# Policy Based Deceptive Reinforcement Learning

**Abhinav Singh**
**1037729**
**abhinavkumar@student.unimelb.edu.au**

**Ghawady Ehmaid**
**983899**
**gehmaid@student.unimelb.edu.au**

**COMP90055 - Research Project**
**Supervisor: Tim Miller**

**The University of Melbourne**

# SIGNED DECLARATION

We Abhinav Singh and Ghawady Ehmaid certify that: This thesis does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any university; and that to the best of our knowledge and belief it does not contain any material previously published or written by another person where due reference is not made in the text. Where necessary we have received clearance for this research from the Universityś Ethics Committee and have submitted all required data to the School. The thesis is 7200 words in length (excluding text in images, table, bibliographies and appendices).

## Abstract

We study the problem of deceptive reinforcement learning to preserve the privacy of an agent's real goal. Reinforcement learning is a computational framework that helps in finding a policy based on rewards received from the exploratory behavior of the agent in the given environment. An indispensable constituent in the reinforcement learning paradigm is a *reward function*, which is solely responsible for generating rewards based on current state and action. The reward function epitomizes the real objective of the agent, therefore in some situations it is paramount to protect its privacy thereby making it difficult for an adversarial observer to determine the real goal of the agent. We define the problem of privacy protection reinforcement learning and present two models for solving it. Both these models are based on *dissimulation*–a form of deception that 'hides the truth'. The models are evaluated both computationally and via human behavioral evaluations. Results indicate that the resulting policies are indeed deceptive, and the participants could determine the agent's real goal less reliably than that of an honest agent.

## 1 Introduction

In this report, we discuss the problem of deceptive policy reinforcement learning for preserving the privacy of an agent's real intention. Reinforcement learning is a computational framework that helps in finding a policy based on rewards received from the exploratory behavior of the agent in the given environment [Sutton and Barto, 2018]. Recent advances in the field of reinforcement learning have led to a surge in interest from the research community [Mnih *et al.*, 2015]. Researchers are actively working to implement reinforcement learning based methods to solve different problem domains. An indispensable constituent in the reinforcement learning paradigm is a *reward function*, which is solely responsible for generating rewards based on current state and action. The reward function epitomizes the real objective of the agent; therefore, in some situations, it is paramount to protect its privacy thereby, making it difficult for an adversarial observer to determine the real goal of the agent.

Masters and Sardina [2017b] presents an example of a convoy escorting a VIP to one of the secret safe houses in the presence of an adversarial observer who plans to deploy an assassin to one of the hidden safe houses. The agent's objective is to delay the critical information about the real destination from reaching the adversary. In the reinforcement learning framework, protecting an agent's real goal is equivalent to hiding the reward or objective function, which was used to learn an agent's behavioral policy.

A behavior is considered deceptive if it provides misleading information to hide the truth [Bell, 2003]. Bell decomposes deception into two general types: *simulation*, in which someone 'presents the false' to entice observer to believe something that is not true; and *dissimulation*, in which someone 'hides the truth' to avoid revelation of critical information to the adversarial observer.

The majority the models proposed in recent years [Masters and Sardina, 2017b; Keren *et al.*, 2016; Kulkarni *et al.*, 2018], are based around the concept of preserving privacy using dissimulation. However, all these approaches are in the context of model-based planning and require analysis of the model's structure to achieve dis-

simulation; therefore, do not apply to model-free methods such as model-free reinforcement learning. One of the most recent work, [Yang *et al.*, 2020], have attempted to address this gap using Q-learning. However, addressing this area using policy-based reinforcement learning is still an uncharted territory.

In this report, we propose general models for hiding the real objective of the agent using dissimulation, which applies to model-free MDPs. We present two methods: one based on indifference, in which the agent treads in the region of indifference which is formed by the intersection of the circle of maximum probability of all the possible goals; and other one based on Policy-based learning which uses single reward function to incorporate deception in the normal learning process thereby eliminating the requirement of using multiple reward functions as shown in Yang *et al.* [2020]

We assess both our models via human evaluation and naïve intention recognition algorithm [Masters and Sardina, 2019], which involves 30 non-naïve participants. The intention algorithm and participants in human evaluation predicted the possibility of all potential goals in a path planning simulation. The outcomes were compared to different models exhibiting deception by using optimality and deception as a metric. The results showed that our agents were efficient at masking their real objective when compared to an honest agent, but the real objective becomes more apparent as the agent gets closer to it.

### 1.1 Problem overview

Based on our knowledge, previous work of deceptive goal achievement used either model-based path planning [Masters and Sardina, 2017b], model-free reinforcement learning to learn separate Q-Function for each real and fake goals, then devise an action selection logic that maximises the entropy from the observer point of view [Yang *et al.*, 2020], or by applying irrational behavior during action selection [Yang *et al.*, 2020].

To date, there has not been sufficient research that examines the use of a generalised policy-based learning method for hiding the real long-term objectives.

We used reinforcement learning algorithms to train an agent to find a path to reach a predefined target goal in a map considering other possible goals that would act as a decoy. Although the methods proposed could be applied to other problems, we chose a map-based domain for our experiments. The maps structure follows the Movingai[1] format, and we implemented the learning methods using the P4 simulator framework[2]. The choice was made to be able to compare our models against previous work done in this domain.

### 1.2 Research Question

In this paper, we study how to devise a generalised model to achieve rational goal deception using Reinforcement Learning, more specifically policy based reinforcement learning. The proposed approaches use model-free reinforcement learning and consider non-deterministic environments. Model-free methods have shown to be a general-purpose tool for learning complex policies from states [Mnih *et al.*, 2015], [Haarnoja *et al.*, 2018] comparing to model-based reinforcement learning. We also aim to minimize the need for task engineering or handcrafted logic to determine the best policy to take at a specific state.

Our approaches are novel by attempting to learn a deceptive policy directly. Once the model learning converges, the agent will determine a rational and deceptive policy with minimal or no additional logic required for action selection.

To address the generalised model, we need to do a suitable reward shaping and then find a way to assess the quality of the agent's deceptive behavior to optimize between two conflicting objectives: the efficiency to reach the goal while hiding the real intention for as long as possible.

In this paper, we assume that there is one target goal that needs to be achieved; however, the approaches proposed can be enhanced to suit the multi-goal scenario.

## 2 Background and Related Work

### 2.1 Reinforcement Learning

Reinforcement learning is a computational framework that helps in understanding and automating goal-based learning and decision making. It is different from other computational frameworks because of its emphasis on learning via direct interaction with the environment rather than relying on supervision or complete models of the environment. Reinforcement learning is the first field that untangled the computational roadblocks that arise while interacting with an environment to achieve long term goals.

