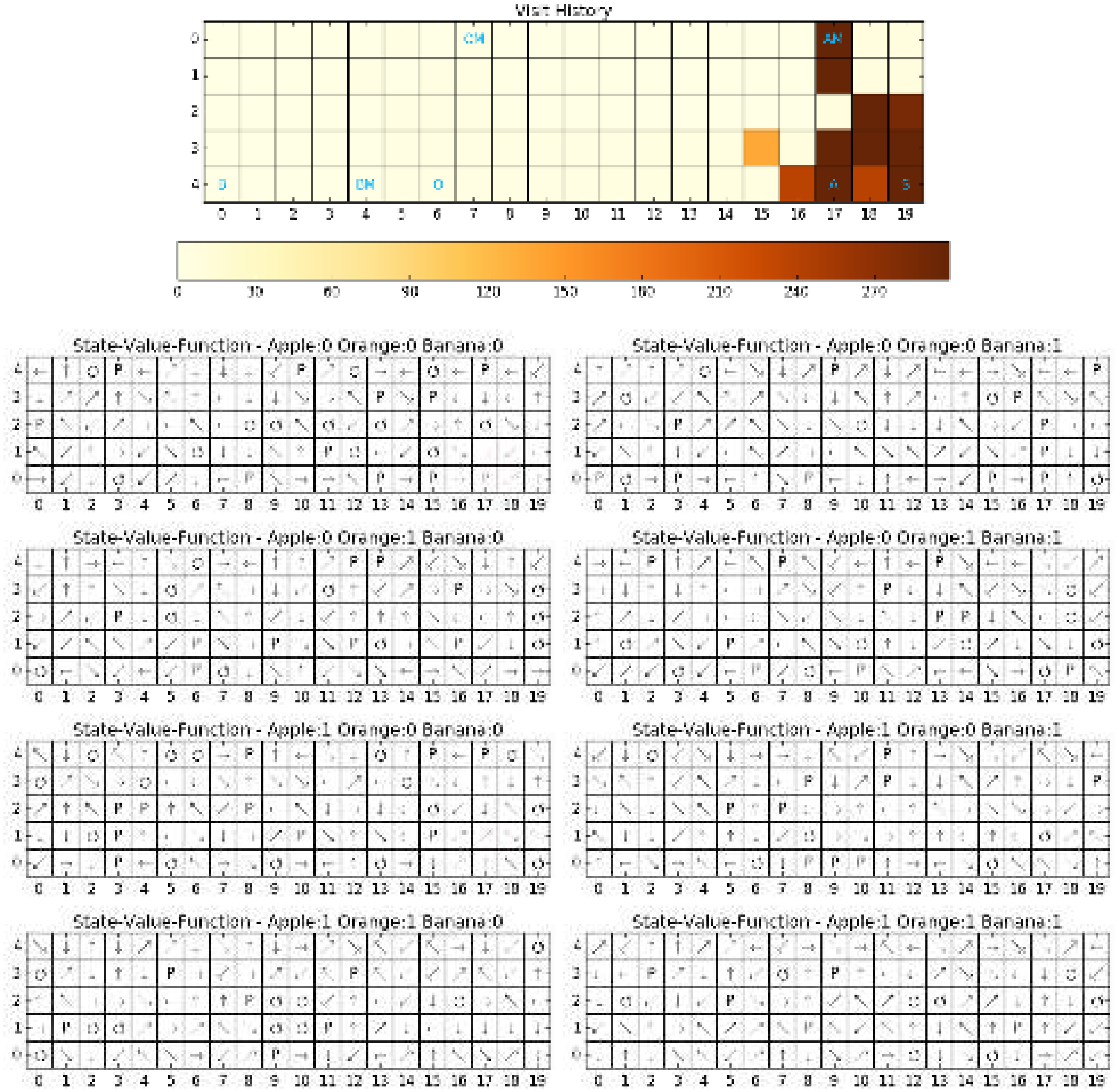


Figure 3.24.: CM Model Evaluation on Scenario 4

Evaluation a 300 learning episodes of our CM approach on the market domain. Same definitions as in Figure 3.21 apply. Parameters are set to $\epsilon = 0.01, \alpha = 0.25, \gamma = 0.5$. The heat-map now shows even less exploration than with SARSA and convergence towards the apple merchant.

Fourth and last, our CPM approach is evaluated on the market domain. The evaluation of the SARSA parameters gives a best set of $\epsilon = 0.01, \alpha = 0.75, \gamma = 0.75$. The results of these evaluations can be found in Figure A.39 while the results of one evaluation with those parameters are shown in Figure 3.25, with more details displayed in Figure A.40. The agent explores way more than with using the previous approaches and even manages to learn to bring a banana to the banana merchant but the evaluation details indicate a very murky and unstable behavior of learning episodes being interrupted now and then when making no progress, which results in a high amount of exploration.

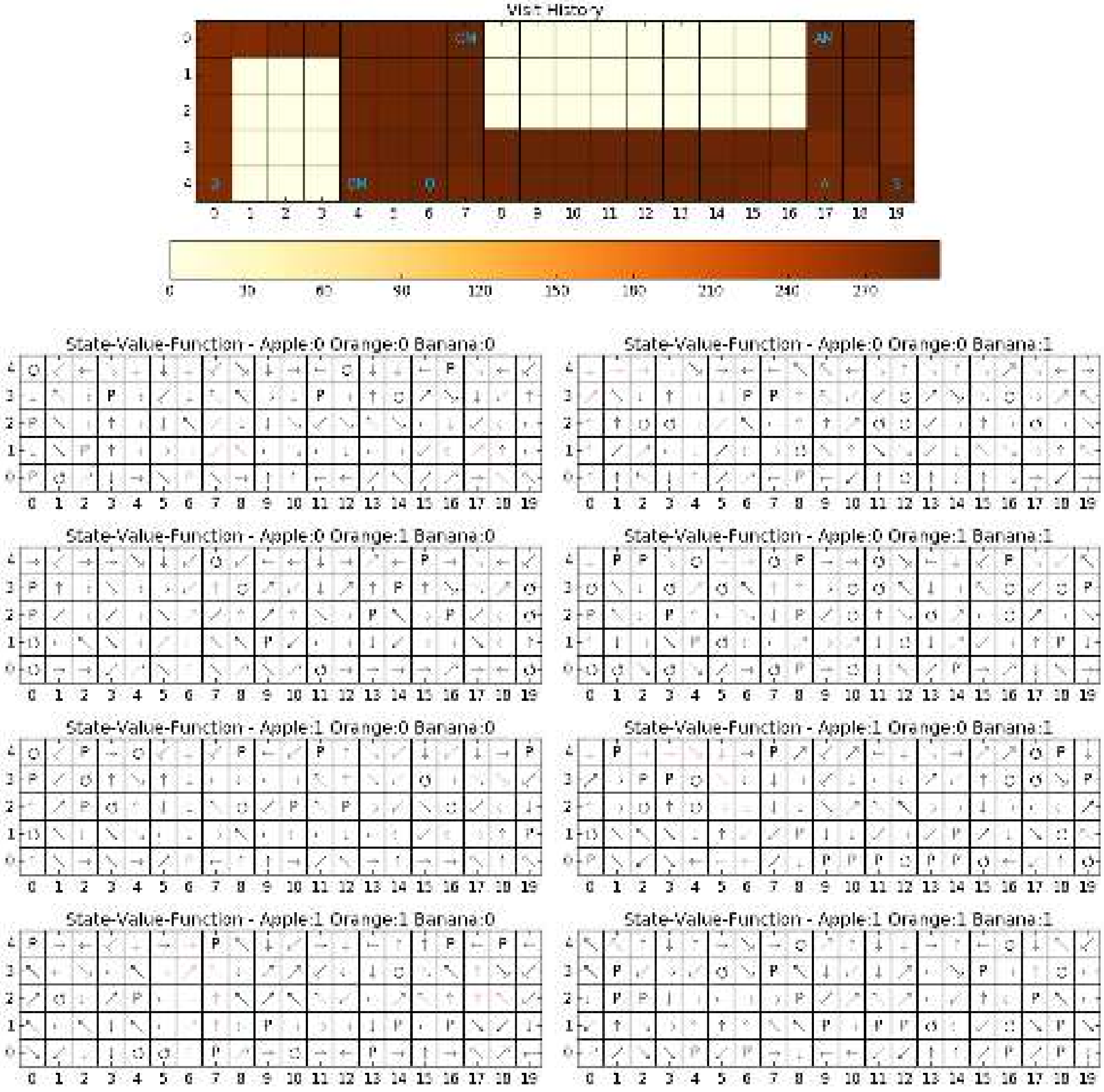


Figure 3.25.: CPM Model Evaluation on Scenario 4

Evaluation a 300 learning episodes of our CPM approach on the market domain. Same definitions as in Figure 3.21 apply. Parameters are set to $\epsilon = 0.01, \alpha = 0.75, \gamma = 0.75$. One can now see a lot of exploration in the later episodes. The agent does not learn state-action-functions to pick up the banana, rather than picking up the orange (first row, left plot), but often gets to pick it up randomly and is then able to learn state-action-functions pushing him towards the banana merchant (first and third row, right plots).

4 Discussion

While SARSA performs very well on domains with single terminal states, our experiments already show some big problems with the classic approach. In this chapter we shortly summarize these problems, before we compare the performance of our models for intrinsic motivation with the performance of classic SARSA and TDE.

4.1 SARSA, the Parameters, and the Limitations

As expected, SARSA performs well on the first grid-world in Figure 3.1. The agent is able to learn a policy which defines what actions to choose to reach the positively rewarding terminal-state while observing minimal negative reward. Although, the algorithm is not able to learn the best possible policy with respect to the amount of time-steps respective actions needed to reach the terminal state but comes very close to it. Figure 4.1 shows an example of this.

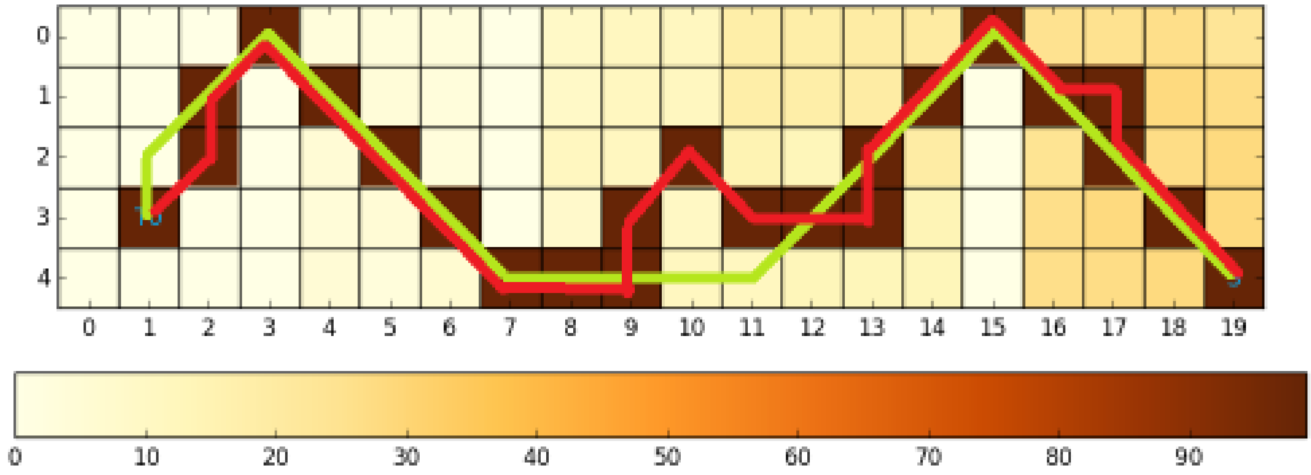


Figure 4.1.: Optimal Policy Problem

This figure shows the resulting visit history for one evaluation of SARSA on our first grid-world in Figure 3.1, with the parameters set to $\epsilon = 0.001,\alpha = 0.5$ and $\gamma = 0.5$. Darker colored states equal states that have been visited last in later episodes. Even though the algorithm converges very fast towards the target position in an optimal way regarding the observed reward, it is not optimal regarding the amount of primitive actions or time-steps needed to reach it. The optimal policy is displayed with a green line, while the policy the agent follows is shown with a red line. One can see it is with 22 actions 3 time-steps away from the optimum of 19 actions.

The classic SARSA algorithm does not perform well on the second grid-world in scenario 1, shown in Figure 3.4, as the agent is only able to learn a policy which leads him towards the goal $T1$ which provides him the smaller reward. Evaluating ϵ does show the agent is able to reach the higher rewarding goal $T0$, with a random movement of about 50%, but this high amount of random exploration results in observing a high amount of negative rewards. Evaluating α and γ for this scenario does not seem to have much of an influence on this kind of problems.

As expected, classic SARSA does not perform well on the third grid-world in scenario 2, displayed in Figure 3.6. Evaluating ϵ shows, it is again able to reach the rewarding terminal state with a high percentage of random movement of $\epsilon >= 0.1$.Evaluating γ shows an even better performance and convergence for a very high value for the discount factor. The best results are observed for $\gamma = 1.0$. Evaluating α shows, the learning rate does not seem to have much influence on this scenario.

The performance of classic SARSA on the first grid-world with respawning targets in scenario 3 is close to optimal regarding the observed reward and the amount the target is repositioned but not regarding the time-steps needed to terminate for each episode. Choosing a bigger ϵ results in reaching the targets more often, and more spawning targets, but also in collecting a higher number of negative rewards. Convergence is only achieved for $\epsilon \leq 0.01$, independent of the discount factor γ and the learning rate α. Choosing high values for these two parameters, the average amount of time-steps required to terminate decreases a bit.

Finally, the evaluations on the market domain show the bad performance of classic SARSA for more complex actions and environments. We observe convergence only for a small $\epsilon \leq 0.1$. Higher values of $\alpha \geq 0.75$ result in a higher mean reward but also in much more time-steps required to terminate. The same yields for low values of $\gamma \leq 0.5$. We discuss the influence of our models for intrinsic motivation in the next section of this chapter and compare the performance of these models with SARSA and TDE.

The compared best results of our three models for intrinsic motivation (CM, IM and CPM), SARSA and the time-decreasing ϵ approach on scenario 1 are shown and summarized in Figure 4.2. We can observe a convergence towards the smaller reward for SARSA, TDE and CM, whereas the IM approach successfully converges towards the higher reward after some exploration time. The CPM approach does follow the IM approach delayed first but starts to avoid the higher reward at about 100 episodes of learning where it explores more again. Thinking of our theoretical and psychological definitions of competence progress motivation, this is exactly what we expect it to do when it does not make any progress anymore.

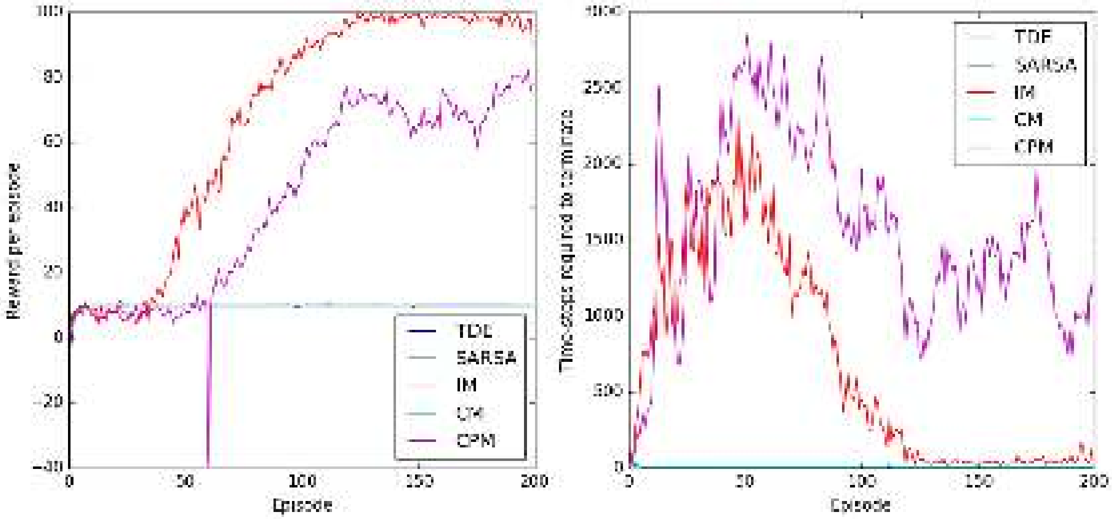


Figure 4.2.: Performance Comparison on Scenario 1

This figure shows the performance comparison of the classic SARSA algorithm, our models for intrinsic motivation and the time-decreasing epsilon (TDE) approach on the two goals domain in scenario 1, shown in Figure 3.4. The left plot displays the observed reward per episode, while the right plot shows the time-steps needed to terminate for each episode. The performance is averaged over 50 evaluations. The IM approach is able to converge towards the higher reward of +100 at about 120 episodes of learning. The CPM approach seems to converge in later episodes first, but starts to diverge at about 130 episodes of learning.

The compared performances on scenario 2 (shown in Figure 3.6) with states providing a negative reward on the path towards the positively rewarding terminal state can be seen in Figure 4.3 All approaches converge towards the reward of +100. The TDE approach achieves full convergence regarding the reward a bit later, but with less then half of the time-steps required, since learning episode 0.

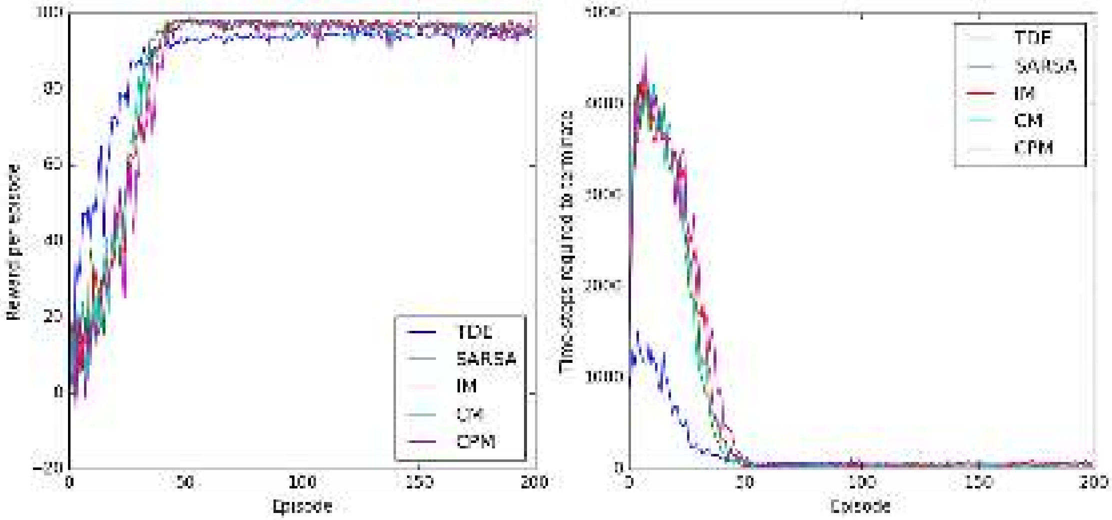


Figure 4.3.: Performance Comparison on Scenario 2

This figure shows the performance comparison of the classic SARSA algorithm, our models for intrinsic motivation and the time-decreasing epsilon (TDE) approach on the traps domain in scenario 2, shown in Figure 3.6. The left plot displays the observed reward per episode, while the right plot shows the time-steps needed to terminate for each episode. The performance is averaged over 50 evaluations. All approaches converge towards the reward of +100. The TDE approach achieves full convergence regarding the reward a bit later, but with less then half of the time-steps required, since learning episode 0.

Figure 4.4, shows the compared performance of the approaches on scenario 3, displayed in 3.1, with randomly repositioning targets, after being visited 10 times. While all other algorithms perform very well on this domain by choosing a high learning rate and low discount factor, TDE performs surprisingly bad. SARSA, IM, CM and CPM converge towards the reward of +10 and are able to reposition the terminal state 19 times which is the optimum. Though they need 500 or more time-steps in every episode of learning, which is far from optimal.