Reinforcement learning uses the formal Markov decision process($MDP$) to elucidate the interaction between an agent and its environment in terms of state, actions and rewards. Sutton and Barto expressed MDPs as a discrete time, stochastic process that can be elucidated by a tuple $< S, A, T, \gamma, R >$ where:

- S is a state with a finite discrete set of states $\{s_0, s_1, ...\}$
- A is a finite, discrete set of actions that an agent can perform$\{a_0, a_1, a_2, ...\}$
- T is the state transition probability function which yields the probability of reaching a particular state $s^{'} \in S$ from state $s \in S$ by performing action $a \in A$, $T(s, a, s^{'})$
- $\gamma$ is the discount factor that assumes the value in the range $0 \leq \gamma \leq 1$ depending upon the importance of the future rewards
- R is the reward function which yields reward value on reaching a particular state $s^{'} \in S$ from state

[1] http://www.movingai.com

[2] https://bitbucket.org/ssardina/soft-p4-sim-core

$s \in S$ by performing action $a \in A$, $R(s, a, s^{'})$

An optimal solution for the MDP model is achieved only when the accumulated future discounted reward for every state in the state space is maximized. This can be done by performing the best possible action from every state $s \in S$ thereby receiving maximum reward over an indefinite horizon.

**Q-Learning**

Q-Learning is an off-policy algorithm for temporal difference learning. It is proven that given sufficient training under any policy, the algorithm converges with probability 1 to a close approximation of the action-value function for an arbitrary target policy. Q-Learning algorithm is capable of learning optimal policy even when actions are selected based on a more exploratory or random policy. It is a model-free reinforcement learning approach that doesn't require the agent to be aware of the environment as it learns from its environment via direct interaction. In contrast, a model-based approach involves information regarding the environment to be provided as a part of the agent's learning process. The algorithm for q-learning is presented in Figure 3. The Q values for

```
Initialize Q (s, a) arbitrarily

Repeat (for each episode):

  Initialize s

  Repeat (for each step of episode):

    Choose a from s using policy derived
    from Q

    Take action a, observe r, s'

    Update Q value using equation [3]

    s ← s';

until s is terminal
```

Figure 1: Q-learning Algorithm, Sutton and Barto [2018]

each state-action pair is updated using below mentioned equation:

$$Q_t(s, a) \longleftarrow Q_{t-1}(s, a) + \alpha TD_t(a, s) \tag{1}$$

Where $\alpha$ is the learning rate which lies in the range $0 \leq \alpha \leq 1$ and signifies the importance of recent information over the older one and $\alpha TD_t(a, s)$ is the temporal difference term which encapsulates the change in the Q value for an action state pair. Temporal difference term can be expanded as:

$$TD_t(a, s) = Q_t(s, a) - Q_{t-1}(s, a) \tag{2}$$

The efficiency of Q-learning algorithm depends upon the trade-off between exploration and exploitation. The $\epsilon$-greedy mechanism, where $\epsilon$ signifies the exploration rate, allows the agent to explore unknown states with $\epsilon$ probability rather than constantly exploiting actions with highest Q-values from a particular state.

**Reward Shaping**

Reward shaping is a method of incorporating additional procedural knowledge to the agent during reinforcement learning with the intent of improving the agent's performance. Adequately shaped rewards help the agent in reducing the number of sub-optimal actions by bootstrapping additional rewards for state-action pairs showing positive progress towards the real goal. The information about additional rewards are generally derived from the environment or the historical data.

Reward shaping is an essential part of the reinforcement learning process because by sharing extra information in the form of additional rewards reduces the number of exploratory steps performed while reaching the goal for the first time. Instead of altering the structure of the reinforcement learning algorithm for incorporating shaped rewards, the same effect can be achieved by initializing the Q-tables since actions are selected based on the temporal difference between the Q-values. Koenig [1996] showed the positive impact of initializing Q-values on the efficiency of reaching the desired goal. Further research in Wiewiora [2003] revealed that shaped rewards could be integrated into learning via Q-value initialization and also proved that the impact of initializing the Q-values using a state potential function is analogous to potential-based reward shaping.

Irrespective of the advantages of reward shaping, there is evidence to prove that incorrect implementation of reward shaping may result in performance deterioration leading to divergence from the real goal. Wiewiora showed that an ill-defined reward shaping would result in the agent's convergence to a sub-optimal path that maximizes the shaped reward but diverges from the goal.

**Policy-based Reinforcement Learning**

In contrast to value-based methods, Policy-based methods are targeted to optimize the policy function $\pi$ that maps states to actions directly instead of optimizing the value function. This is done by updating the parameters $\theta$ of the policy $\pi(a|s;\theta)$ via gradient ascent on the expectation of $R_t$, $\mathbb{E}[R_t]$ (i.e. to increase the expected Return). One of the well known example of this is the REINFORCE algorithm [Williams, 1992]. In this algorithm the policy parameters $\theta$ are updated in the direction of gradient $\nabla_\theta \mathbb{E}[R_t]$, which is estimated by the score function of the log likelihood of actions; $\nabla_\theta \log \pi(a_t|s_t;\theta) R_t$. This approach is unbiased, however it does have a high variance due to the way the $R_t$ is estimated, from sampling, and the possible wide changes in the results. As per Williams, this variance can be reduced by introducing a baseline function $b_t(s_t)$ that depends on the state regardless of the action. This function is deducted from the return, hence, the gradient is updated to $\nabla_\theta \log \pi(a_t|s_t;\theta)(R_t - b_t(s_t))$. The most commonly used function as a baseline is the state-value function $V^\pi(s_t)$. The state value can be estimated by parameterising $w$ where $w$ is a parameter vector that is learned by some methods such as Monte Carlo. $R_t - b_t$ can be considered as the estimation of the Advantage of

on an action $A(s,a) = Q^{\pi}(s,a) - V^{\pi}(s)$ as $R_t$ is an estimation of $Q^{\pi}(a_t, s_t)$. In general, the Advantage function provides a relative measure of the importance of the action and leads to faster identification of the right actions in policy evaluation. From this approach, the actor-critic architecture is devised, where the policy $\pi$ is the actor and the baseline $b_t$ is the critic Degris *et al.* [2012].

Overall, the policy-based approach is more efficient in high dimensional action space and converges faster than value-based methods as the action space typically is more limited than the possible rewards, especially when considering discrete action spaces. Via gradient methods, the policy updates are smoother and will eventually converge to either local or global optimal.

## 2.2 Deceptive Planing

Masters and Sardina [2017b] applied Bell and Whaley's probabilistic goal recognition theory to path planning scenario. They assume the existence of an observer trying to identify an agent's goal or intention by observing her actions. The observer is modelled as a probabilistic goal recognition system which returns a probability distribution across potential goals $P(G|\vec{o})$ given the observations made so far.

Masters and Sardina [2017b] further simplified the probability distribution $P(G|\vec{o})$ by only referring to the final observation in the sequence of observations and achieved identical probability distribution $P(G|n)$ where n is agent's current location. $P(G|n)$ is calculated by calculating the cost difference between the cheapest plan to the goal from the final observation in the sequence, and the cheapest plan that could have been used to reach the goal from the starting position.

$$costdiff(s,g,n) = optc(n,g) - optc(s,g) \tag{3}$$

where s is the start, g is goal and n is the most recently observed location of the agent whose destination we wish to determine. Consequently, the lower the cost difference between the two paths, the higher the probability for that goal being the real goal.

Masters and Sardina decomposed the concept of deception into step and path. A step is considered truthful if,at that step the probability of the real goal dominates the probability of any other goal,that is $P(g_r|\vec{o}) \geq P(g|\vec{o})$ for all $g \in G \setminus \{g_r\}$. Otherwise the step is considered deceptive. The author also quantifies simulation as the amount by which a false goal dominates the real goal. The greater the dominance, the greater the deception.

$$simulation(O.n) = \max_{g_i \in G \setminus \{g_r\}} P(g_i|O.n) - P(g_r|O.n) \tag{4}$$

Dissimulation is when the probability of the real goal $g_r$ is less than or equal to the probability of the another goal. It is quantified using Shannon's entropy:

$$dissimulation(O.n) = \sum_{g_i \in G} P(g_i|O.n) * \log_2 P(g_i|O.n) \tag{5}$$

In a multi-goal domain, at its start every path is deceptive and truthful at its goal. For a path $\pi$, its first truthful point $FTP_{\pi}$ is defined as the node $\pi^i$, which is itself truthful but all previous nodes $\pi^j$ were deceptive, for all $j \in \{0, ..., i-1\}$ and last deceptive point $LDP_{\pi}$ is a node $\pi^i$, which is itself deceptive but all the upcoming nodes $\pi^j$ are truthful for all $j \in \{i+1, ..., |\pi|\}$. A path is considered **strongly deceptive** if it is deceptive to its last deception point($LDP$) or weakly deceptive if its includes truthful steps before its last deception point($LDP$).

To assess the deceptiveness of a path, it is important to calculate $P(g|n)$ where $g \in G$ for all the steps in the path. This assessment will help in finding out the steps which are truthful if at the particular step the probability of the real goal dominates the probability of any other goal. The relative deceptive strength of any two paths can be ranked on the basis of **deceptive density**. The deceptive density of any path from its start to its goal is defined as the inverse on the number of truthful steps performed by the agent. Thus, the greater the number of truthful steps , the lesser the deceptive density of the path thereby making it easier for the observer to identify the real goal and intent of the agent.

$$density(\pi) = \frac{1}{|N_t|} \tag{6}$$

Deceptive ability of the path can also be explained in terms of LDP. Since the LDP signifies the point in the path at which an observer ceases to be deceived. Therefore, it is really important for the agent to delay the LDP in the path without sacrificing any progress made towards the real goal($g_r$). According to Masters and Sardina, the LDP that maximizes the extent of deception occurs at a node or step which is closest to the real goal's radius of maximum probability. The radius of maximum probability for a particular goal is given as the tipping point where the probability of an agent travelling to any other goal is equal to the probability of travelling to the real goal($g_r$) denoted by $costdiff(s, g_r, n) = costdiff(s, g^{'}$,n). In other words, all the nodes lying within the radius of maximum probability of a particular goal signify higher probability that an agent travelling within the radius will travel to that goal. Since $costdiff(s, g_r, n) = costdiff(s, g^{'}$,n),thus

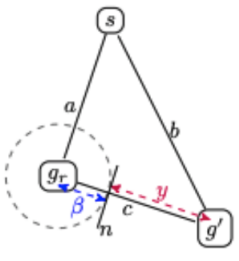



Figure 2: Labels a,b and c stand for optimal cost between nodes

$$\beta = \frac{c + a - b}{2} \tag{7}$$

From the above calculation, it is clear that the LDP which is closest to the radius of maximum probability has the maximum extent of deception.

A small nuance with Masters and Sardina's approach is that it is only applicable to model-based path planning problems and doesn't generalise to MDPs.

## 3 Method

In this section, we present two models for deceptive reinforcement learning. The first is based on the action selection that leads to reaching a region of indifference, in which the probability of reaching any of possible targets is similar. The second model applies a certain level of positive influence to a policy that encourages having less optimal behaviour and satisfies, to a certain extent, a policy that gets closer towards all possible goals. Our main aim is to optimize two conflicting objectives, i.e., efficiently reaching the real goal (optimality) vs. hiding the real intent of the agent as long as possible (deception). Since all the models are judged based on optimality and deception, striking a great balance between the two is paramount.

### 3.1 Indifferent Model

The main idea behind this model is to engineer the problem into a multi sub-goal problem where the sub-goals are deceptive and lie in the region of indifference. To obscure the agent's real intent the problem is broken into multiple sub-goal in such a way that in the process of achieving these sub-goals, the agent behaves rationally and also deceives the adversarial observer by treading in the region of indifference. The region of indifference lies at the overlap of the radius of the maximum probability of all possible goals.

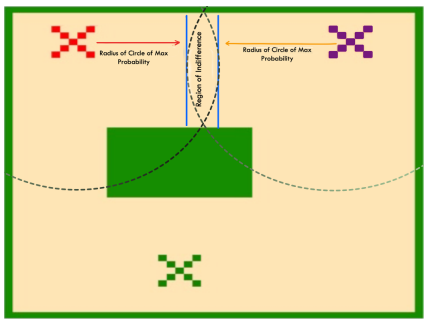



Figure 3: Region of indifference is indicated in blue

As shown in the Figure 3, the region of indifference lies at the overlap of the circle of the maximum probability of two possible goals shown in red and purple. The agent establishes sub-goals in the region of indifference and achieves them before heading towards the real goal, and the point at which the agent achieves all the sub-goals and starts heading towards the real goal is called the last deceptive point [Masters and Sardina, 2017b]. In other words, all the nodes lying within the radius of maximum probability of a particular goal signify higher probability that an agent traveling within the radius will travel to that goal.

The circle of maximum probability is the circle whose radius is the radius of maximum probability. The radius of maximum probability for a particular goal is given as the tipping point where the likelihood of an agent traveling to any other goal is equal to the probability of traveling to the real goal($g_r$).