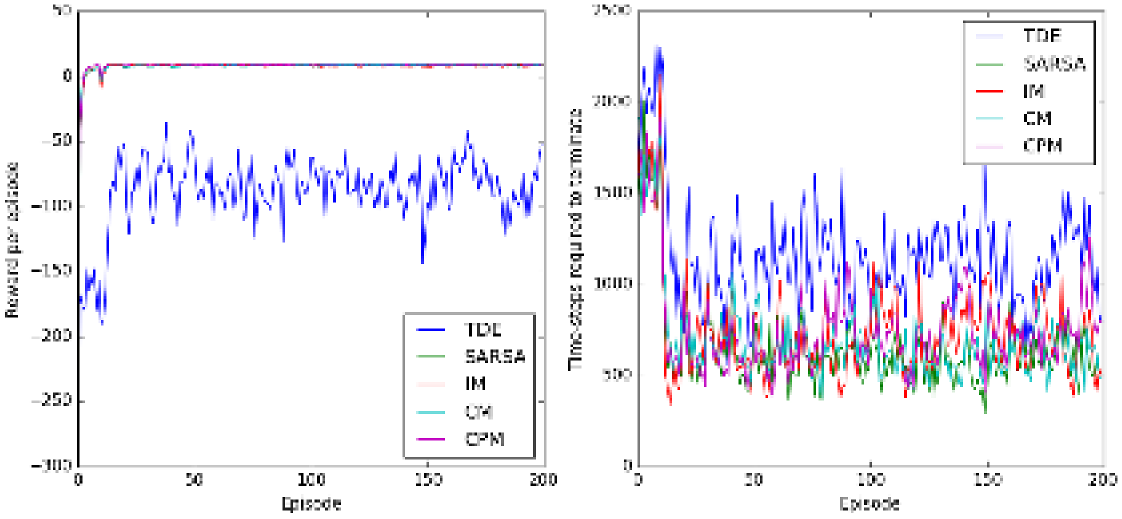


Figure 4.4.: Performance Comparison on Scenario 3

This figure shows the performance comparison of the classic SARSA algorithm, our models for intrinsic motivation and the time-decreasing epsilon (TDE) approach on the first grid-world domain in scenario 3, shown in Figure 3.1, with respawning terminal states. The left plot displays the observed reward per episode, while the right plot shows the time-steps needed to terminate for each episode. The performance is averaged over 50 evaluations. All approaches except TDE manage to converge towards the reward of +100 but always require 500 or more time-steps to reach the terminal state, which is far from optimal.

Our last scenario, the market domain shown in Figure 3.20, was introduced as a more complex grid-world with higher dimensional states and more abstract actions need to reach the terminal state. While the time-critical performance problem is addressed in other work including SMDPs and the option framework [42], our approaches focus on the exploration/exploitation trade-off in order to avoid local maxima and learn convergence towards the highest rewarding terminal state. Figure 4.5 indicates the superior performance of our IM and CPM approach regarding the observed reward, but also shows they are far from optimal regarding the amount of primitive actions or time-steps needed to terminate.

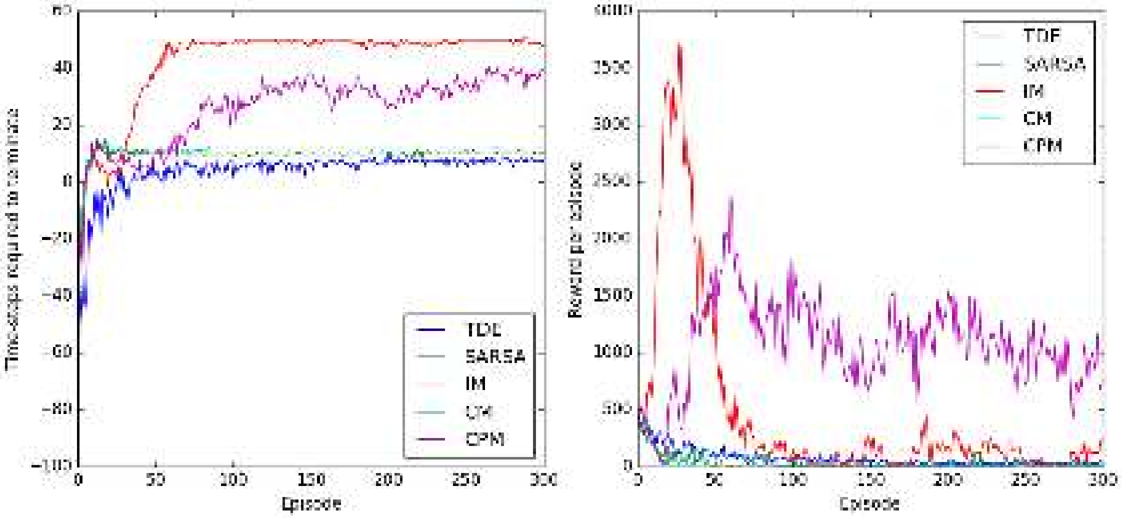


Figure 4.5.: Performance Comparison on Scenario 4

This figure shows the performance comparison of the classic SARSA algorithm, our models for intrinsic motivation and the time-decreasing epsilon(TDE) approach on the market domain in scenario 4, shown in Figure 3.20. The left plot displays the observed reward per episode, while the right plot shows the time-steps needed to terminate for each episode. The performance is averaged over 20 iterations.

One can see, SARSA, TDE and CM are converging towards the terminal state which is the easiest to reach but provides the lowest reward of 10. In theory, we would expect the IM and CPM algorithms to converge towards the terminal state providing the highest reward of 100, which is reached by picking up the banana in (4,0) and bringing it to the banana merchant in (4,4) but if we look at the observed rewards in Figure 4.5, these approaches only seem to converge towards the medium reward of 50 which is achieved by picking up an orange in (4,6) and delivering it to the orange merchant in (0,7). If we look at the observed rewards of one iteration of IM and CPM, shown together in Figure 4.6, we can see them actually doing so.

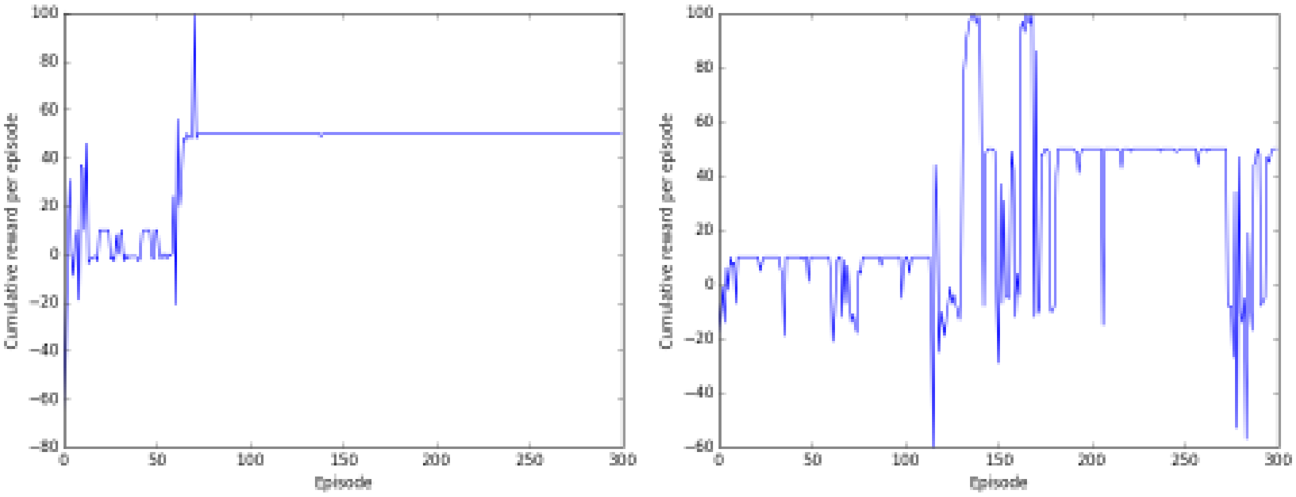


Figure 4.6.: IM and CPM Evaluation Observed Reward

This figure displays the observed reward per episode for one evaluation of the IM approach (left plot) and the CPM approach (right plot). While IM converges towards the reward of +50, the convergence using the CPM approach is frequently interrupted by phases of exploration, which actually enables the agent to learn reaching the maximum reward of +100 between episodes 120 and 180.

To understand what is happening we need to think about our approaches again: In the IM approach for example, we define the intrinsic reward by $r_i = C/l_g$. We set the constant C to be −100 and never change it through the experiments. This constant is actually part of one of the problems of our approaches. In theory SARSA should converge towards a level of misachievement $l_g = 0$ or a very small value which should provide maximum negative intrinsic reward for the incompetence motivation. Here we have two problems: First, SARSA does very seldom come close to $l_g = 0$, even for domains with a single terminal-state, see the final state-action-function in Figure A.2 for example. We observe a level of misachievement of $l_g \in [1,5]$ very often. In the market domain, $l_a = 5$ would result in an intrinsic reward of r_i=−100/5=−20 for the IM approach which would still provide a reward of r=$r_e - r_i$=50 − 20=30 for the algorithm and would not drive the agent away from the medium terminal state of delivering the orange. To sum it up, the constant C does actually control to which maximum attend the intrinsic reward is able to influence the extrinsic reward but one should probably define a range for levels of misachievement, for example define $l_a \in [0,...,5]$ as the best possible performance and the highest/lowest intrinsic reward. This would address the second problem of our approaches which lies in the nature of SARSA using an ϵ-greedy policy. The randomness results in a high variance of l_a and therefor a high variance of r_i which results sometimes in unstable results and and spikiness of the resulting graphs (see Figure A.41 for example). Figure 4.7 does show an evaluation of our IM extended SARSA approach, with parameter settings of $\epsilon = 0.001, \alpha = 0.75, \gamma = 0.5$ and the intrinsic reward constant set to $c = -200$, yielding another problem of our approaches.

The IM approach is starting to converge towards the highest reward of +100 provided by the banana merchant between episodes 130 and 230. The agent then goes randomly back towards bringing the apple to the apple merchant. Evaluations show it randomly receives the reward of +10 with a level of misachievement of about $l_g = 250$. This high level of misachievement results in a relative small negative intrinsic reward, so the agent will not discard this terminal state and converges towards it. Some other evaluations show, the agent sometimes switches randomly between the reward of +10, +50 and +100. Due to the randomness, it reaches these targets with a very high level of misachievement and so a very small negative intrinsic reward. This results in phase-wise learning to reach the terminal states instead of convergence towards one specific target. The observed behavior is sometimes very chaotic. The problem might lay in our definition of competence or might again lay in the nature of SARSA but future research is needed to bring up a solution for this.

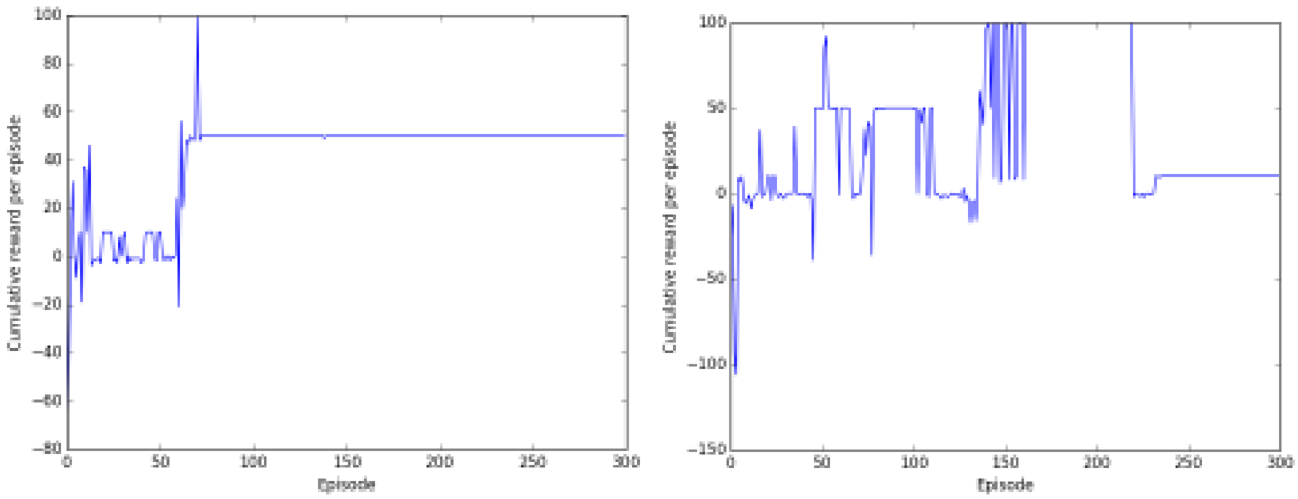


Figure 4.7.: The Effect of Rising Constant C

The figure shows the reward per episode of two evaluations of our IM extended SARSA approach. The parameters are set to set to $\epsilon = 0.001, \alpha = 0.75, \gamma = 0.5$. The evaluation in the left plot used an intrinsic reward constant of $c = -100$, while the right one used $c = -200$. We expected the agent to dismiss the medium reward of +50, towards which the left evaluation converged. The agent actually learns to receive the highest reward of +100 between episodes 130 and 230 episodes in the right evaluation, but then randomly dismisses it and learns to receive the smallest reward of +10. The reasons for this yield another problem of our approaches and are explained below.

To understand the CPM approach better, we should have another look at the iteration results in Figure A.40 where one can see, CPM is actually learning a policy which leads the agent towards the banana merchant at about 150 episodes but dismisses it at about 250 episodes and converges towards the medium rewarding terminal state of the orange merchant. Looking at the state-action-functions in Figure 3.25 confirms this as the agent has learned to pick up a banana when it is close to it and bring it to the merchant. The randomness of SARSA is to blame for the averaged bad performance of CPM on this domain. Future work is needed to investigate the influence of changing the constant C and of using ranged levels of misachievement, as explained above.

Finally, we have presented interesting approaches, based on neurobiological and psychological definitions of motivation, to deal with the exploration/exploitation problem in the reinforcement learning framework. Although the behavior is sometimes murky, IM and CPM outperform classic SARSA and the TDE approach regarding the observed external reward, especially for scenario 1 and scenario 4. Our CM approach does not seem to bring many benefits, at best a faster convergence than SARSA towards the same reward in scenario 1 and scenario 4. Our approaches do perform as bad as SARSA on scenario 2 and get outperformed by the TDE approach. For scenarios like this, we suggest to use knowledge-based models, for example giving a growing negative intrinsic reward for already explored states. All approaches except TDE manage to perform well on scenario 3. We have also shown some of the problems of our approaches and some already existing problems of the RL framework which are summarized and addressed in the next chapter as interesting topics for future research.

5 Outlook

Many challenges still remain for future work and some already existing approaches to improve reinforcement learning could be used together with models for intrinsic motivation to guide learning and result in an autonomous development. This chapter gives an overview about these challenges and approaches.

5.1 Option Theory - Temporally Extended Actions for Reinforcement Learning

One challenge for artificial intelligences is to learn, plan and represent knowledge at different levels of temporal abstraction. One approach to address this challenge within the framework of reinforcement learning and Markov decision processes is the option theory given by Sutton, Precup and Singh in [42]. They use options, an extended notion of the classic actions in the RL and MDP framework, to describe closed-loop policies for taking one ore more primitive actions over time. Defining these options over a MDP gives a semi-Markov decision process (SMDP). Several methods, like SMDP Q-Learning [42], exist to solve SMDPs. Using these methods enables taking advantage of the simplicities and efficiencies sometimes available at higher levels of temporal abstraction and will result in a significant higher performance on learning tasks [42]. One could combine the SMDP framework with our methods for intrinsic motivation to achieve even better results, similar to [39] and [44].
Most of the goals or terminal states in these experiments with algorithms for SMDP learning are hand-made, created together with the grid-world they are located in. Fully autonomous learning would require an automatic discovery of these goals without the need of interaction or judgment by a human observer. One approach for automatic goal selection for SMDPs is advocated by Stolle in [51]. Intrinsic Motivation models could then be also used to guide this goal selection in a different manner than presented in this thesis. Instead of using them to deal with the exploration/exploitation trade off, one could also use them to guide goal selection and optimal exploration, similar to [50] and [47].

5.2 Real Physical Embodied Applications

While our models and the already existing ones are easily applied on low-dimensional state spaces, future work is needed to build models well-suited for the continuous space of physical robots. Oudeyer and Kaplan already make suggestions how to use the models in [7] and some already existing approaches can be found e.g. in [41], [50], [47] and [46].

5.3 The Competence Problem

As mentioned in the previous chapter, using models for competence-based intrinsic motivation, an assumption we made, is to give the agent knowledge about the competence criteria needed to achieve a good performance in absolving a task, as well as the definitions of good performance, designed together with the grid-world domains. For a more autonomous development the agent might need to learn these two things on it's own: First, the competence criteria which might be the minimum time required to reach a target, or the maximum external reward while trying to do so. Second, it needs to judge about the quality of performance, building up a scale for the discovered variable of competence and judge about good and bad performance. It will then be able to learn tasks motivated by competence, incompetence, or the progress on it, as shown in chapter 1. Future work is also needed to investigate the influences of the constant C used in our models for intrinsic motivation, as well as in the work of Oudeyer and Kaplan [7]. Furthermore, one could evaluate these approaches with a ranged level of misachievement l_a to increase the stability of learning, as explained in chapter 4, as well as using a goal-space region with a simple distance threshold for terminal states (presented in [7]), instead of using single terminal states.

Bibliography

[1] F. Rheinberg, *Intrinsische Motivation und Flow-Erleben*, pp. 331–354. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006.

[2] H. Heckhausen, "Motiv und motivation," *Handbuch psychologischer Grundbegriffe, S*, pp. 296–313, 1977.

[3] V. Vroom, "Work and motivation," *John Willey & Sons, New York*, 1964.

[4] F. Rheinberg, "Zweck und taetigkeit," 1989.

[5] R. M. Ryan and E. L. Deci, "Intrinsic and extrinsic motivations: Classic definitions and new directions," *Contemporary educational psychology*, 2000.

[6] M. R. Lepper, D. Greene, and R. E. Nisbett, "Undermining children's intrinsic interest with extrinsic reward: A test of the 'overjustification' hypothesis.," *Journal of Personality and social Psychology*, vol. 28, p. 129, 1973.

[7] P. Y. Oudeyer and F. Kaplan, "What is intrinsic motivation? a typology of computational approaches," *Frontiers in neurorobotics*, vol. 1, 2007.

[8] C. Hull, "Principles of behavior," 1943.

[9] K. Montgomery, "Exploratory behavior and its relation to spontaneous alternation in a series of maze exposures.," *Journal of Comparative and Physiological Psychology*, vol. 45, no. 1, p. 50, 1952.

[10] H. F. Harlow, M. K. Harlow, and D. R. Meyer, "Learning motivated by a manipulation drive.," *Journal of Experimental Psychology*, vol. 40, no. 2, p. 228, 1950.

[11] R. W. White, "Motivation reconsidered: The concept of competence.," *Psychological review*, vol. 66, no. 5, p. 297, 1959.

[12] L. Festinger, "A theory of cognitive dissonance evanston," *IL: Row, Peterson*, vol. 1, 1957.

[13] J. Kagan, "Motives and development.," *Journal of personality and social psychology*, vol. 22, no. 1, p. 51, 1972.

[14] M. Zuckerman, *Sensation seeking*. Wiley Online Library, 1979.

[15] J. M. Hunt, "Intrinsic motivation and its role in psychological development," in *Nebraska symposium on motivation*, vol. 13, pp. 189–282, 1965.

[16] W. N. Dember and R. W. Earl, "Analysis of exploratory, manipulatory, and curiosity behaviors.," *Psychological review*, vol. 64, no. 2, p. 91, 1957.

[17] D. E. Berlyne, "Conflict, arousal, and curiosity.," 1960.

[18] R. de Charms, *Personal causation*. 1968.

[19] E. L. Deci and R. M. Ryan, "The general causality orientations scale: Self-determination in personality," *Journal of research in personality*, vol. 19, no. 2, pp. 109–134, 1985.

[20] M. Csikszentmihalyi, "The flow experience and its significance for human psychology.," 1988.

[21] M. Herger, "Flow-theory," 2015.

[22] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*, vol. 1. MIT press Cambridge, 1998.

[23] R. A. Howard, "Dynamic programming and markov processes," 1960.

[24] Y. Dodge, *The Oxford dictionary of statistical terms*. Oxford University Press on Demand, 2006.

[25] M. Tokic and G. Palm, "Value-difference based exploration: adaptive control between epsilon-greedy and softmax," *KI 2011: Advances in Artificial Intelligence*, pp. 335–346, 2011.

[26] C. J. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, no. 3-4, pp. 279–292, 1992.

[27] G. A. Rummery and M. Niranjan, *On-line Q-learning using connectionist systems*, vol. 37. University of Cambridge, Department of Engineering, 1994.

[28] B. F. Skinner, *Science and human behavior*. Simon and Schuster, 1953.

[29] K. C. Berridge, "Food reward: Brain substrates of wanting and liking," *Neuroscience Biobehavioral Reviews*, vol. 20, no. 1, pp. 1 – 25, 1996.

[30] R. Arkin, "Moving up the food chain: Motivation and emotion in behavior-based robots," *Who needs Emotions: The Brain Meets the Robot*, 2005.

[31] C. Breazeal and B. Scassellati, "Robots that imitate humans," *Trends in cognitive sciences*, vol. 6, no. 11, pp. 481–487, 2002.

[32] J. Weng, "Developmental robotics: Theory and experiments," *International Journal of Humanoid Robotics*, vol. 1, no. 02, pp. 199–236, 2004.

[33] G. Konidaris and A. Barto, "An adaptive robot motivational system," in *SAB*, pp. 346–356, Springer, 2006.

[34] D. McFarland, "Towards robot cooperation," *From animals to animats*, vol. 3, pp. 440–444, 1994.

[35] A. Maslow and K. Lewis, "Maslow's hierarchy of needs," *Salenger Incorporated*, vol. 14, 987.

[36] S. Singh, R. L. Lewis, A. G. Barto, and J. Sorg, "Intrinsically motivated reinforcement learning: An evolutionary perspective," *IEEE Transactions on Autonomous Mental Development*, vol. 2, no. 2, pp. 70–82, 2010.

[37] A. G. Barto, S. Singh, and R. L. Lewis, "Intrinsically motivated machines," 2005.

[38] Ö. Şimşek and A. G. Barto, "An intrinsic reward mechanism for efficient exploration," in *Proceedings of the 23rd international conference on Machine learning*, pp. 833–840, ACM, 2006.

[39] N. Chentanez, A. G. Barto, and S. P. Singh, "Intrinsically motivated reinforcement learning," *Advances in neural information processing systems*, pp. 1281–1288, 2005.

[40] W. Schneider and R. M. Shiffrin, "Controlled and automatic human information processing: I. detection, search, and attention.," *Psychological review*, vol. 84, no. 1, p. 1, 1977.

[41] A. Stout, G. D. Konidaris, and A. G. Barto, "Intrinsically motivated reinforcement learning: A promising framework for developmental robot learning," tech. rep., MASSACHUSETTS UNIV AMHERST DEPT OF COMPUTER SCIENCE, 2005.

[42] R. S. Sutton, D. Precup, and S. Singh, "Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning," *Artificial intelligence*, vol. 112, no. 1-2, pp. 181–211, 1998.

[43] P. Stone, *Layered learning in multiagent systems: A winning approach to robotic soccer*. MIT Press, 1998.

[44] A. Stout and A. G. Barto, "Competence progress intrinsic motivation," in *Development and Learning (ICDL), 2010 IEEE 9th International Conference on*, pp. 257–262, IEEE, 2010.

[45] A. G. Barto and S. Mahadevan, "Recent advances in hierarchical reinforcement learning," *Discrete Event Dynamic Systems*, vol. 13, no. 4, pp. 341–379, 2003.

[46] A. Gabriel, R. Akrour, J. Peters, G. Neumann, *et al.*, "Empowered skills," 2017.

[47] V. G. Santucci, G. Baldassarre, and M. Mirolli, "Grail: A goal-discovering robotic architecture for intrinsically-motivated learning," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 8, no. 3, pp. 214–231, 2016.

[48] A. Baranes and P.-Y. Oudeyer, "Robust intrinsically motivated exploration and active learning," in *Development and Learning, 2009. ICDL 2009. IEEE 8th International Conference on*, pp. 1–6, IEEE, 2009.

[49] A. Baranes and P.-Y. Oudeyer, “Maturationally-constrained competence-based intrinsically motivated learning,” in *Development and Learning (ICDL), 2010 IEEE 9th International Conference on*, pp. 197–203, IEEE, 2010.

[50] A. Baranes and P.-Y. Oudeyer, “Active learning of inverse models with intrinsically motivated goal exploration in robots,” *Robotics and Autonomous Systems*, vol. 61, no. 1, pp. 49–73, 2013.

[51] M. Stolle, *Automated discovery of options in reinforcement learning*. PhD thesis, McGill University, 2004.

[52] A. Baranes and P.-Y. Oudeyer, “Intrinsically motivated goal exploration for active motor learning in robots: A case study,” in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pp. 1766–1773, IEEE, 2010.

A Appendix

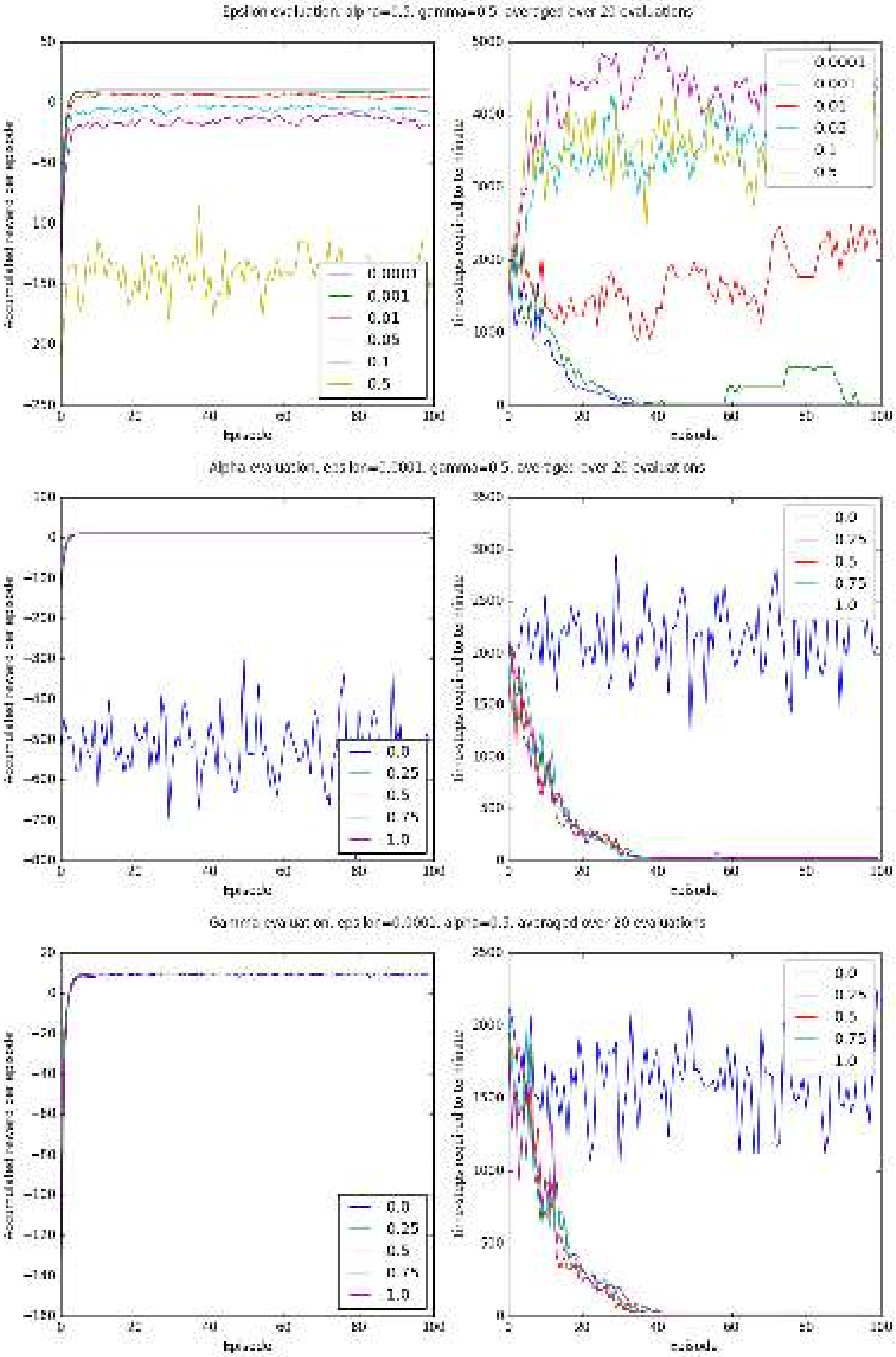


Figure A.1.: SARSA Parameter Evaluation for First Grid-World

This figure displays the results of evaluating the SARSA parameters on the first grid-world. The first row shows the results of evaluating ϵ on the left and the results of evaluating α on the right, the second row displays the results of evaluating γ. Each plot itself shows the reward per episode for each evaluation on the left, while the right side displays the amount of time-steps needed to terminate for each episode of each evaluation. We average over 20 evaluations and observe the best performance for $\epsilon = 0.0001$, while α and γ seem to have no influence on the performance unless they are set higher than 0.

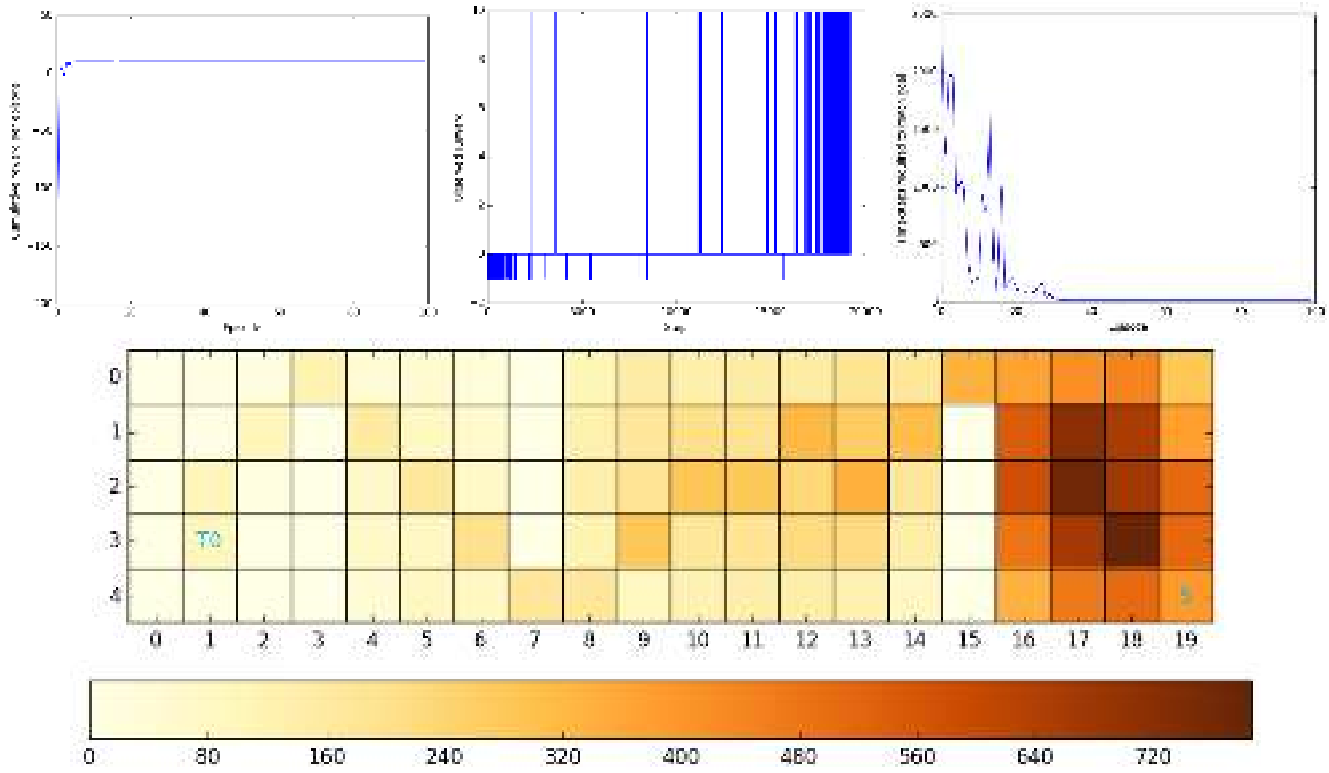


Figure A.2.: SARSA Evaluation Details on First Grid-World

The results of one evaluation of the classic SARSA algorithm with the parameters set to $\epsilon = 0.0001$, as the best performing ϵ from the above evaluations, $\alpha = 0.5$ and $\gamma = 0.5$ as the mean of their value range. The top left plot shows the accumulated reward at the end of each learning episode. One can see the reward is converging towards +10, provided by the terminal position $T0$. The top middle plot shows the current observed reward for each time-step. The agent is learning its state-action-function, exploring randomly first (negative rewards of -1), which decreases more and more and results in reaching the terminal state more and more, providing the reward of +10. The top right plot displays the amount of time-steps needed to terminate for each episode. One can see the agent exploring in the beginning up to episode 20, where it has learned to get to $T0$ with almost optimal amount of time-steps and reaches convergence at about 30 episodes of learning. A heat-map over the amount of times a state has been visited by the agent is shown in the bottom plot. Darker boxes equal states that have been visited more often. The color-bar displays the amount of times a state has been reached. One can see the agent is following a path through the grid-world, but does mainly explore its environment east of the first wall and next to its starting position S.

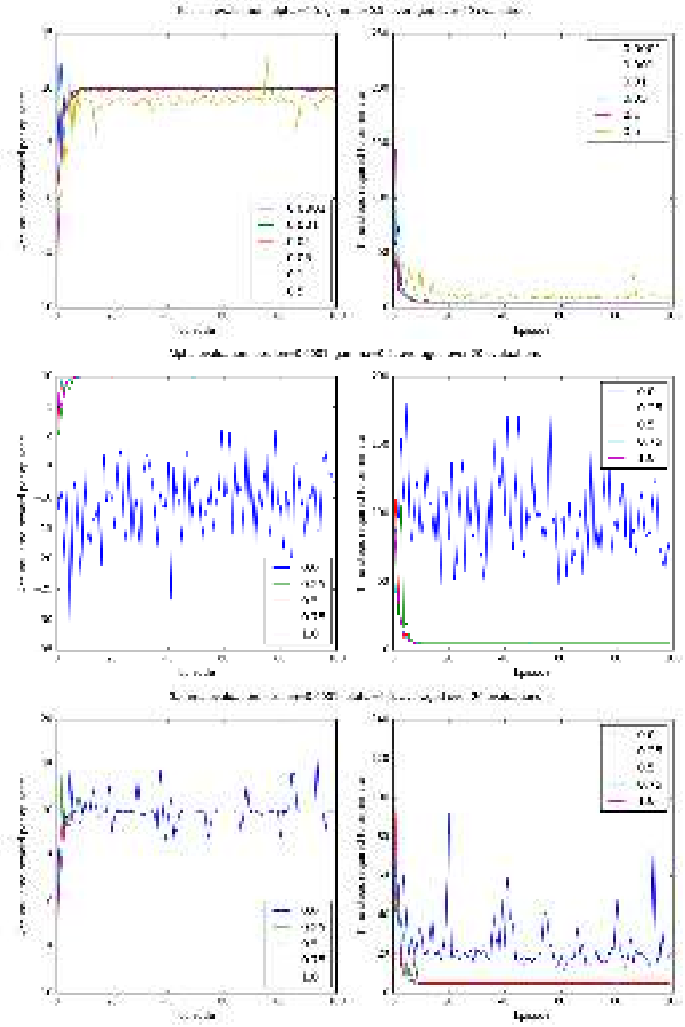


Figure A.3.: SARSA Parameter Evaluation for Scenario 1

Same definitions as in Figure A.1 apply. While convergence towards the reward of +10 is only achieved with $epsilon \leq 0.1$, SARSA is never able to converge towards the higher reward of +100. α and γ seem to have no influence again, unless they are set higher than 0.

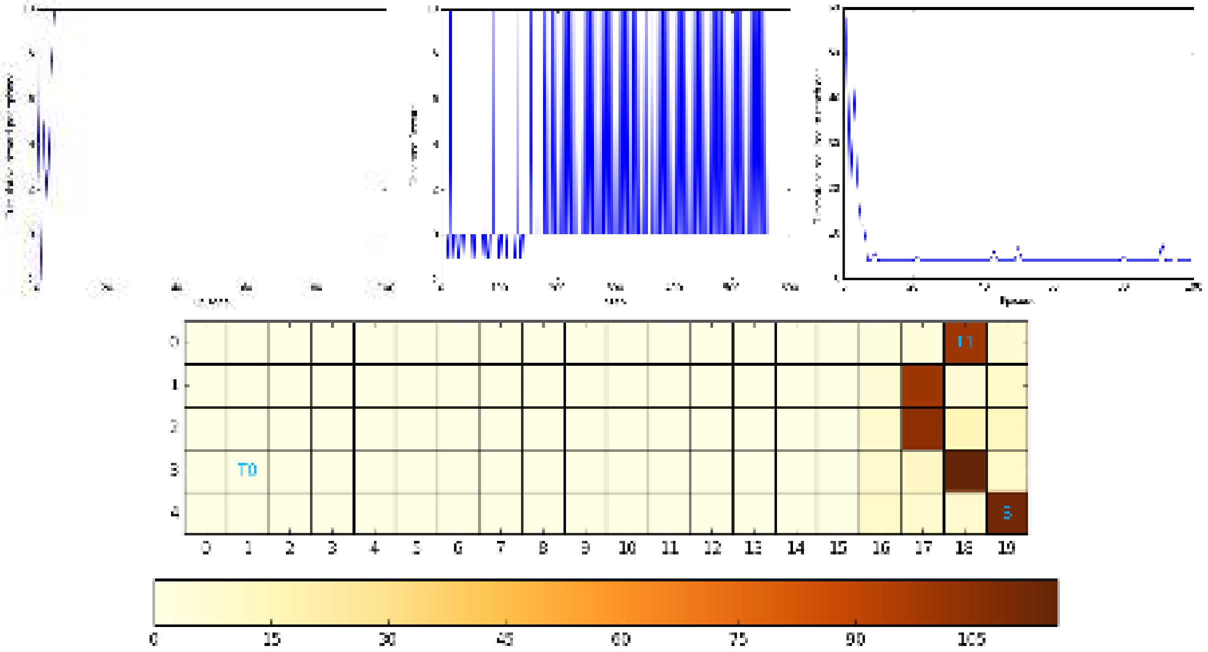


Figure A.4.: SARSA Evaluation Details on Scenario 1

Same definitions as in Figure A.2 apply. The parameters are set to $\epsilon = 0.01, \alpha = 0.5, \gamma = 0.5$. Learning an optimal path of 4 time-steps, the agent is converging towards the target $T1$, providing him a reward of +10, in less than 10 episodes of learning. Though it is never able to learn reaching $T0$, providing the much higher reward of +100.

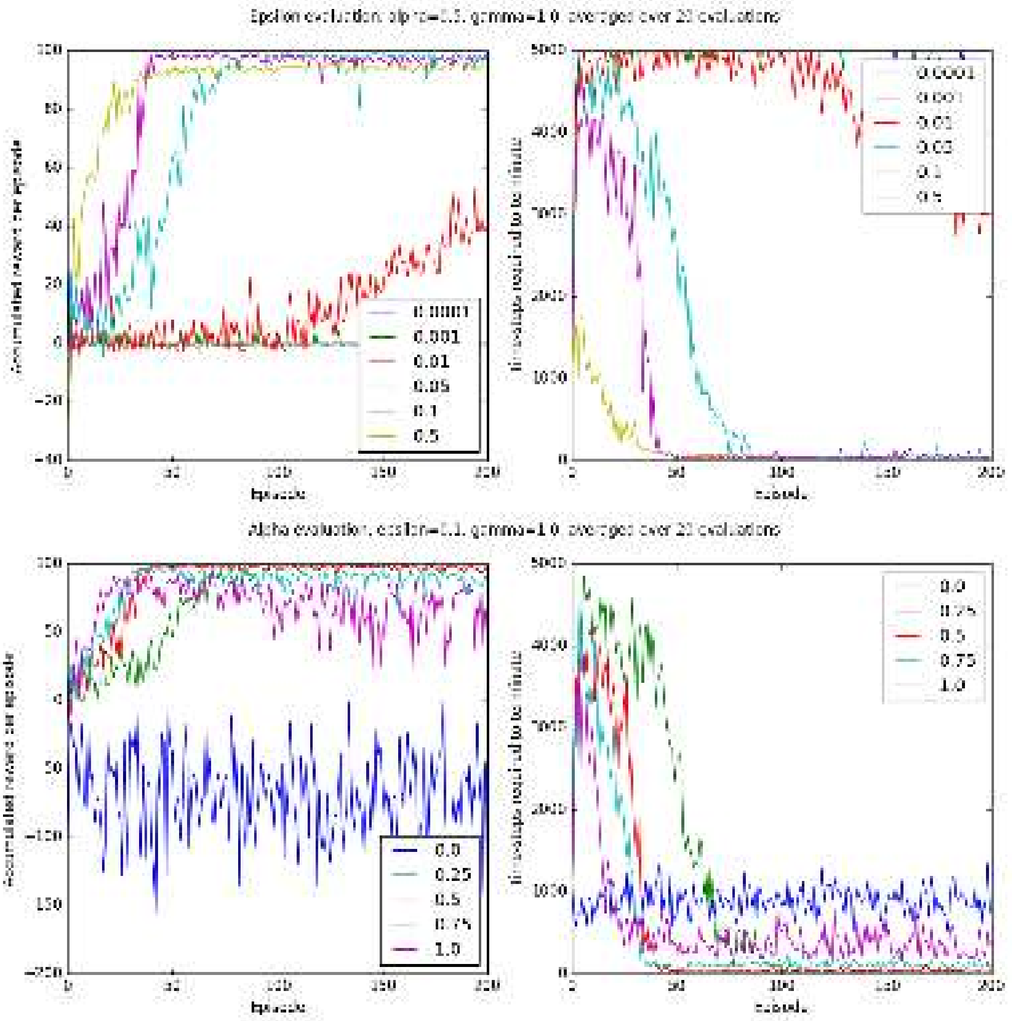


Figure A.5.: SARSA Parameter Evaluation for Scenario 2

Same definitions as in Figure A.1 apply. Convergence is reached best for high amounts of random exploration ($\epsilon \geq 0.05$, a medium learning rate of $\alpha = 0.5$ and the highest possible discount factor of $\gamma = 1.0$.