The agent has different reward functions for each interim goal and the true goal. The agent uses these reward functions to achieve interim goals before moving towards the real goal. The first interim goal is constructed in the region of indifference, which exists at the overlap of the circle of maximum probability of all possible goals. For subsequent interim goals, any goal which doesn't exist in the direction of the real goal is pruned. All the interim goals are established during the exploration phase on the basis of the fact that the overlap of the circle of maximum probability will occur at the mid point of all possible goals. The goals selected for mid point calculation must be rational with respect to the real goal otherwise all such goals are pruned.

$$Interim(s) = \sum_{g_f \in G \setminus \{g_r\}} [V_{g_f}(s) - V_{g_r}(s)] \tag{8}$$

$$InterimGoal = argmin_s Interim(s) \tag{9}$$

The identification of the interim goals can also be achieved with the help of multiple reward functions trained on all possible goals. As shown in Equation 9, any state that minimises the value of the Interim function will make the observer indifferent because from that state the value of reaching either the real goal or fake goal will be identical. Generally, the states lying in the region of indifference will force the interim function to zero, but because all those states might not be reachable from the current state, therefore, it becomes important to select a state which is reachable for the current map configuration and minimizes the Interim function.

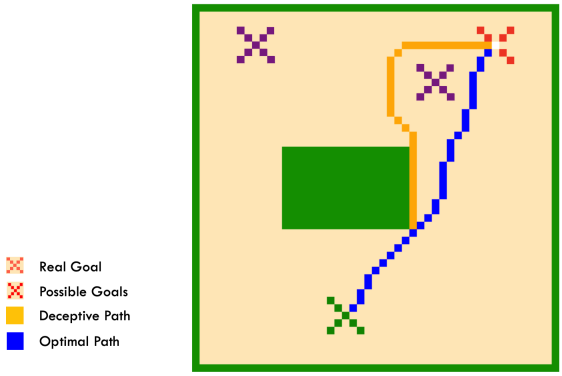



Figure 4: Path traced by the agent using Indifference Model

In Figure 4, the agent constructs the first interim goal with the help of the circle of maximum probability of all possible goals shown in purple and red. The agent uses the specific reward function to reach the first interim goal and after reaching the first interim goal depending upon the position of other goals with respect to the first interim goal, the agent might choose to create another interim goal by only considering goals which exists in the direction of the real goal and pruning away other goals or directly tread towards the real goal.

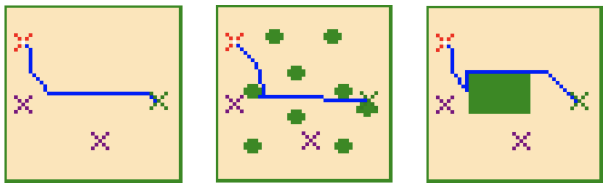


Figure 5: Indifference Model on different terrains (green blocks are obstacles)

The agent using the indifference model may create multiple interim goals as the stepping stone before reaching the final goal. The relevance of the interim goal is judged based on its optimality with respect to the real goal. Different versions of the indifference models may prioritize one or both to achieve a varying degree of deception. Models prioritizing only the deception parameter will select all the interim goals that add deception irrespective of its impact on rationality, but models prioritizing optimality or trying to strike a great balance will prune all the interim goals that add irrationality irrespective of its impact on deception. The impact of different versions of indifference model on optimality and deception is discussed and compared elaborately in Section 4.1.

### 3.2 Deceptive Policy Model

Reference to our main aim to devise a general model that learns a deceptive policy without the need for hard-crafted logic. In this method, we focus on policy-based reinforcement learning and having a generalised objective to avoid the need for hand-crafted logic. Policy-based approaches are versatile comparing to Value-based methods. They would work in different situations, have a better convergence property, more effective in high-dimensional action spaces and also can generate stochastic policies. We also look at a different policy based algorithms to determine how to best design and optimise such a model.

The main idea of this approach is to train a policy-based model to take into consideration all possible targets, real and fake. This is done by introducing an extra deceptive component in the value function update at the policy evaluation and improvement steps. The deceptive component is bound by $\lambda$ which acts as a balancing factor between *optimality* and *deception*. Figure 6 shows how this approach can converge to a deceptive adaptive policy using the same start and possible target positions with different map terrains, which makes it a more generalised approach.

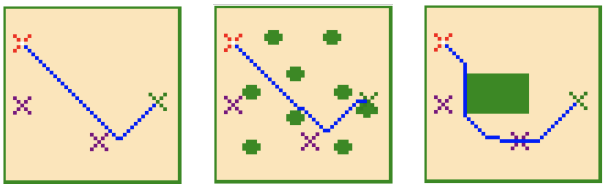


Figure 6: Deceptive Policy Model on different terrains ($\lambda$=1, $w$=2), (green blocks are obstacles)

As in any policy-based algorithm, we need to define an objective function to assess and improve the effectiveness of the policy. Standard policy approaches, such as advantage-actor-critic [Mnih *et al.*, 2016], exploit the expected value function that measures the potential future rewards that could be taken from action a given state s:

$$V^{\pi}(s) \leftarrow \sum_{s' \in S} \mathcal{P}(s'|s, \pi(s))[r + \gamma V^{\pi}(s')] \quad (10)$$

Such that $\mathcal{P}(s'|s, \pi(s))$ denotes the transition probability of reaching state s' when being state s and following policy $\pi(s)$ in a non-deterministic environment. $r(a|s)$ is the reward received by taking action $a$ at state $s$.

In the deceptive policy model, we added a deceptive component $\mathcal{D}(s)$ weighted by $\lambda$ to the value function as follows:

$$V^{\pi}(s) = \sum_{s' \in S} \mathcal{P}(s'|s, \pi(s))[\nabla + \gamma V^{\pi}(s') + \underbrace{\lambda \mathcal{D}(s)}_{\text{Deceptive}}] \quad (11)$$

Such that:

$$\mathcal{D}(s) = \mathit{Penalty}(s, g_r) - w \sum_{g_f \in G \setminus \{g_r\}} \mathit{Penalty}(s, g_f) \quad (12)$$

and:

$$\mathit{Penalty}(s, g_r) = |V(g_r) - V(s)| \quad (13)$$

$$\mathit{Penalty}(s, g_f) = |V(g_f) - V(s)| \quad (14)$$

The deceptive component $\mathcal{D}(s)$ considers the penalty to reach the real goal $g_r$ from state s, reduced by the sum of the penalties to reach each other possible fake goals weighted by an augmentation constant $w$ (Equation 12. The *Penatly* function is nothing but the difference in the value at state s and the value at the possible goal state; either the real goal $g_r$ or each of the possible goals $g_f \in G \setminus \{g_r\}$ as per Equations 13 and 14 respectively. As $V(s)$ represents the expected total reward for an agent starting from state s to the real goal. And $V(g_f)$ represents the expected total reward for an agent starting from the fake goal $g_f$ to the real goal; the difference between these values will represent the cost required to reach the fake goal $g_f$ from state s. If state s is closer to the real goal than $g_f$ then this value is used

as a penalization factor which the policy must minimize. For implementations in the path planning scenario, this penalty mimics the cost (which can use heuristics such as euclidean distance) to reach the possible goal state from current state s.