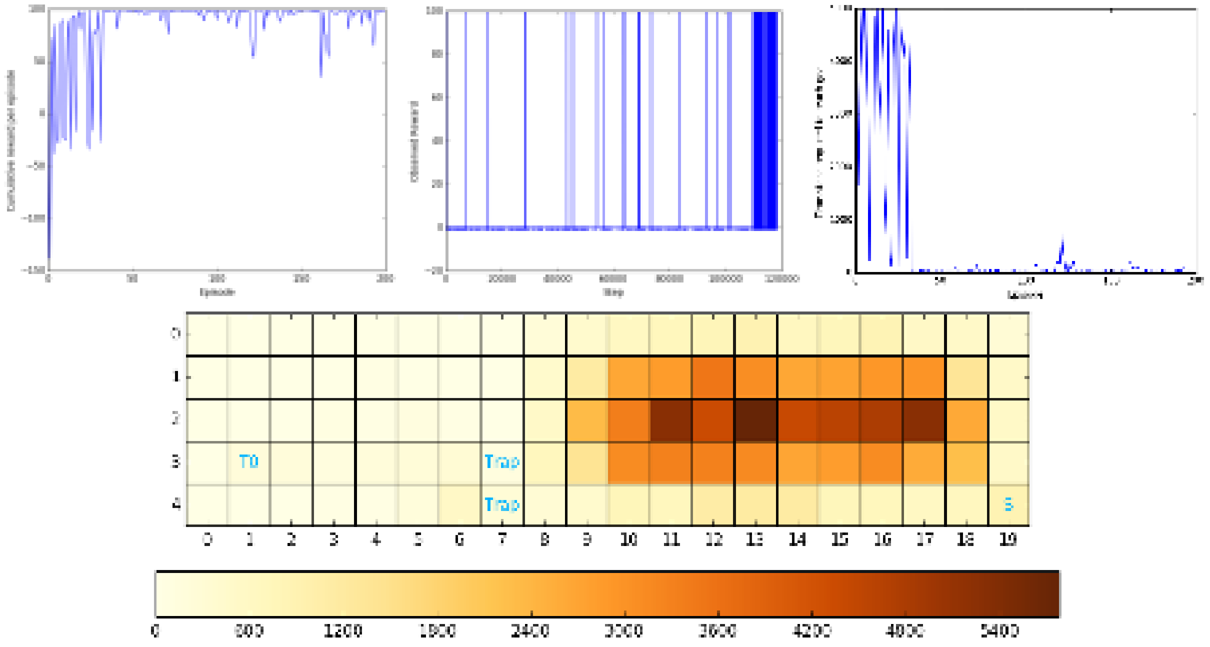


Figure A.6.: SARSA Evaluation Details on Scenario 2

Same definitions as in Figure A.2 apply. The parameters are set to $\epsilon = 0.1, \alpha = 0.5, \gamma = 1.0$. One can observe convergence for about 40+ episodes, but the agent is often disturbed due to the high amount of random exploration, which happens mainly east of the wall and results in the consistent high amount of observing reward -1.

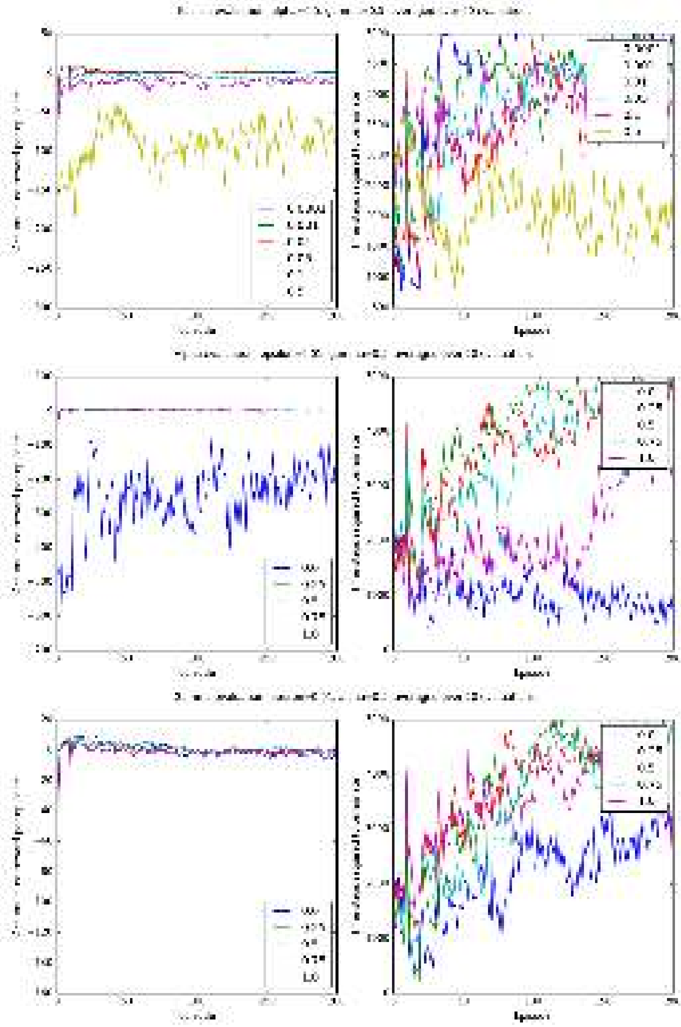


Figure A.7.: SARSA Parameter Evaluation for Scenario 3

Same definitions as in Figure A.1 apply. Convergence regarding the observed reward is only reached for $\epsilon \leq 0.01$, which results in a divergence regarding the time-steps needed to terminate. This can be reduced by choosing the highest possible learning rate $\alpha = 1.0$ and the lowest possible discount factor $\gamma = 0.0$, to make the agent adaptive but shortsighted.

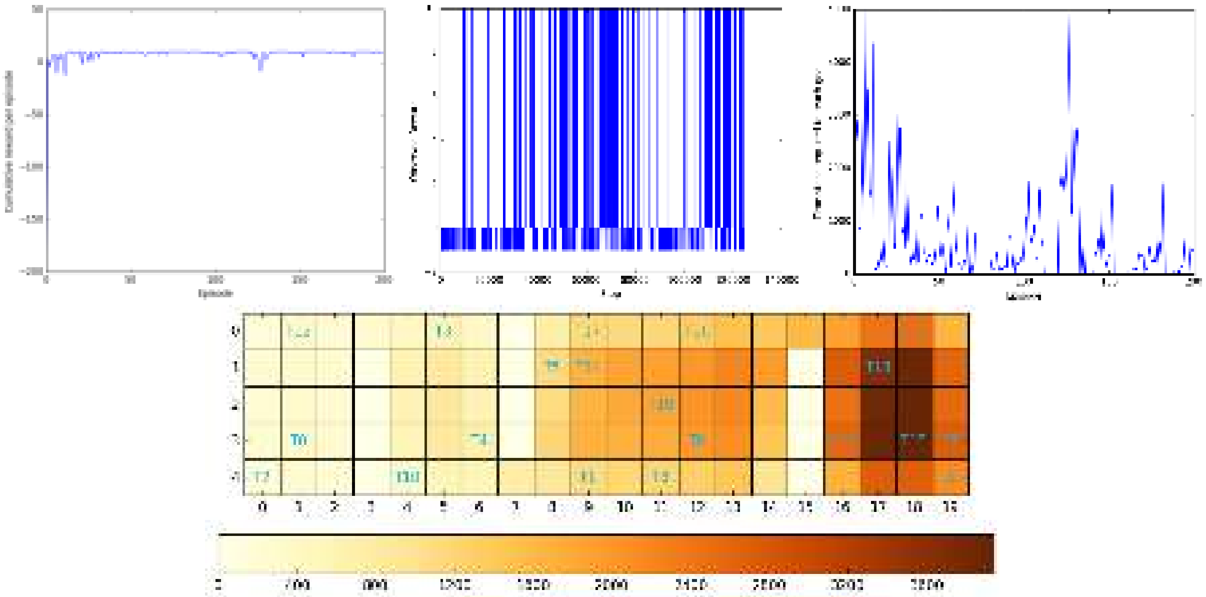


Figure A.8.: SARSA Evaluation Details on Scenario 3

The same definitions as in Figure A.2 apply. The algorithms parameters are set to $\epsilon = 0.01, \alpha = 1.0, \gamma = 0.0$. While the agent performs almost optimal regarding the observed external reward, it takes surprisingly long to terminate in many episodes. It also does observe a negative reward of -1 surprisingly often. The exploration takes mainly part east to the first wall or between the first and second wall (from the agent's starting position S).

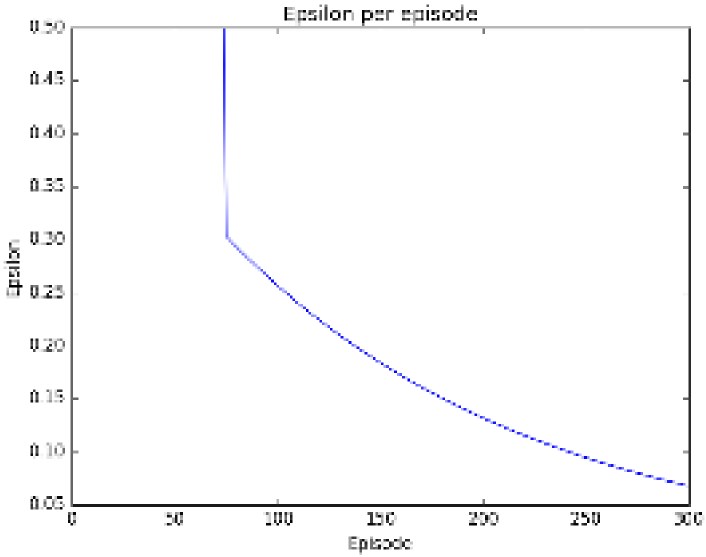


Figure A.9.: Time Decreasing Epsilon

The ϵ used for the "time-decreasing epsilon" (TDE) approach in chapters 3 and 4 for comparison with our models for intrinsic motivation and SARSA. ϵ is set to 0.5 for 1/4 of the learning episodes, before it decreases exponentially with $\epsilon = 0.5e^{(} - t/0.5E)$, where t is the current episode and E is the amount of learning episodes.

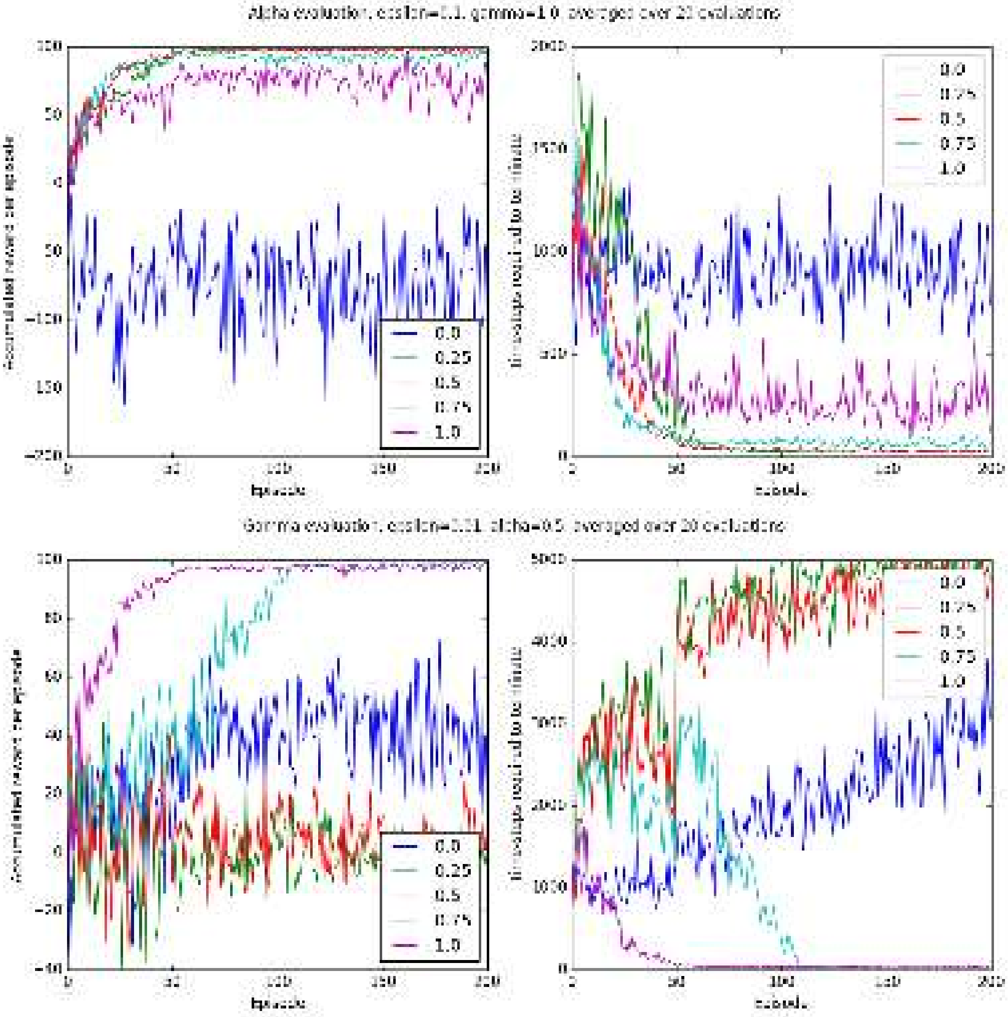


Figure A.10.: TDE Parameter Evaluation for Scenario 2

The same definitions as in Figure A.1 apply. Convergence is achieved for medium learning rates $\alpha \in [0.25, 0.5, 0.75]$ and high discount factors $\gamma \geq 0.75$.

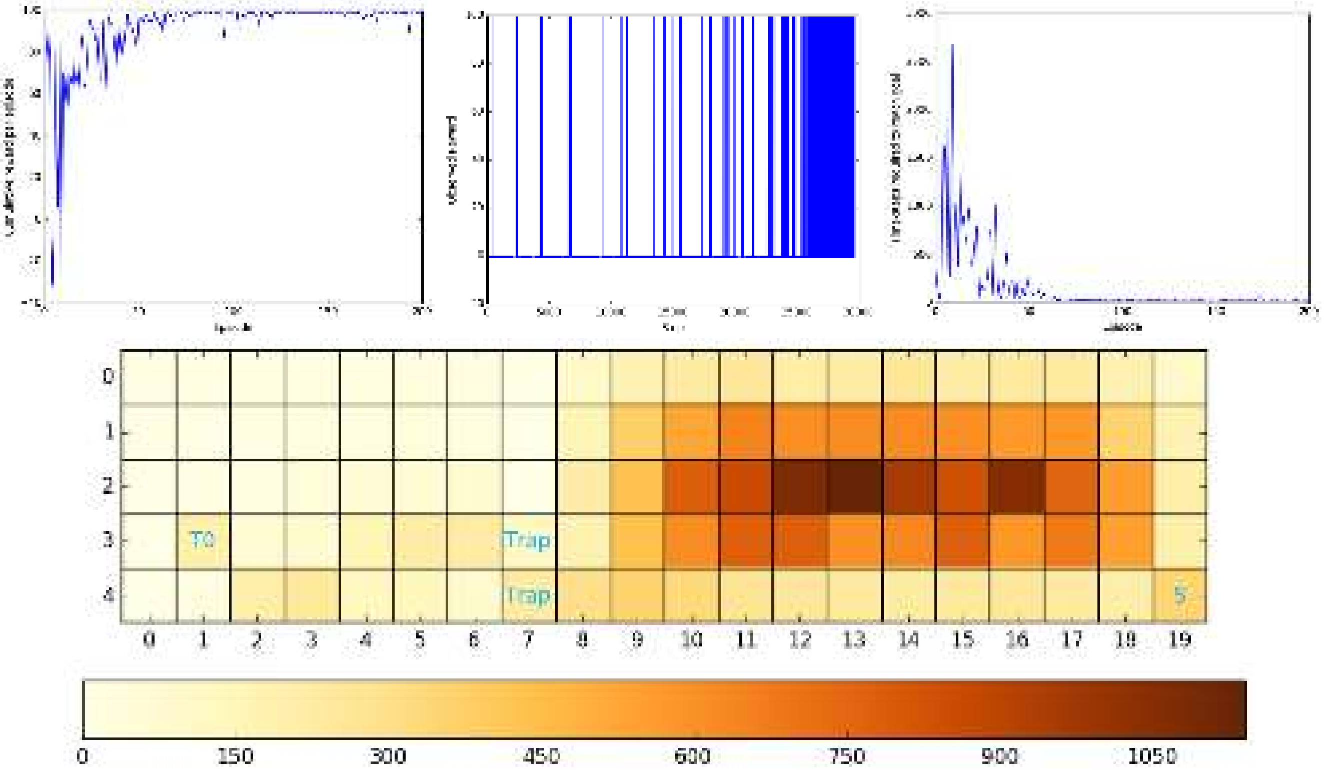


Figure A.11.: TDE Evaluation Details on Scenario 2

This figure shows the details of one evaluation of TDE on scenario 2. The same definitions as in Figure A.2 apply. The parameters are set to $\alpha = 0.5, \gamma = 1.0$. Convergence is reached at about 60 episodes of learning. Compared with the results of evaluating SARSA, the visited world now looks a bit more like a path through the environment, which indicates a little more stable learning.