$\lambda$ and $w$ are hyper-parameters; the higher the value of $\lambda$, the more influence the deceptive component gets and hence the agent deviates away from the optimal path; and $w$ augments the negative influence of the penalty to reach the fake goals as compared to the real goal. Increasing the value of $w$, will add more weight for the agent to get closer to the fake goals as well. However, reducing it will have an impact on time the policy will take to converge.

In general, this approach adds a penalisation factor to the value function of the policy such that a policy that leads to a direct optimal path to the real goal without considering other goals will have a lower potential accumulated future rewards. Therefore, influence the policy to learn a sub-optimal way to the real goal in the direction of other goals as well, which makes it still rational but more deceptive.

Algorithm 1 shows how the value function changes based on (Equation 11) can be incorporated in the policy iteration algorithm during policy evaluation and improvement phases. Similarly, this formulation can be incorporated into other policy gradient-based algorithms. For instance, in Actor-Critic Algorithm that uses Temporal Difference (TD), the critic determines if there are any improvements by calculating the new state value function. Then the evaluation is the TD error:

$$\delta_s = r + \gamma V\left(s'\right) - V\left(s\right) \tag{15}$$

So $\delta_s$ is used then to adjust the parameters reduce this error.

$$\delta_s = r + \gamma(V\left(s'\right) + \boldsymbol{\lambda \mathcal{D}(s')}) - (V\left(s\right) + \boldsymbol{\lambda \mathcal{D}(s)}) \tag{16}$$

**Hyper-parameters optimization** Figure 7 shows how changing the value of $\lambda$, with fixed w, impacts the level of deception of the agent. The start position is marked with green X on the rightmost side of the map; the real goal is marked with red X located on the top left hand and the other three possible fake goals marked with purple X distributed across the map.

Even with a small value of $\lambda$ (See Figure 7a), the agent behaves in a deceptive but still optimal, cost-effective, and rational way. With increasing the value of $\lambda$, the agent direction gets influenced more with the two closer targets in the middle top and bottom sides of the map. This policy variation is quite visible in Figure 7e; the policy was initially drawn towards the top goal that is closer to the start position, but then the policy changes direction as it gets closer to the two other goals on the left and bottom side before finally heading towards the real goal. However, increasing the value of $\lambda$ beyond a certain extent would lead to an agent with some irrational behaviour (See Figure 7f. This is because the

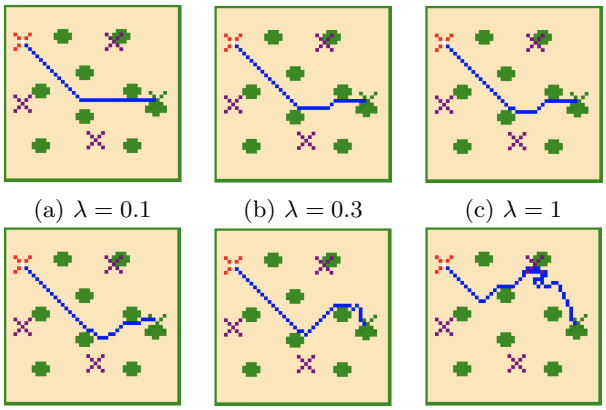
**Algorithm 1** Policy Based Deception

Initialise $V(s) \in R$, $\pi(s) \in \mathcal{A}(s)$ arbitrarily for all $s \in \mathcal{S}$

**function** EVALUATE($\pi, V$) ▷ Policy Evaluation
  Initialise $V(s) = 0$, for all $s \in \mathcal{S}$
  **repeat**
    **for each** $s \in \mathcal{S}$ **do**
      $v \leftarrow V(s)$
      $V^{\pi}(s) \leftarrow \sum_{s' \in S} \mathcal{P}(s'|s, \pi(s))[r + \gamma V^{\pi}(s') + \boldsymbol{\lambda \mathcal{D}(s)}]$
      $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
  **until** $\Delta < \theta$ (a small positive number)

**function** IMPROVE($\pi, V$) ▷ Policy Improvement
  policy-stable $\leftarrow$ *true*
  **for each** $s \in \mathcal{S}$ **do**
    $a \leftarrow \pi(s)$
    $\pi(s) \leftarrow \arg\max_a \sum_{s' \in S} \mathcal{P}(s'|s, \pi(s))[r + \gamma V^{\pi}(s') + \boldsymbol{\lambda \mathcal{D}(s)}]$
    **if** $a \neq \pi(s)$ **then**
      policy-stable $\leftarrow$ *false*
  **if** policy-stable = *true* **then**
    **return** V; Stop
  **else**
    EVALUATE($\pi, V$)

agent would be encouraged to find a policy that is more deceptive and in line with proximity to all possible goals, before reaching the real goal. It is interesting to see from the Figure that the policy can mimic the behaviour of pruning possible goals that are far without explicitly incorporating the logic. This point is worth exploring more in future work to analyze to which level the model is capable of generalising.



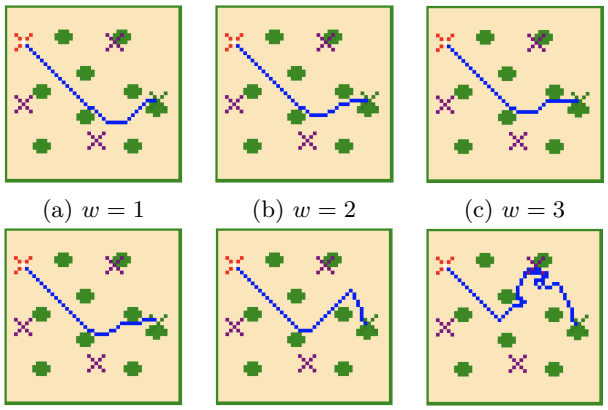
(a) $\lambda = 0.1$ (b) $\lambda = 0.3$ (c) $\lambda = 1$

(d) $\lambda = 1.5$ (e) $\lambda = 2$ (f) $\lambda = 3$

Figure 7: Varying $\lambda$ on policy model ($w = 3$)

Changing the value of $w$ have a relatively smoother impact on the generated policy, as seen in Figure 8 with fixed $\lambda = 1$. However, similar behaviour of irrationality appears with higher weights proportional to the number

of fake goals. We also notice that lowering $w$ could lead to a longer time for the policy to converge.