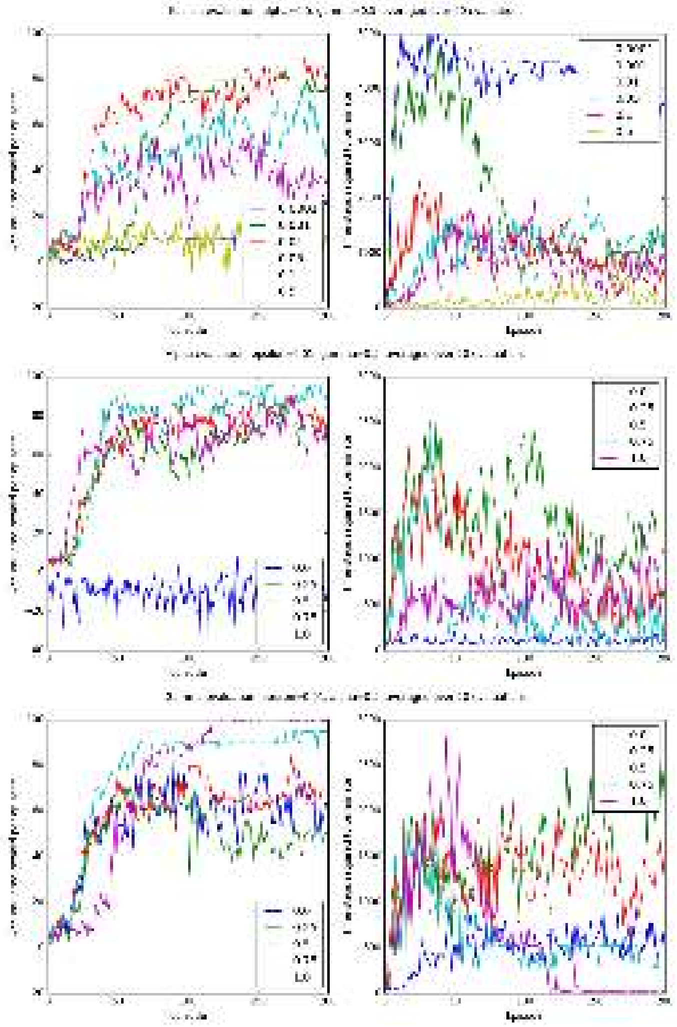


Figure A.12.: IM Model Parameter Evaluation for Scenario 1

The same definitions as in Figure A.1 apply. The highest reward is reached with $\epsilon = 0.01, \gamma = 0.75, \alpha = 1.0$

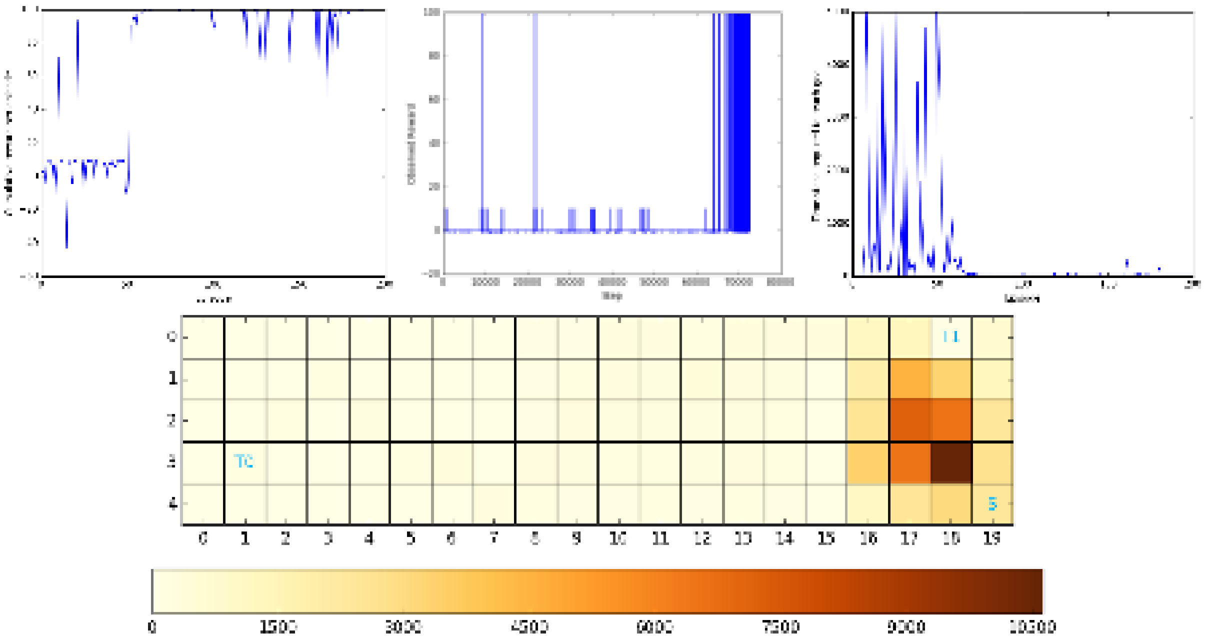


Figure A.13.: IM Model Evaluation Details on Scenario 1

One evaluation of running our IM approach on scenario 1. The SARSA parameters are set to $\epsilon = 0.01, \alpha = 0.75, \gamma = 1.0$, and we use 200 episodes of learning. After about 50 episodes of exploration and learning to reach $T1$, the negative intrinsic reward becomes high enough to push the agent towards $T0$. Though the heat-map shows the agent is mainly exploring east of the first wall.

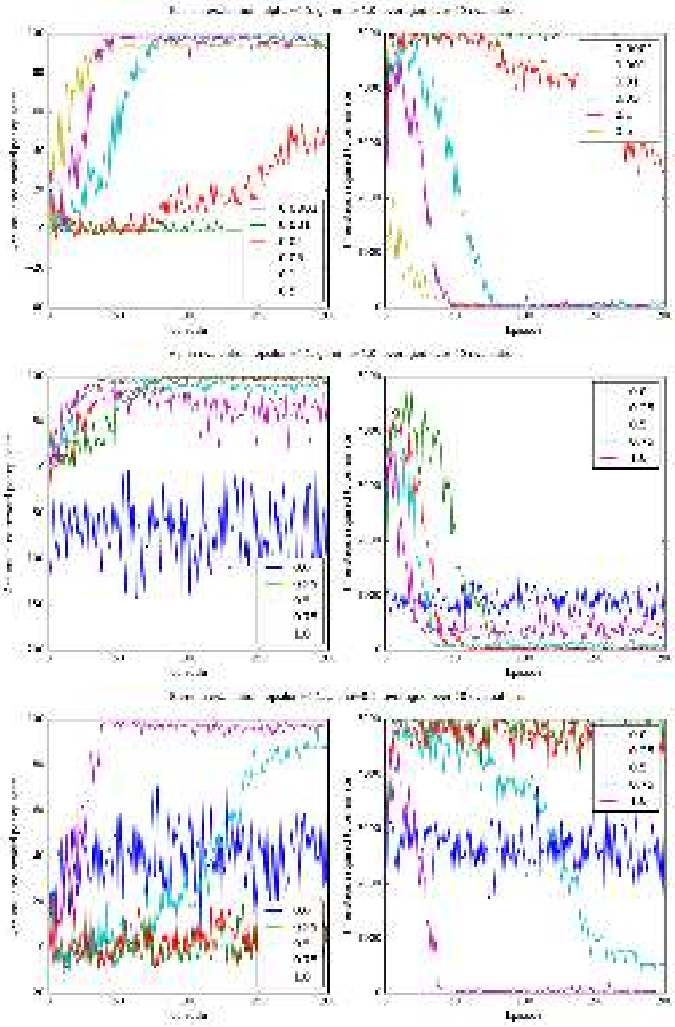


Figure A.14.: IM Model Parameter Evaluation for Scenario 2

The same definitions as in Figure A.1 apply. Convergence is reached best with a set of $\epsilon = 0.1, \alpha = 0.5, \gamma = 1.0$.

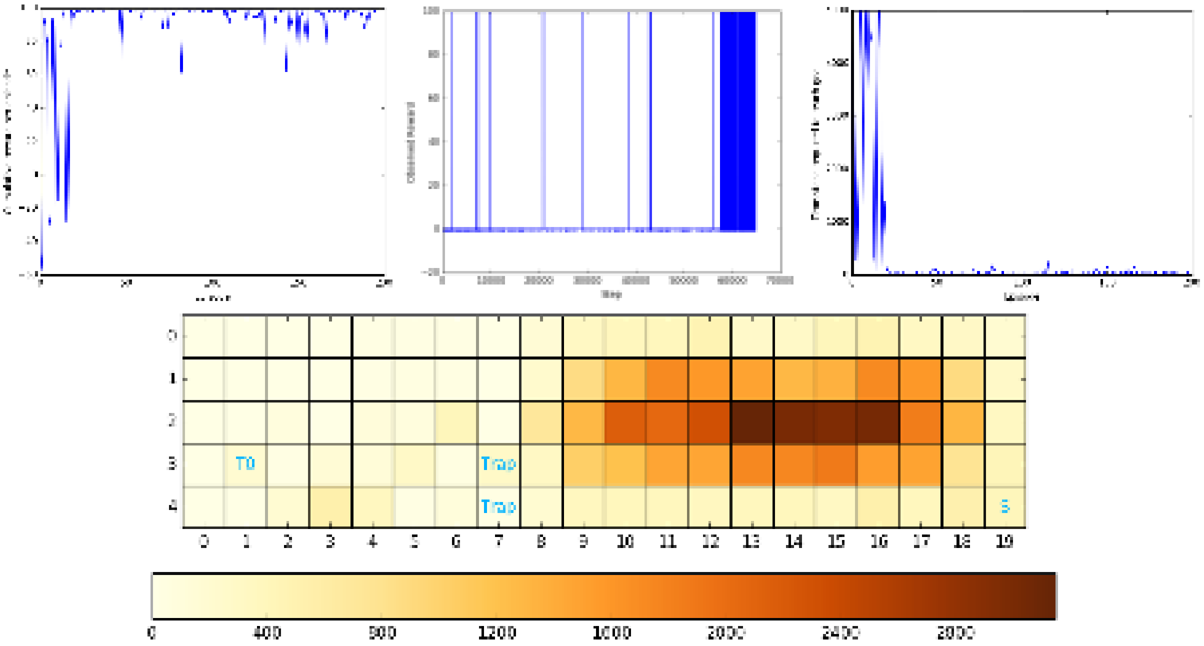


Figure A.15.: IM Model Evaluation Details on Scenario 2

One evaluation of running our IM approach on the second scenario with traps. The same definitions as in Figure A.2 apply and the SARSA parameters are set to $\epsilon = 0.1, \alpha = 0.5, \gamma = 1.0$, and we use 200 episodes of learning. Convergence is reached before episode 50, but it is more unstable than using TDE and the visit history does not show a path through the grid-world.

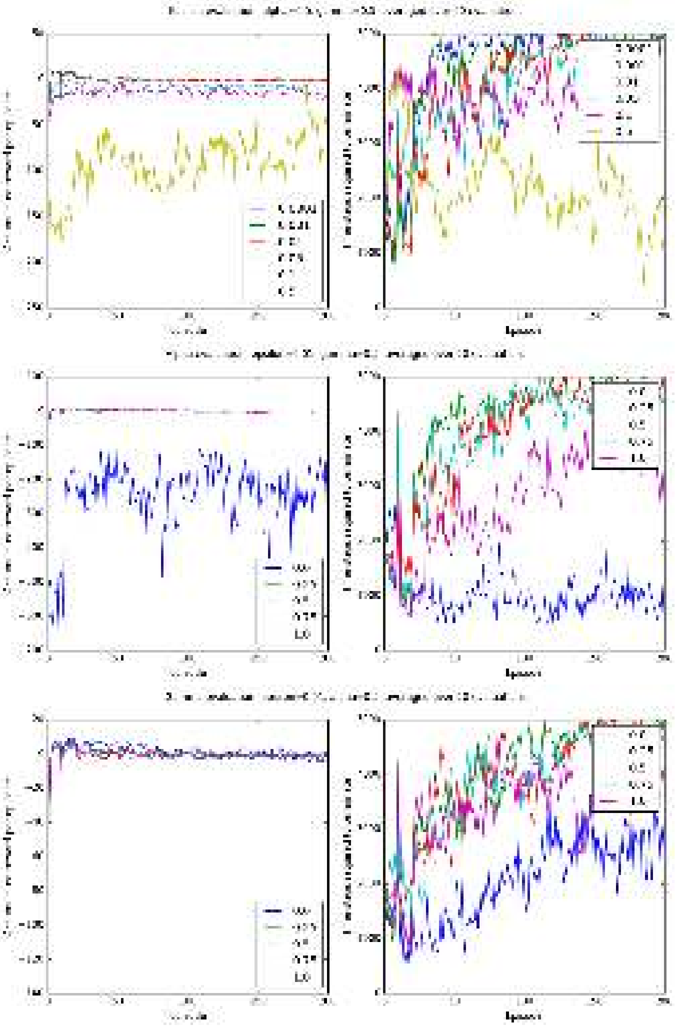


Figure A.16.: IM Model Parameter Evaluation for Scenario 3

The same definitions as in Figure A.1 apply. Convergence is reached best with $\epsilon = 0.01, \alpha = 1.0, \gamma = 0.0$.

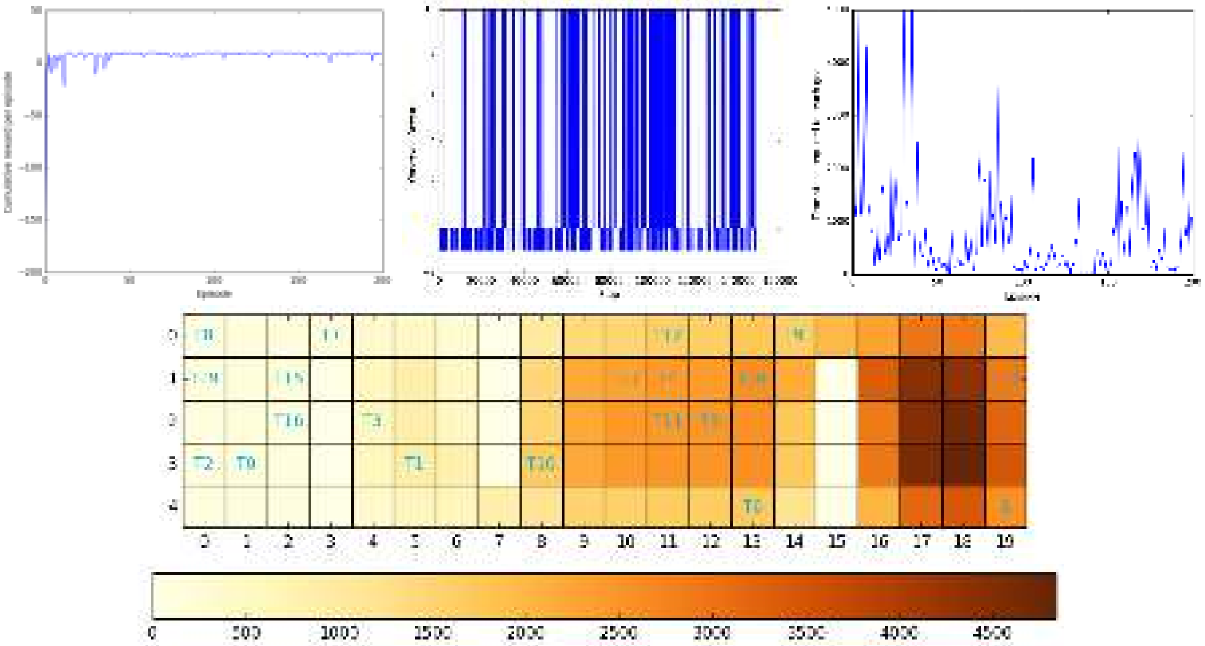


Figure A.17.: IM Model Evaluation Details on Scenario 3

One evaluation of running our IM approach on scenario 3 with randomly repositioning terminal states, using the same definitions as in Figure A.2. The SARSA parameters are set to $\epsilon = 0.01, \alpha = 1.0, \gamma = 0.0$, and we use 200 episodes of learning. The behavior is very similar to the one using SARSA (See Figure A.8.

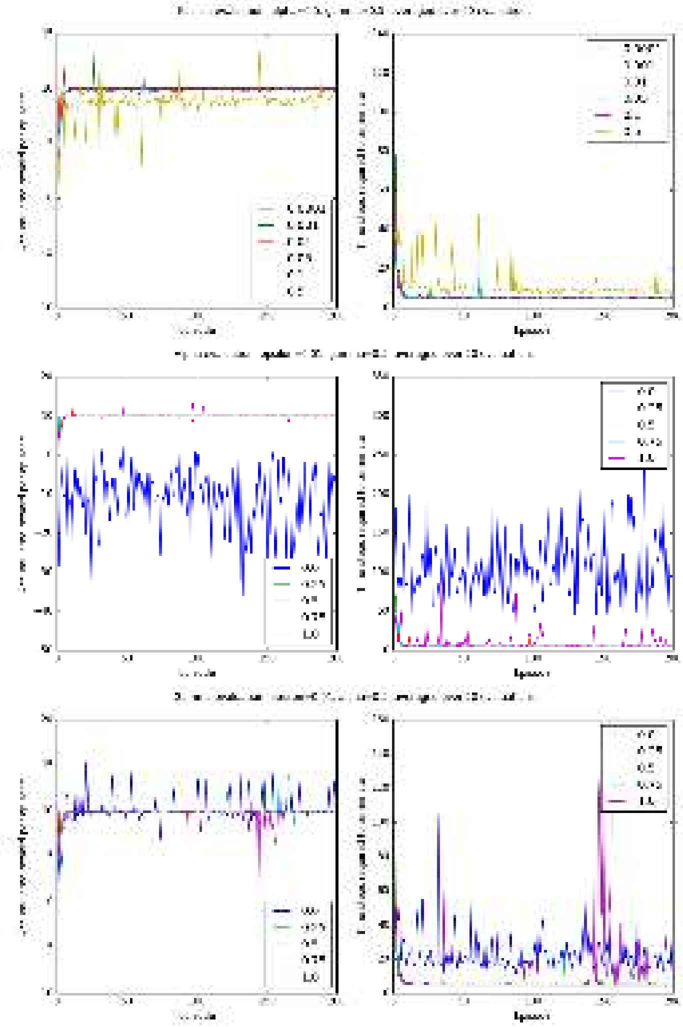


Figure A.18.: CM Model Parameter Evaluation for Scenario 1

The same definitions as in Figure A.1 apply. Convergence is reached best for $epsilon 0.1$, while α and γ seem to have no influence, unless they are set higher than 0.

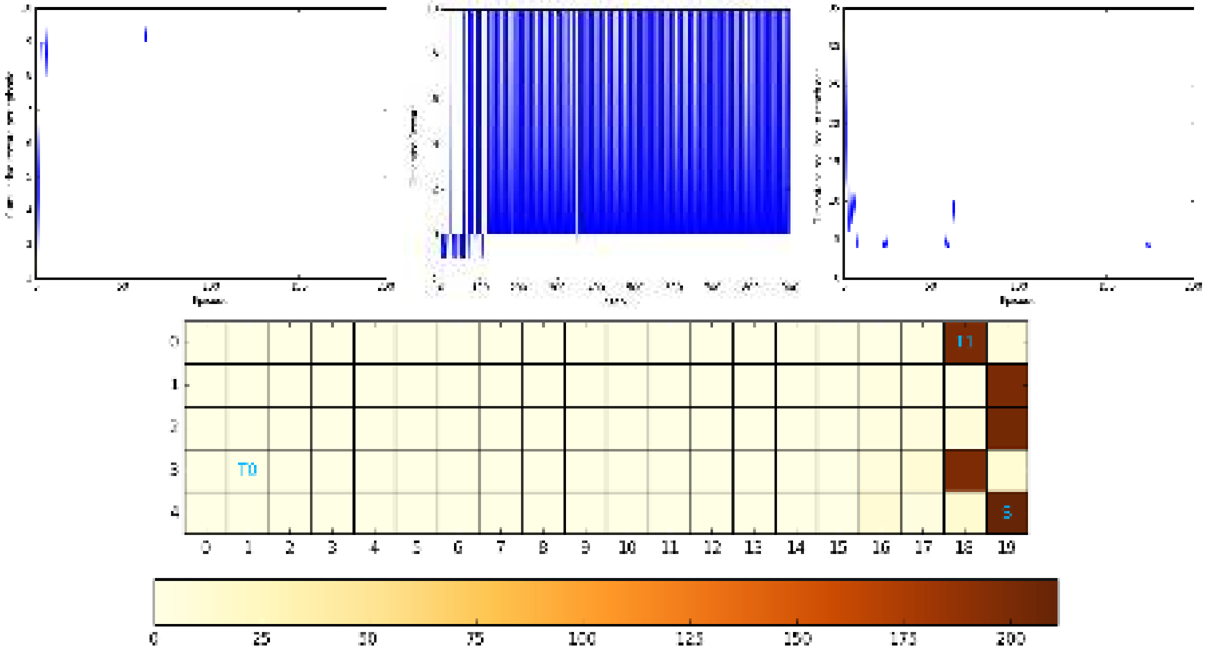


Figure A.19.: CM Model Evaluation Details on Scenario 1

One evaluation of running our CM approach on scenario 1 with two terminal states. Same definitions as in Figure A.2 apply. The SARSA parameters are set to $\epsilon = 0.01, \alpha = 0.5, \gamma = 0.25$, and we use 200 episodes of learning. The agent learns to reach $T1$ following an optimal path.

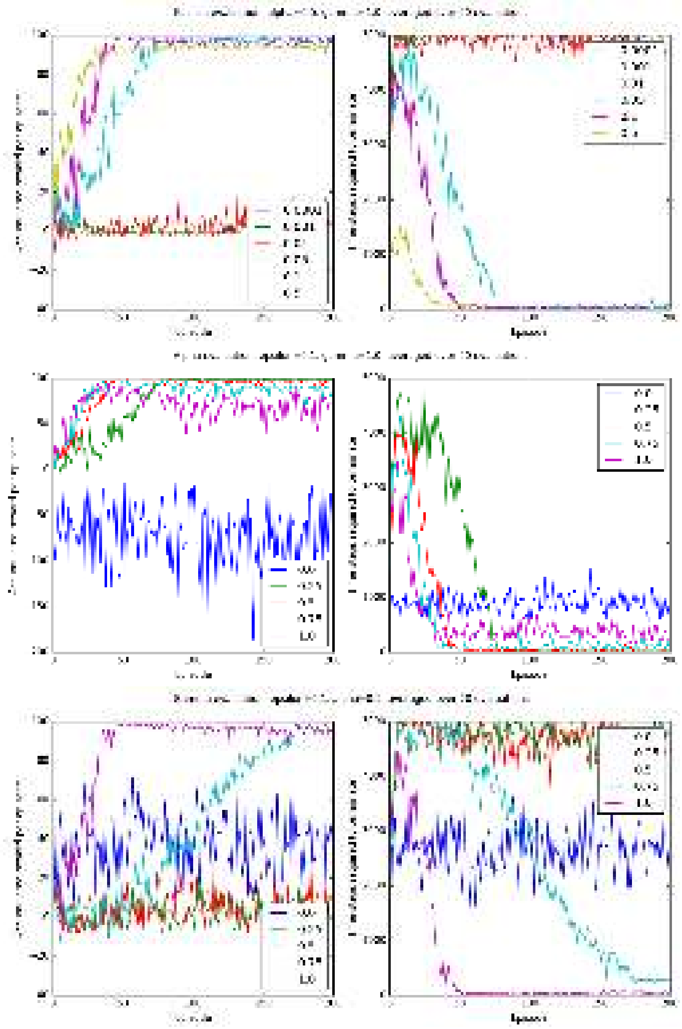


Figure A.20.: CM Model Parameter Evaluation for Scenario 2

Same definitions as in Figure A.1 apply. Convergence is again reached best with $\epsilon = 0.1, \alpha = 0.5, \gamma = 1.0$.

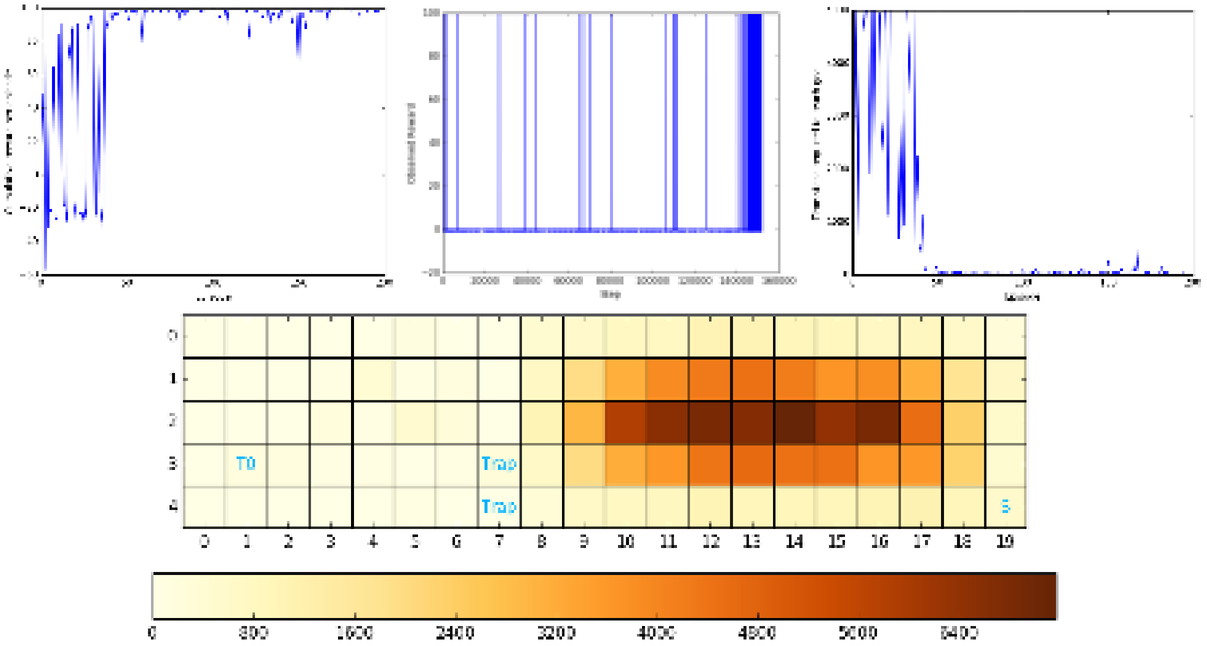


Figure A.21.: CM Model Evaluation Details on Scenario 2

The same definitions as in Figure A.2 apply. One evaluation of running our CM approach on scenario with traps. The SARSA parameters are set to $\epsilon = 0.1, \alpha = 0.5, \gamma = 1.0$, and we use 200 episodes of learning. The agent converges towards the optimum regarding the observed reward and amount of time-steps needed to terminate at about 50 episodes of learning, but many time-steps are again spent exploring east of the wall.

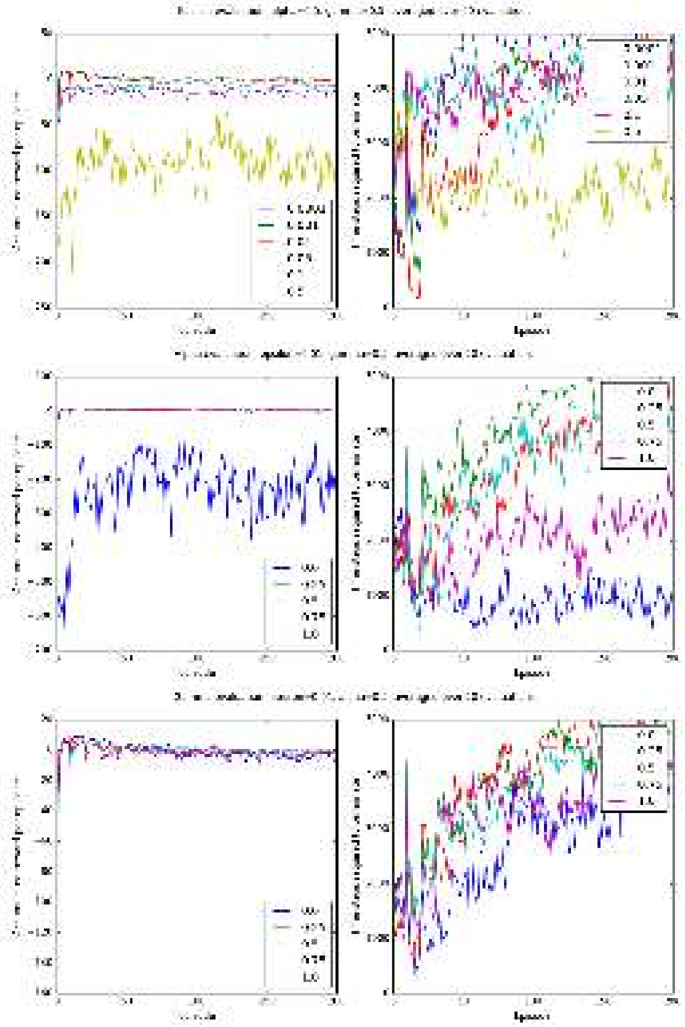


Figure A.22.: CM Model Parameter Evaluation for Scenario 3

This figure uses the same definitions as Figure A.1. The agent converges best with the parameters set to $\epsilon = 0.01, \alpha = 1.0, \gamma = 0.0$.

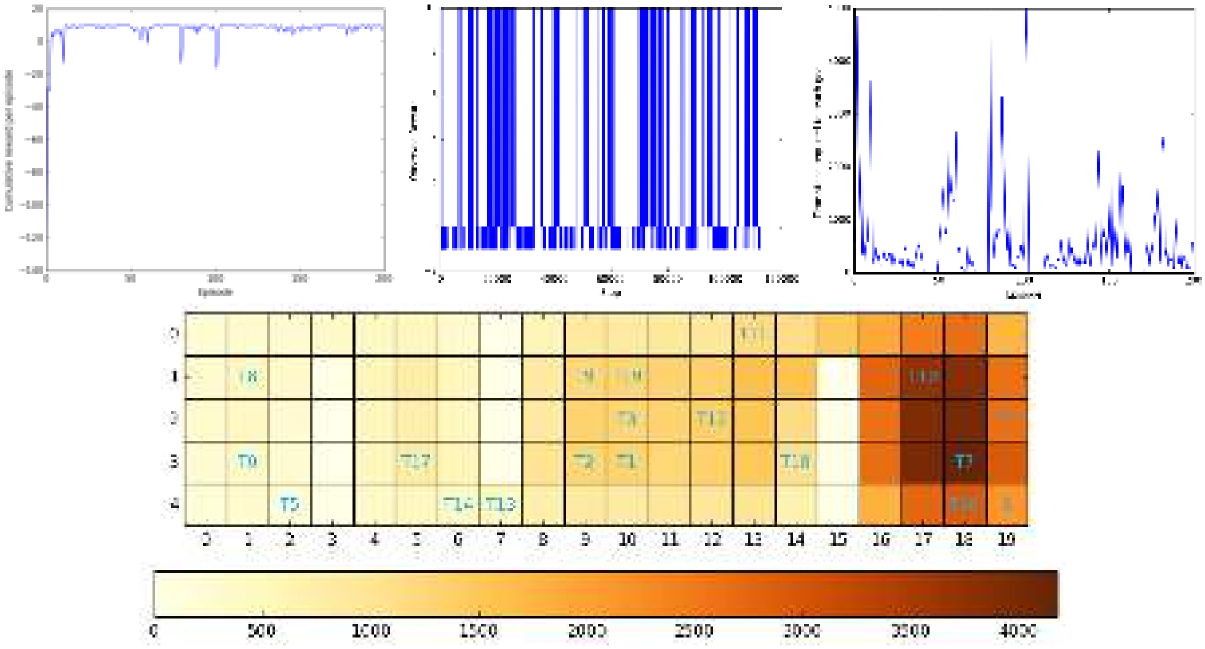


Figure A.23.: CM Model Evaluation Details on Scenario 3

One evaluation of running our CM approach on scenario 3 with randomly repositioning terminal states. This figure uses the same definitions as Figure A.2. The SARSA parameters are set to $\epsilon = 0.01, \alpha = 1.0, \gamma = 0.0$, and we use 200 episodes of learning. The agents behavior is almost optimal regarding the reward and the amount of times the target position is repositioned, but takes surprisingly long to terminate in many episodes and does surprisingly often observe the reward of -1. Exploration is mainly done east of the first wall.

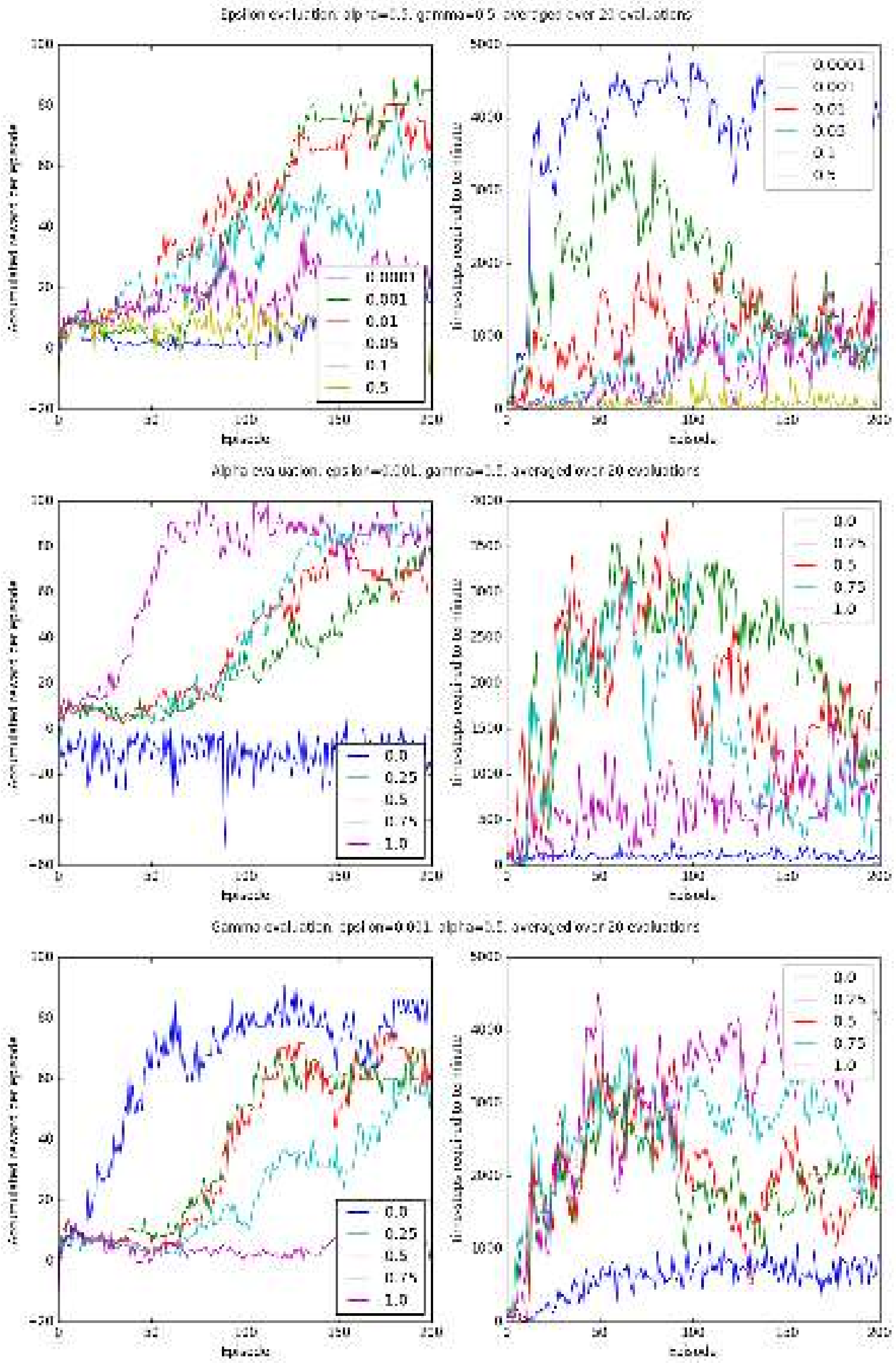


Figure A.24.: CPM Model Parameter Evaluation for Scenario 1

This figure uses the same definitions as Figure A.1 and indicates the best performance for a parameter set of $\epsilon = 0.001, \alpha = 0.75, \gamma = 0.0$.

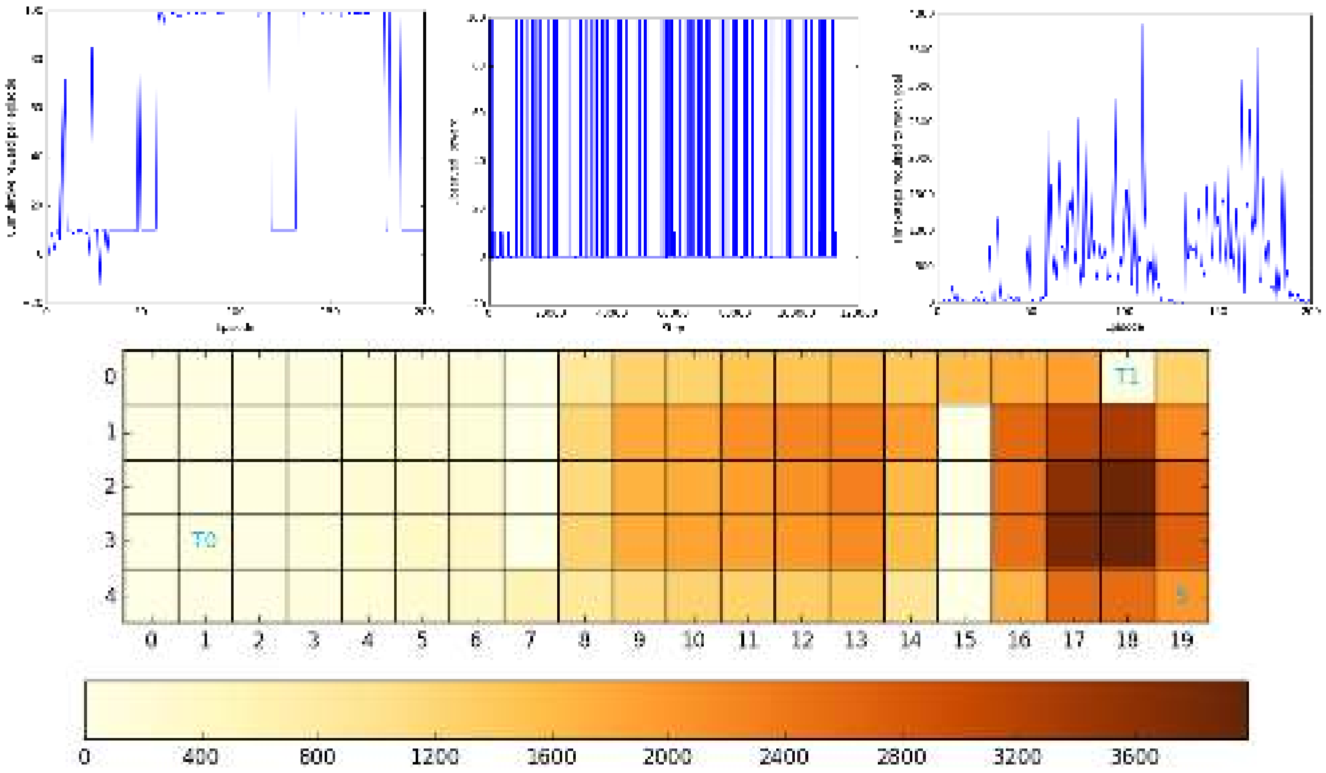


Figure A.25.: CPM Model Evaluation Details on Scenario 1

The same definitions as in Figure A.2 apply. One evaluation of running our CPM approach on the first scenario with two terminal states. The SARSA parameters are set to $\epsilon = 0.001, \alpha = 0.75, \gamma = 0.0$, and we use 200 episodes of learning. One can see, the agent is learning to reach $T1$ for about 50 episodes and then switches to $T0$, providing the higher external reward of +100. It gets interrupted at about 120 episodes of learning and at about 180 episodes of learning, where it gets back to learning to reach $T1$. This interruptions result in more exploration, which takes part mainly est of the second wall from the agent's starting position S.

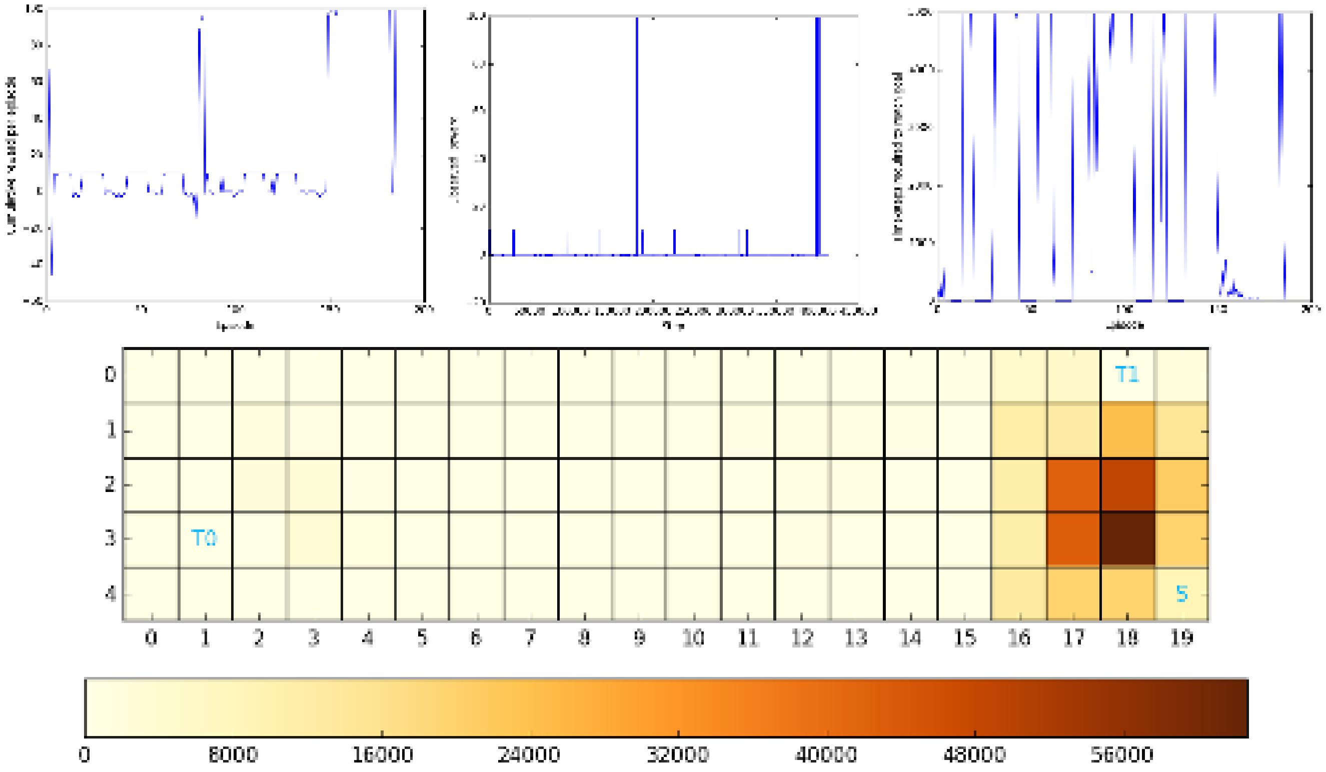


Figure A.26.: CPM Model Evaluation 2 Details on Scenario 1

Another evaluation of running our CPM approach on the first scenario with two terminal states. The SARSA parameters are set to $\epsilon = 0.001, \alpha = 0.75, \gamma = 0.5$, and we use 200 episodes of learning. We can now observe less exploration between the first and second wall from the start position S. The agent is interrupted from learning to reach $T1$ in longer phases and starts learning to reach $T0$ not until episode 150, but the heat-map in Figure **??** does show a more clear path through the grid-world.