(a) $w = 1$ (b) $w = 2$ (c) $w = 3$

(d) $w = 4$ (e) $w = 6$ (f) $w = 10$

Figure 8: Varying $w$ on policy model ($\lambda = 1$)

Based on our admittedly somewhat limited evaluation due to the computational and time limitations, we found that having ($\lambda = 1$ and $w = \textit{Number of fake goals}$) strikes a reasonable balance of being deceptive while rational. Hence, we selected these values for the Deceptive Policy Method evaluation and comparison with other models in Section 4.1.

## 4 Experimental Evaluation

In this section, we present an experimental evaluation of our two models from Section 3 using a modified version of Masters and Sardina's framework for path planning (P4 simulator) and via human evaluation. The experiment aims to quantify the level of **deception** compared to the truthful baseline model and also to measure the **optimality** of the deceptive path by considering the proportion by which the cost of the deceptive path deviates from the true optimal path. We also included the output of two models from Yang *et al.* and Masters and Sardina for comparison against previous work done on the same problem.

### 4.1 Computational Evaluation

**Experiment Design**

We implemented the deceptive policy and indifference models defined in Section 3 using the P4 simulator framework for evaluating their deceptive behaviour[3].

**Agents Evaluated** We evaluated six different agents in our experiments:

1. an 'honest' agent, which uses value-based reinforcement learning to find the optimal policy for the real reward function to the target goal,
2. a deceptive agent using deceptive policy model, with $\lambda = 1$ and $w = 2$,

3-4. two deceptive agents using different versions of indifference model, exhibiting trade-off between optimality and deception,

5. a deceptive agent based on the 'Ambiguity' model as presented in [Yang *et al.*, 2020],
6. a deceptive agent based on the path planning model '$\pi d4$' as presented in [Masters and Sardina, 2017b]. This model was chosen because it was it identified by Masters and Sardina to be strongly deceptive to the full extent with least cost out to the other models proposed in the paper.

**Measurement Metric** For ranking the models based on deception and optimality, we implemented two metrics:

1. The total cost of the path traversed by the agent using a particular model.
2. The probability assigned to each possible goal being the true target. This is derived using a naïve intention recognition algorithm for path planning [Masters and Sardina, 2017a], which uses the concept of the cost difference between the agent's path using a particular model and agent's honest path.

**Reward Configuration** Rewards are configured in such a way that the P4 simulator awards 10,000 for reaching the real goal. The living rewards are configured as $-1$ for actions up, down, left or right and $-\sqrt{2}$ for any diagonal move.

**Map Configurations** The entire experimental setup uses three different map structures: (1) $49 \times 49$ map with a large obstacle concentrated at the center as shown in Figure 15(a); (2) $49 \times 49$ map with random scattered and high-density obstacles as shown in Figure 15(b); (3) $49 \times 49$ map with no obstacles, such as in Figure 15(c) For each map structure, we configured eleven different variations by changing the coordinates of starting point, real and possible goals leading to 33 different map configurations, as shown in Figure-16. All 33 map configurations are applied to all models and intention recognition measures are calculated at every 10% along the agent's path.

**Results**

Figure 9 and 10 show an overview of the results from the experiments conducted. Figure 9 shows the aggregated average of the intention recognition probability of the true goal calculated at intervals of 10% of the total path traversed (Density). As mentioned in 4.1, each map contains three possible destinations (one real and two fake). The overall trend showed that the accuracy of the intention recognition algorithm in recognizing the true goal was positively proportional to the increase in density. However, with deceptive models, the intention recognition algorithm failed to clearly identify the agent's real goal, especially for densities between 20% to 80%.

By comparing the results in Figures 9 and 10, there seem to be, as expected, a clear trade-off between deception and optimality (cost-efficiency). Overall both Masters and Sardina's $\pi$d4 and our Deceptive Policy model

[3]https://github.com/morningcloud/Deceptive-Reinforcement-Learning

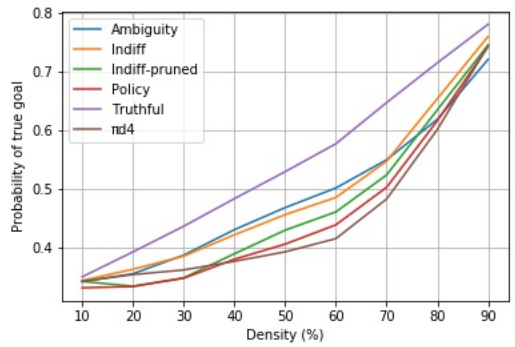



Figure 9: Goal Recognition

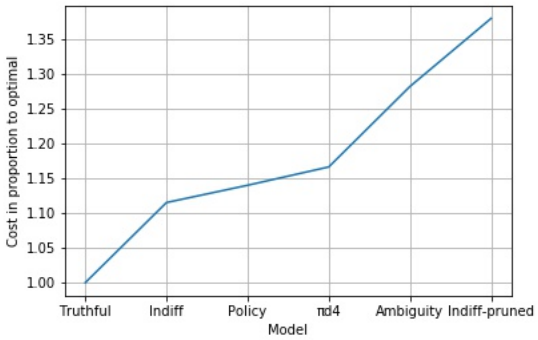



Figure 10: Cost in proportion to true optimal

shows relatively close results comparing both the effectiveness in deception and the cost efficiency; Masters and Sardina's $\pi$d4 model is the most deceptive for densities after 35%, while two of our models "Indifferent with pruning" and "Deceptive Policy" are more deceptive before that.

The Deceptive Policy model, however, continues to be close to the $\pi$d4 model with regards to deception, but slightly more cost efficient. Considering both measures, we can say that the deceptive policy model is on a par with the known published work in this field. It is worth noting that $\pi$d4 uses the probability heat-map from [Masters and Sardina, 2017a], which is also used by the naïve intention recognition algorithm used for evaluation. This indicates that the results of $\pi$d4 model is the theoretical lower bound any deceptive rational algorithm could reach while being assessed using this particular intention recognition algorithm.