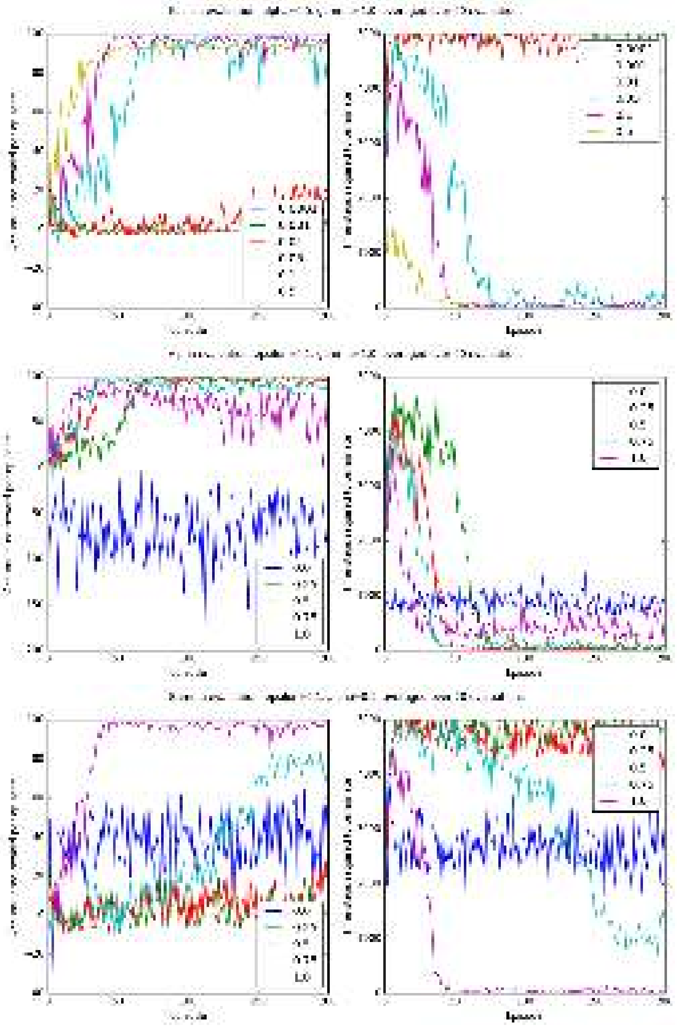


Figure A.27.: CPM Model Parameter Evaluation for Scenario 2

This figure uses the definitions given in Figure A.1. Convergence is reached best with $\epsilon = 0.1, \alpha = 0.5, \gamma = 1.0$.

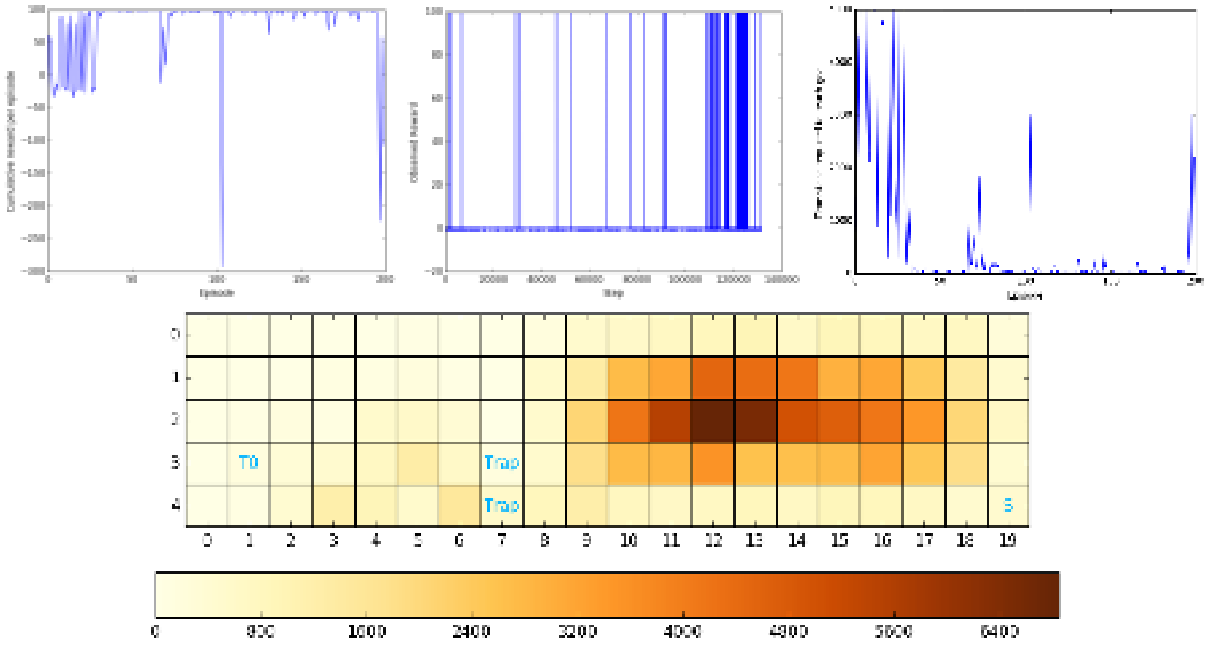


Figure A.28.: CPM Model Evaluation Details on Scenario 2

One evaluation of running our CPM approach on the second scenario with traps.The figure uses the same definitions we use in Figure A.2. The SARSA parameters are set to $\epsilon = 0.1, \alpha = 0.5, \gamma = 1.0$, and we use 200 episodes of learning. The results are very similar to the other approaches on this scenario, but the convergence gets interrupted in some episodes (e.g. episode 100), which results in a slightly higher amount of exploration.

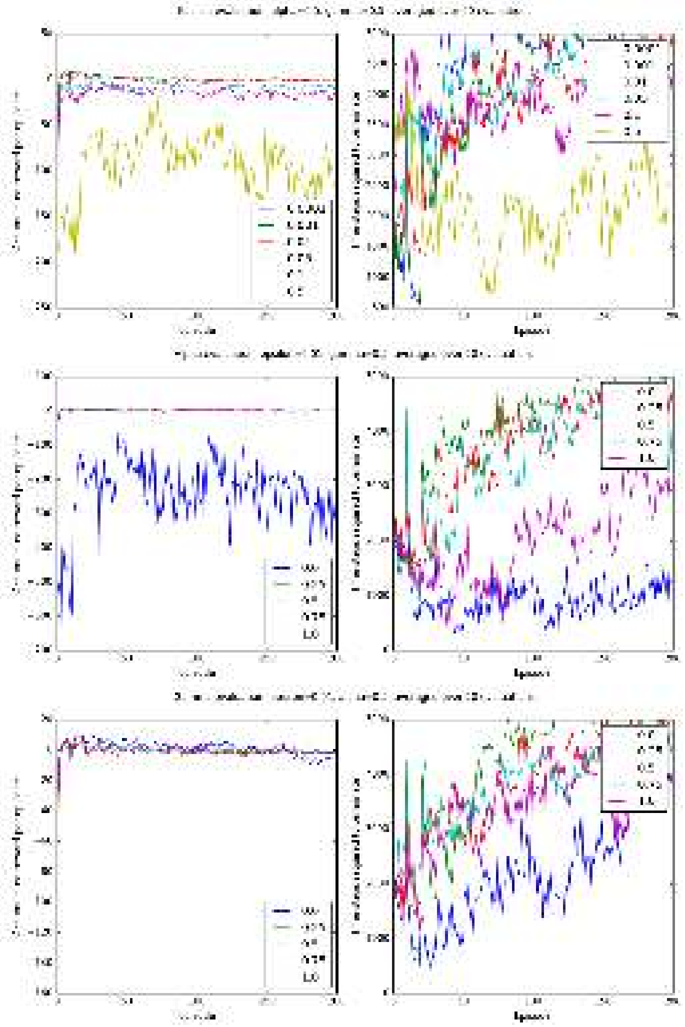


Figure A.29.: CPM Model Parameter Evaluation for Scenario 3

The same definitions as in Figure A.1 apply. The best performance is reached with a parameter set of $\epsilon = 0.01, \alpha = 1.0, \gamma = 0.0$.

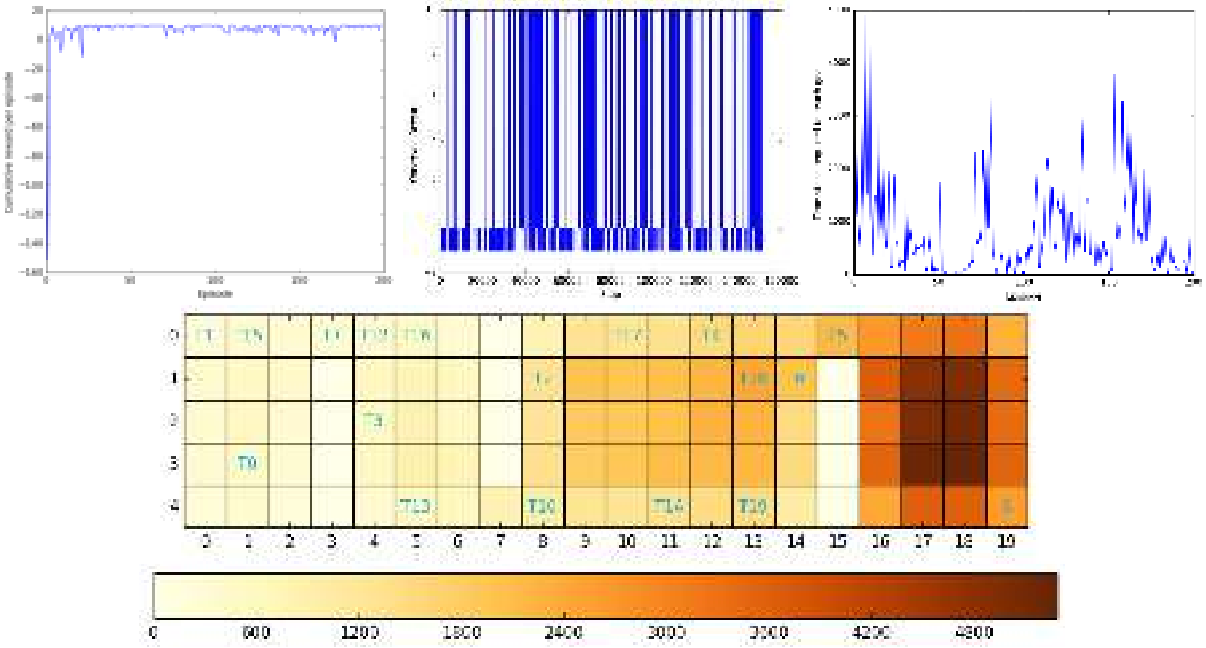


Figure A.30.: CPM Model Evaluation Details on Scenario 3

One evaluation of running our CPM approach on the first grid-world with randomly repositioning terminal states. The same definitions as in Figure A.2 apply. The SARSA parameters are set to $\epsilon = 0.01, \alpha = 1.0, \gamma = 0.0$, and we use 200 episodes of learning. One can again observe a good performance regarding the external reward, but surprisingly many time-steps needed to terminate in many episodes and an surprisingly high amount of observed rewards of -1. Most exploration happens eastern of the second wall from the agent's starting position S.

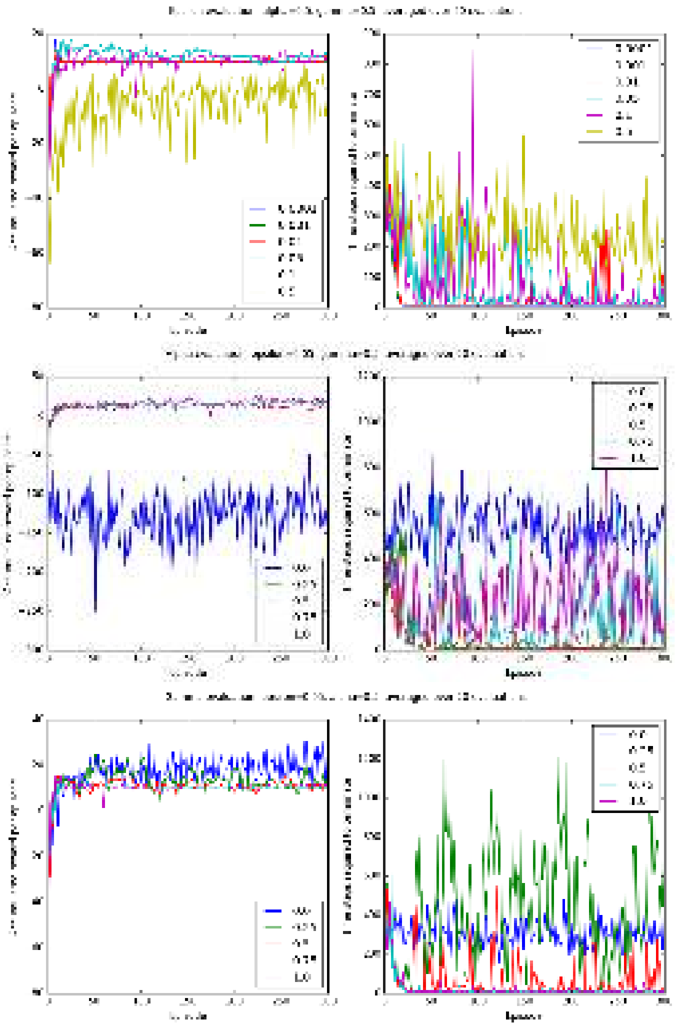


Figure A.31.: SARSA Parameter Evaluation for Scenario 4

The same definitions as in Figure A.1 apply. Convergence is only achieved with $\epsilon \leq 0.01, \alpha \in [0.5, 0.25], \gamma \in [0.75, 1.0]$, where smaller values for γ result in a bit higher mean reward, but also in much more time-steps needed to terminate.

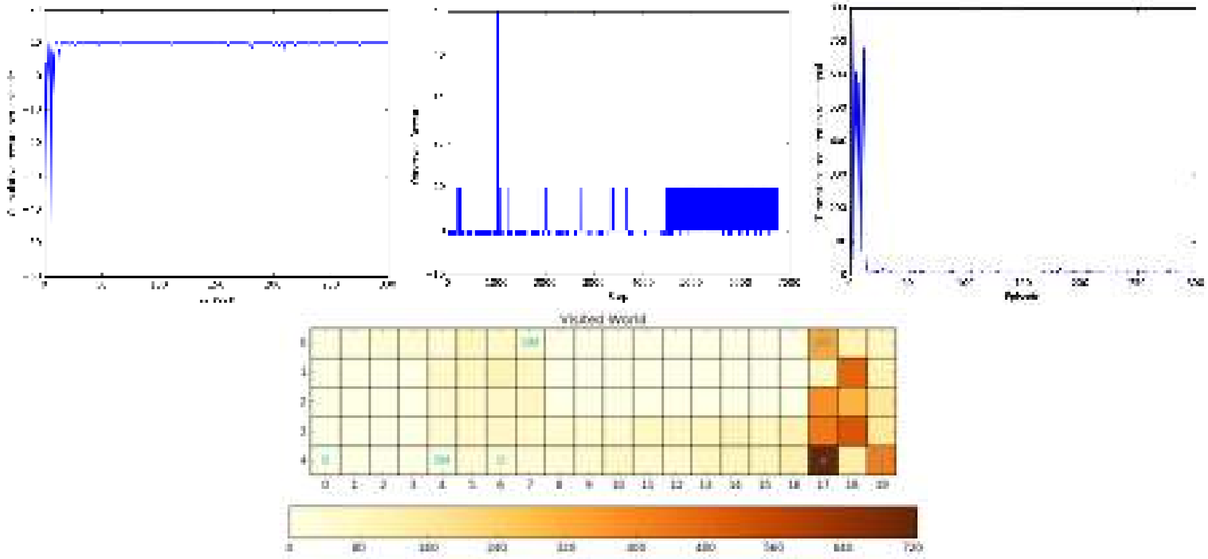


Figure A.32.: SARSA Evaluation Details on Scenario 4

One evaluation of running the classic SARSA algorithm on the market domain. This figure uses the same definitions as Figure A.2. The SARSA parameters are set to $\epsilon = 0.05, \alpha = 0.75, \gamma = 0.75$, and we use 300 episodes of learning. One can see the agent is converging towards bringing the apple to the apple merchant in less than 25 episodes. The remaining environment is explored very little.

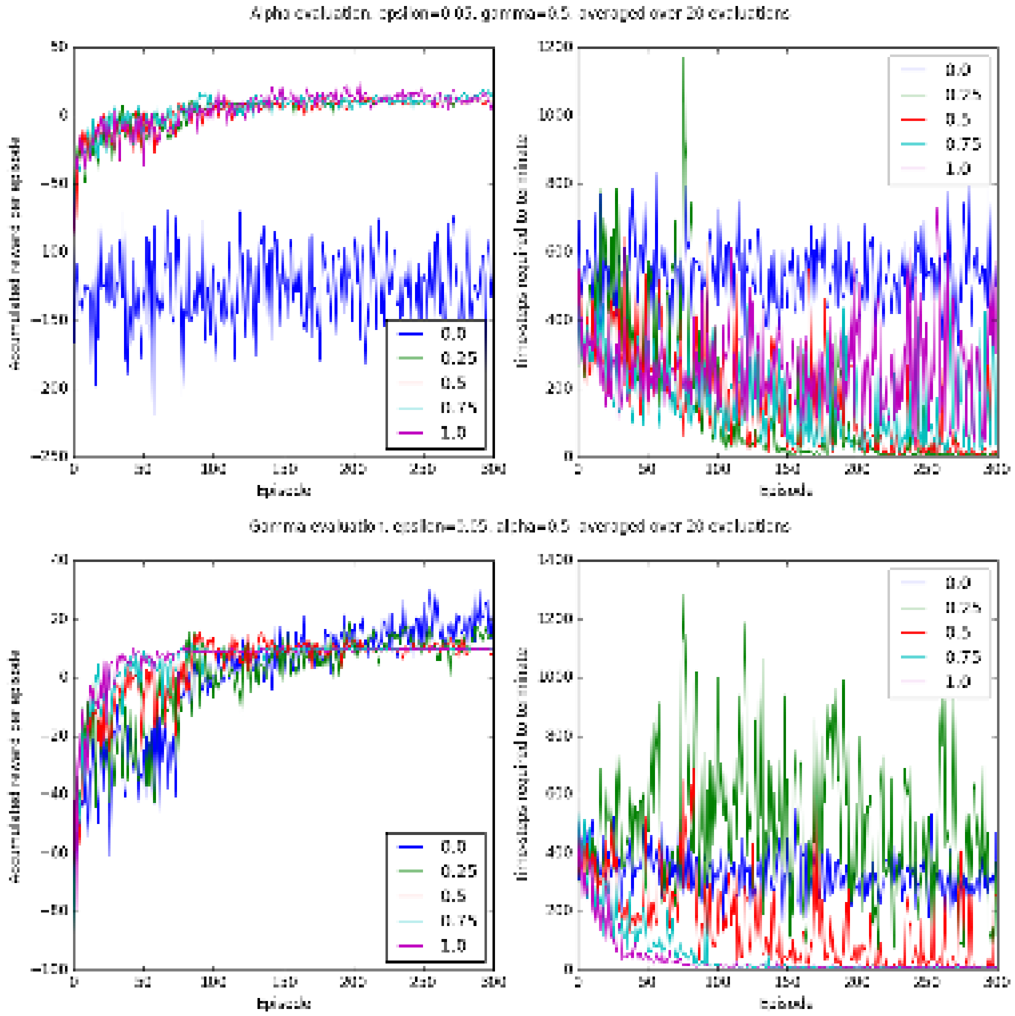


Figure A.33.: TDE Parameter Evaluation for Scenario 4

The definitions in Figure A.1 are used for this figure as well. The most stable performance is achieved with $\alpha \in [0.25, 0.5]$ and $\gamma \in [0.75, 1.0]$

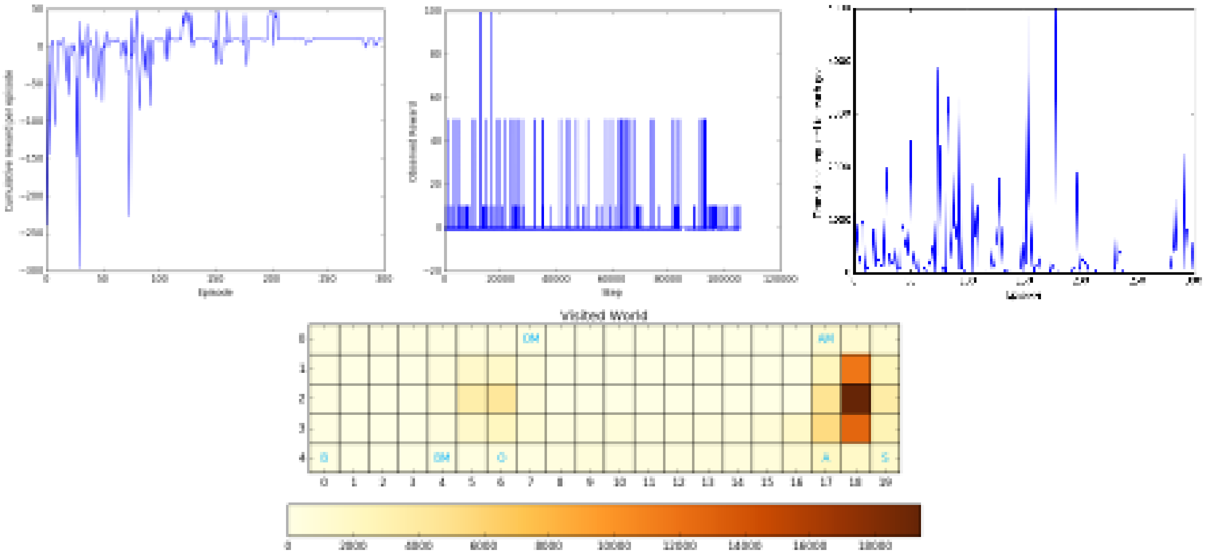


Figure A.34.: TDE Model Evaluation Details on Scenario 4

One evaluation of running the TDE approach on the market domain. The same definitions as in Figure A.2 apply. The SARSA parameters are set to $\alpha = 0.5, \gamma = 0.75$, and we use 300 episodes of learning. The agent needs about 200 episodes of learning until it converges towards bringing the apple to the apple merchant, where it observes a reward of +10. Due to the random exploration, it also manages to bring the orange to the orange merchant quite often, where it observes a reward of +50. It even manages to bring the banana to the merchant a few times and observe a reward of +100. While it converges regarding the observed reward, it needs a lot of time-steps to terminate in some episodes.