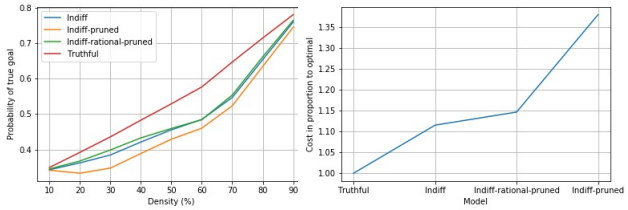
Comparing to the other reinforcement learning-based models used in this experiment, the deceptive policy model scores the best on the scale of deception and have a relatively efficient cost proportion. The possible reason behind this advantage is that other models use and alternate between multiple reward functions; In case of Indifferent models, these reward functions refer to the interim sub-goals that were pre-calculated, which interns mean that it is optimized to that specific target and not the overall problem. In case of Yang *et al.*'s Ambiguity model, each of the reward functions refers to the optimal path to reach one of the possible targets separately and then the Q function to be used during action selection is chosen based on the entropy of the current observed state. Hence, these multiple reward functions are still bounded to optimize actions for different objectives. However, the deceptive policy-based model learns an inherently deceptive policy using one objective function. This means that once the policy converges, it is guaranteed to be optimal for the enforced constraints; at every step in the path, the action is selected by following one policy that takes into consideration the optimal expected future rewards that satisfy both deception and goal completion.

We also believe that this algorithm can show additional improvement by implementing deep neural network-based approximation methods to calculate the policy and use the gradient of the log of the policy and the advantage of taking action to update the weights of the network. We have attempted to implement such an approach using advantage actor-critic (A3C). However, we faced challenges to stabilising the model to give consistent learning results on every run; we believe this would be from the way we designed the environment simulation in this algorithm; we have set an upper bound on the maximum number of steps an agent can take per episode after which the simulation would terminate with a negative reward. In this way, if the model does not find the real goal in the first set of episodes, it will not be able to tune the policy in the right trajectory and hence fail to converge towards the target. Given more time we believe by changing the limit of the step-per-episode constraint and fine-tuning, the neural network-based methods would generalize more and lead to better performance.

Figure 11a presents an overview of the results from experiments performed on different versions of the indifference model. Different versions of the model prioritize either deception or optimality and establish a notion of a trade-off between the two parameters. On the scale of deception, the agent using the model *Indiff-pruned* outperforms other versions of the indifference model because it only prioritizes deception by creating interim goals which may not be rational and thereby hurting optimality. Figure 11b strengthens the argument that there exists an inverse relation between deception and optimality by clearly showing that the agent's path with highest deception has highest cost(or lowest optimality). Agent using the model *Indiff-rational-pruned* prioritizes optimality over deception by neglecting all interim goals which aren't optimal. *Indiff-rational-pruned* model performs better on the scale of optimality than deception. *Indiff* model doesn't prune interim goal and shows almost identical performance to *Indiff-rational-pruned* on the scale of deception and optimality, but *Indiff-rational-pruned* model will outperform the *Indiff* model in striking a fine balance between deception and optimality in

case of multiple fake goals positioned in different directions with respect to the real goal.

In Figure 9 the Indifferent models appear next to the Deceptive Policy model on the scale of deception and optimality. The *Indiff-pruned* version of Indifference model start on par with the policy model for the first 35% of path completion and ends the journey being less deceptive. The *Indiff* version of the indifferent model shows almost identical performance as the ambiguity model on the scale of deception and optimality but outperforms the ambiguity model in its simplicity of solving the problem of hiding the agent's real intention. The ambiguity model generates sequence of actions that have positive reward for several reward functions, including the true reward function by checking the divergence from optimal behavior which requires training multiple reward functions and calculations for obtaining divergence whereas the simplest version of indifference model(*Indiff*) only trains multiple reward functions on the real goal and interim goals to achieve identical performance.




(a) Goal Recognition (b) Proportional cost

Figure 11: Indifferent Models

### Limitations

There are some limitations surrounding our evaluation and model implementation. Even though the Deception Policy model shows relatively good results in this experiment. As highlighted in Section 3.2, the model is subject to hyper-parameter tuning. Hence, the performance would be relative to the appropriate selection of the right hyper-parameter values, which may differ based on the map size and configuration.

## 4.2 Human Evaluation

We conducted a human behavioral experiment to judge the ability of models described in Section-3 to deceive human subjects as an additional verification to the intention recognition algorithm presented in [Masters and Sardina, 2017b].

### Experiment Design

The experiment design implemented for human subject evaluation was similar to the one used for evaluations using the intention recognition algorithm in section 4.1. The three-striking differences between the experimental designs were: (1) The intention recognition algorithm used as a measurement metric was replaced by human subjects for estimating the goal distribution; (2) the human subjects were only presented with a single map configuration as shown in Figure-12; and (3) the output shown included only two of the models; the version of Indifferent model without pruning and the Deceptive Policy model.

Human subjects in this evaluation were not naïve because they were explicitly informed before the experiment that the agents have the tendency to deceive and hide their real intentions, unlike previous evaluations using naïve intention recognition algorithm, which assumed the agent under observation has no tendency to deceive or hide its real intention.

The human evaluation includes 30 university student participants, aged between 20-35 years and from diverse gender identity and cultural backgrounds. All the participants are post-graduate students but are studying different degrees, including IT, Engineering, Medical, Arts, and Business.

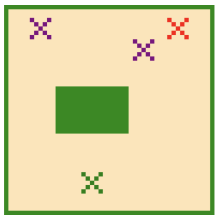


Figure 12: Map configuration for human evaluation

### Results

Figure 13 shows the average probability prediction of the real goal reported by participants at five checkpoints starting from 10%, and then every 20% interval the total path traversed until 90%.

Interestingly, the human observers were not decisive on any model, including the Truthful path till past the 50% checkpoint. The Indifferent model appears to be more deceiving to the human observers than the Deceptive Policy model for most of the checkpoints except the 70% checkpoint. After concluding the experiment, the majority of the participants also reported that the Indifferent model seemed more deceptive to them than the Deceptive Policy-based one. This feedback is understandable considering the generated path from each of the models on this particular map (See Appendix 6 for the images of both generated maps); the indifferent model included more natural-looking curves. At 50% of the traversed path, it deviates away from the real goal, which looks more deceptive to the human. However, for the Deceptive Policy model, the path was sharper and seemed to be heading towards the fake goal that is closer to the target goal. This led most participants to report that the agent is either heading towards the fake goal closer to the target or the real target itself with equal probability. At 70% checkpoint, the Deceptive Policy agent becomes very close to the fake goal that is near the target, which led to a high increase in the deception rate.