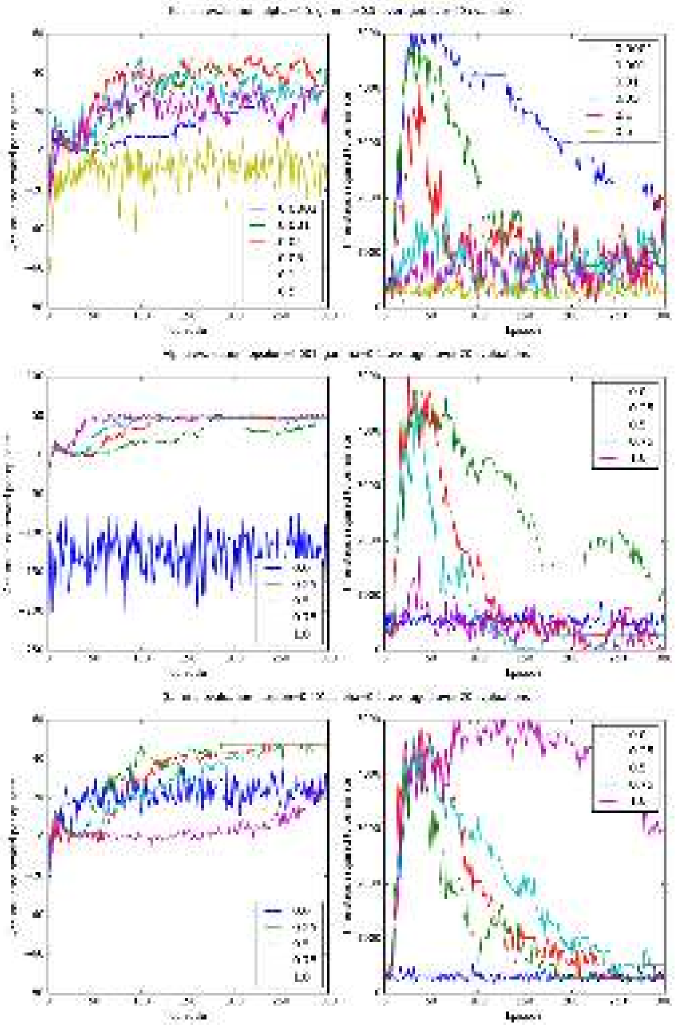


Figure A.35.: IM Model Parameter Evaluation for Scenario 4

The same definitions as in Figure A.1 apply. Convergence towards the reward of 50 is achieved best with the parameters set to $\epsilon = 0.001, \alpha = 0.75, \gamma = 0.25$.

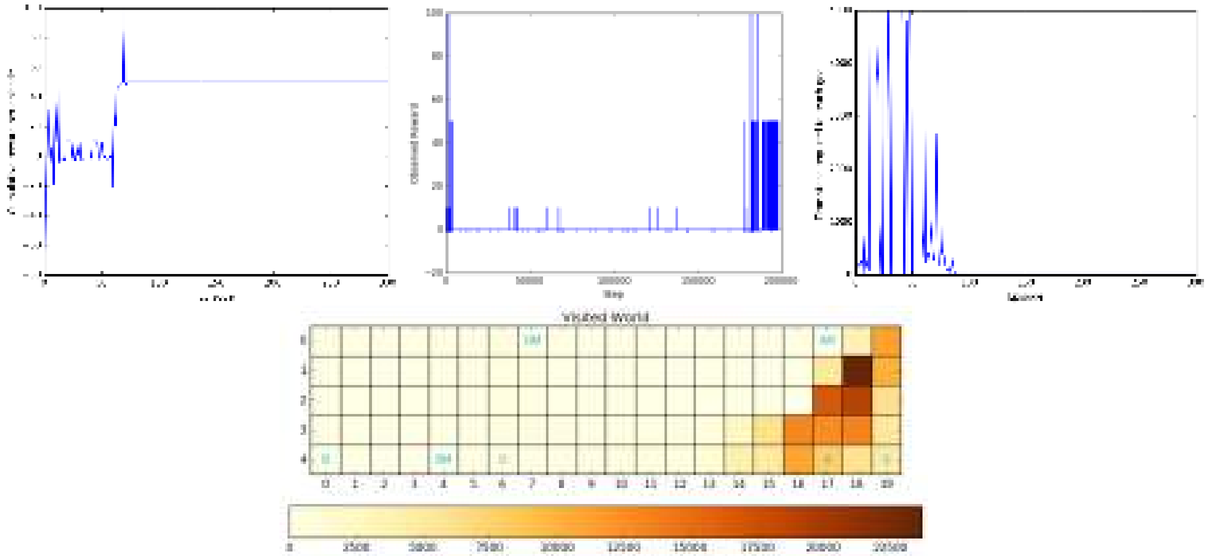


Figure A.36.: IM Model Evaluation Details on Scenario 4

One evaluation of running our IM approach on the market domain. The same definitions as in Figure A.2 apply, the SARSA parameters are set to $\epsilon = 0.001, \alpha = 0.75, \gamma = 0.25$, and we use 300 episodes of learning. After about 100 episodes of exploring and learning to get rewarded by the apple merchant, the agent converges towards the orange merchant providing the higher reward of +50. It even manages to observe a reward of +100 due to the banana a few times. Though most of the time-steps are still spent around the starting position S, the apple A and the apple merchant AM in the east of the grid-world.

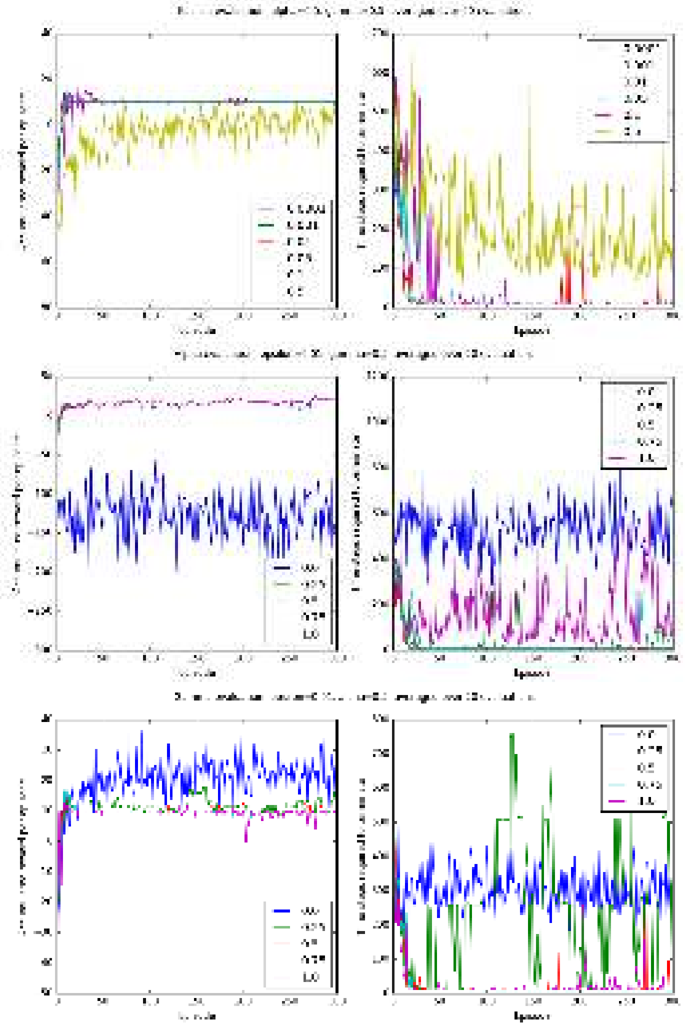


Figure A.37.: CM Model Parameter Evaluation for Scenario 4

The same definitions as in Figure A.1 apply. Convergence is achieved best with the parameters set to $\epsilon = 0.01, \alpha = 0.25, \gamma = 0.5$.

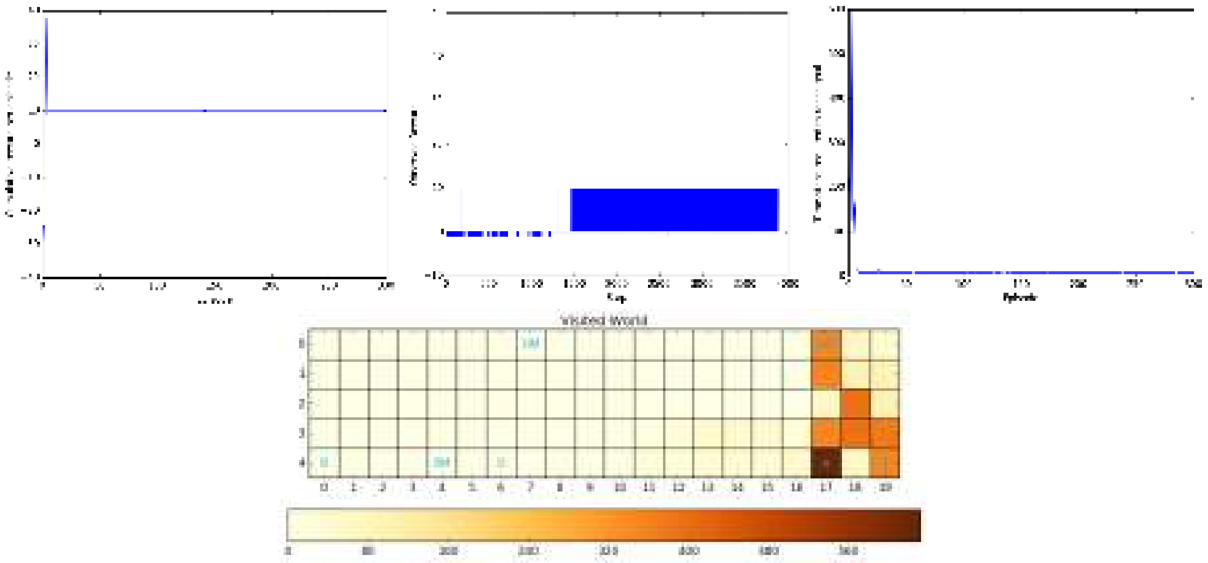


Figure A.38.: CM Model Evaluation Details on Scenario 4

One evaluation of running our CM approach on the market domain. This figure uses the same definitions as Figure A.2. The SARSA parameters are set to $\epsilon = 0.01, \alpha = 0.25, \gamma = 0.5$, and we use 300 episodes of learning. The heat-map shows, the agent now explores even less than with the classic SARSA approach. It converges towards the reward of +10, observed by picking up the apple A and bringing it to the apple merchant AM, in about 10 episodes of learning.

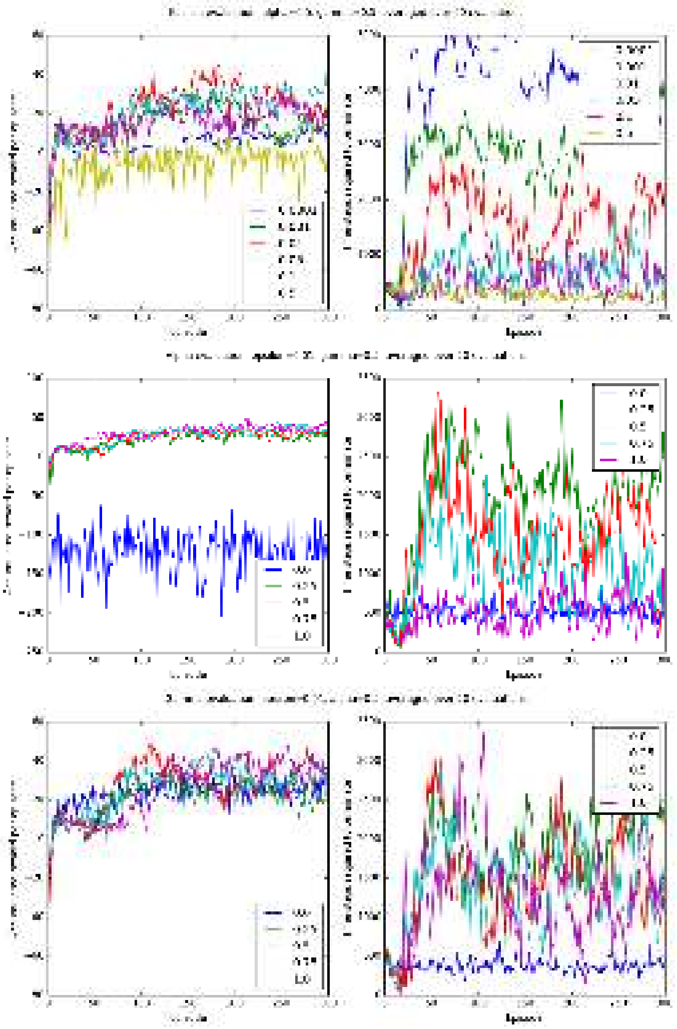


Figure A.39.: CPM Model Parameter Evaluation for Scenario 4

This figure uses the same definitions as Figure A.1. The performance of CPM looks very unstable for every set of parameters, but we choose use $\epsilon = 0.01, \alpha = 0.75, \gamma = 0.75$ as the best possible set.

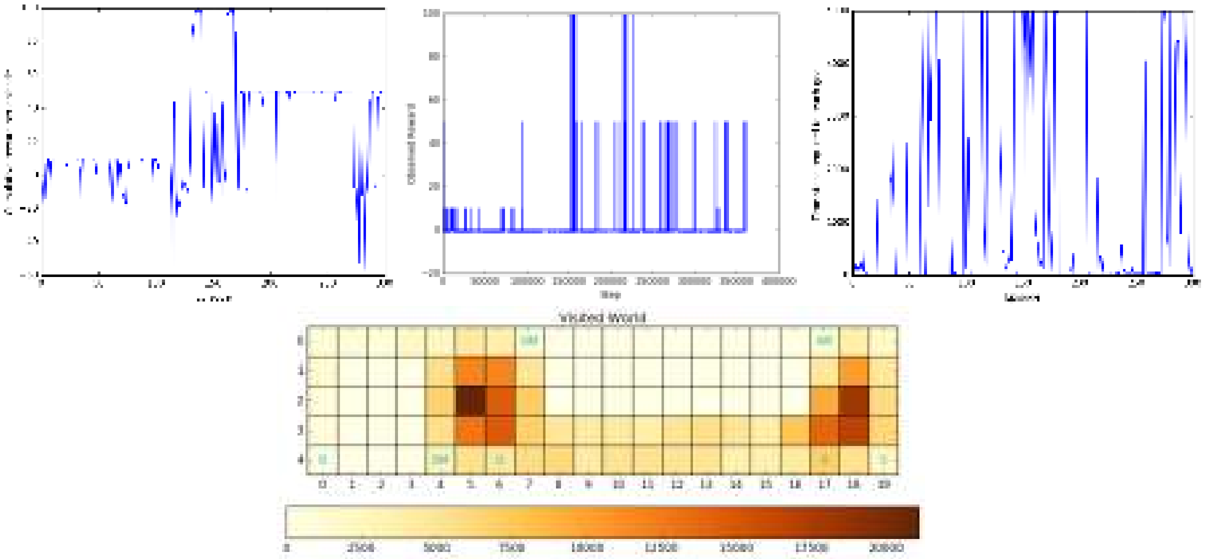


Figure A.40.: CPM Model Evaluation Details on Scenario 4

One evaluation of running our CPM approach on the market domain. The same definitions as in Figure A.2 apply. The SARSA parameters are set to $\epsilon = 0.01, \alpha = 0.75, \gamma = 0.75$, and we use 300 episodes of learning. One can now observe very unstable results, but the agent is even trying to learn to get rewarded by the banana merchant (r=+100) between episodes 100 and 200. Before that, it learns to get rewarded by the apple merchant (r=+10). After 200 episodes, it focuses on learning to pick up the orange and bring it to the orange merchant (r=+50). Theses phases of learning are interrupted by phases with no competence progress and result in more exploration and more time-steps needed to terminate. The heat-map shows, the agent is now exploring much more of its environment.

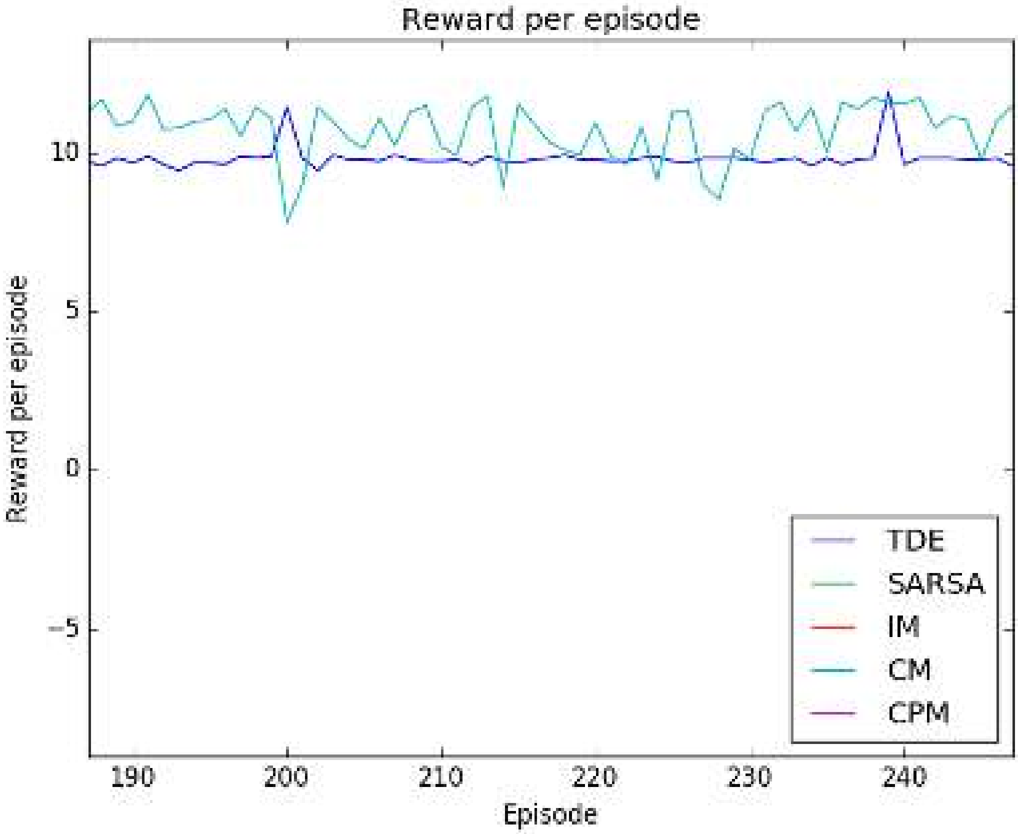


Figure A.41.: Example for Spiky Results

Example for spiky results of our algorithm due to the randomness of SARSA and our definition of competence. These results could possibly be smoothed by defining a ranged level of misachievement, instead of a discrete value.