It is notable from Figure 14a and 14b that the hu-

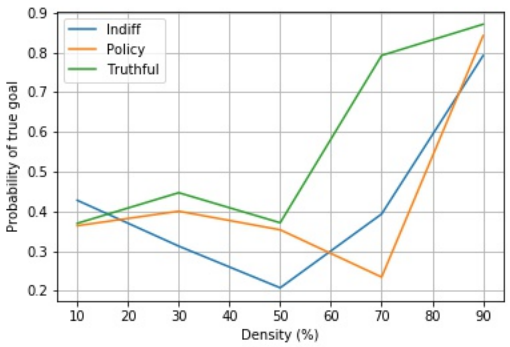
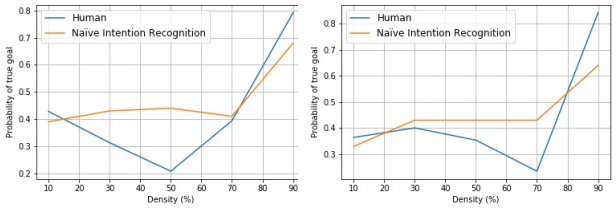



Figure 13: Human Evaluation

man participants were deceived more than the intention recognition algorithm till the 70% checkpoint. Interestingly, the human evaluation gave inverse results compared to the intention recognition algorithm on the Indifferent Model Relatively; the human results show that they are increasingly deceived up to the 50% checkpoint, while the computational evaluation algorithm was showing the opposite. However, with the Deceptive Policy method, both results, although different, were heading the same direction.

Overall, we can see that both models did deceive human observers to a reasonable extent.




(a) Indifferent Model (b) Policy Model

Figure 14: Human VS Naïve Intention Recognition Evaluation

### Limitations

Several limitations surrounding our human subject evaluation are mentioned below:

1. Although participants are from different degrees and backgrounds, we acknowledge that this may not reflect a completely unbiased view. This is due to the limited number of students, and some of them have prior knowledge about reinforcement learning and autonomous path planning algorithms. This background knowledge may have influenced their judgment.
2. Entire evaluation is focused on the path planning domain, which is considered a perfect fit for human behavioral experiments due to human prowess in intention recognition supported by strong spatial reasoning leaving other domains with a plethora of assumptions, e.g., partially observable, indeterministic and multi reward environments unexplored.
3. The naïve intention recognition metric only uses the notion of the cost difference between an agent's deceptive and honest path, which is not sufficient to encapsulate and measure the extent of the deception. The intention recognition metric being a weak indicator of deception does not provide a strong base to compare the human subject experiments.
4. The human subjects were only presented with one map configuration, thereby limiting the scope for generalization.

## 5 Conclusions and Future Work

In this paper, we have presented two novel[4] deceptive methods using model-free reinforcement learning. The first is the Indifferent model that generates interim subgoals at the overlap of the circle of maximum probability of possible goals, and the second is a Deceptive Policy model that adds a deception component to the value-function updates in policy evaluation and improvement steps. We also implemented these models on a deceptive path-planning problem as a proof of concept. We have shown that it is possible to devise a single deceptive policy based on reinforcement-learning using a single objective and reward function without the need to train multiple non-deceptive reward functions separately and add further logic during the action selection stage. Finally, we showed that reinforcement-learning based methods are capable of achieving a performance that is close to model-based path planning approaches (e.g., [Masters and Sardina, 2017b]), and with further refinements, it could show more competitive advantages and have better generalisation properties.

There is scope to develop more optimised policy-based methods by using Multi-objective reinforcement learning (MORL) instead of scaling down the two conflicting objectives of deception and goal completion into a single linear objective function. This can be applied in future work. We still believe that these models can be extensively generalized by implementing deep neural network-based approximation methods to help identify the best policy for hiding the agent's real intent in case of large state space. We are currently trying to implement more sophisticated gradient-based policy optimization techniques such as the advantage actor-critic to ensure faster convergence in large state spaces.

## Acknowledgments

We thank our supervisor Tim Miller for his support and continuous guidance and advice.

[4]To the best of our knowledge

## References


[Bell, 2003] J. Bowyer Bell. Toward a theory of deception. *International Journal of Intelligence and Counterintelligence*, 16(2):244–279, 2003.

[Degris *et al.*, 2012] Thomas Degris, Patrick M Pilarski, and Richard S Sutton. Model-free reinforcement learning with continuous action in practice. In *2012 American Control Conference (ACC)*, pages 2177–2182. IEEE, 2012.

[Haarnoja *et al.*, 2018] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *International Conference on Machine Learning (ICML)*, 2018.

[Keren *et al.*, 2016] Sarah Keren, Avigdor Gal, and Erez Karpas. Privacy preserving plans in partially observable environments. In *Proceedings of IJCAI'16*, pages 3170–3176, 2016.

[Koenig, 1996] S. Simmons Koenig. The effect of representation and knowledge on goal-directed exploration with reinforcement-learning algorithms. *R.G. Machine Learning*, 1996.

[Kulkarni *et al.*, 2018] Anagha Kulkarni, Matthew Klenk, Shantanu Rane, and Hamed Soroush. Resource bounded secure goal obfuscation. In *AAAI Fall Symposium on Integrating Planning, Diagnosis and Causal Reasoning*, 2018.

[Masters and Sardina, 2017a] Peta Masters and Sebastian Sardina. Cost-based goal recognition for path-planning. In *AAMAS*, pages 750–758. IFAAMAS, 2017.

[Masters and Sardina, 2017b] Peta Masters and Sebastian Sardina. Deceptive path-planning. In *Proceedings of IJCAI'17*, pages 4368–4375, 2017.

[Masters and Sardina, 2019] Peta Masters and Sebastian Sardina. Goal recognition for rational and irrational agents. In *Proceedings of AAMAS'19*, pages 440–448. IFAAMAS, 2019.

[Mnih *et al.*, 2015] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.

[Mnih *et al.*, 2016] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937, 2016.

[Sutton and Barto, 2018] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[Wiewiora, 2003] E. Wiewiora. Potential-based shaping and q-value initialization are equivalent. *Journal of Artificial Intelligence Research*, 19:205–208, Sep 2003.

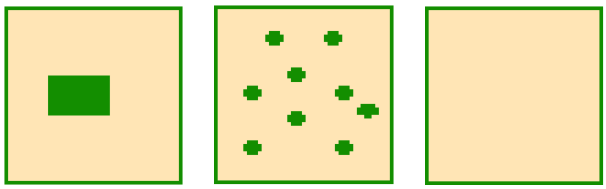
[Williams, 1992] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

[Yang *et al.*, 2020] Yue Yang, Zhengshang Liu, and Tim Miller. Deceptive reinforcement learning for preserving the privacy of reward functions. *Paper ID: 4313*, 2020.

# 6 Appendix

## 6.1 Computational Experiment Environments

### Map Structures



(a) Concentrated (b) Scattered (c) No obstacles

Figure 15: Different Map Structures

### Start and Goals Coordinates

*Note:* Below coordinate combinations were used in all 3 map structures while running the experiments.

Green X denotes starting position of the agent, Red X denotes the real goal and Purple X denotes possible fake goals.

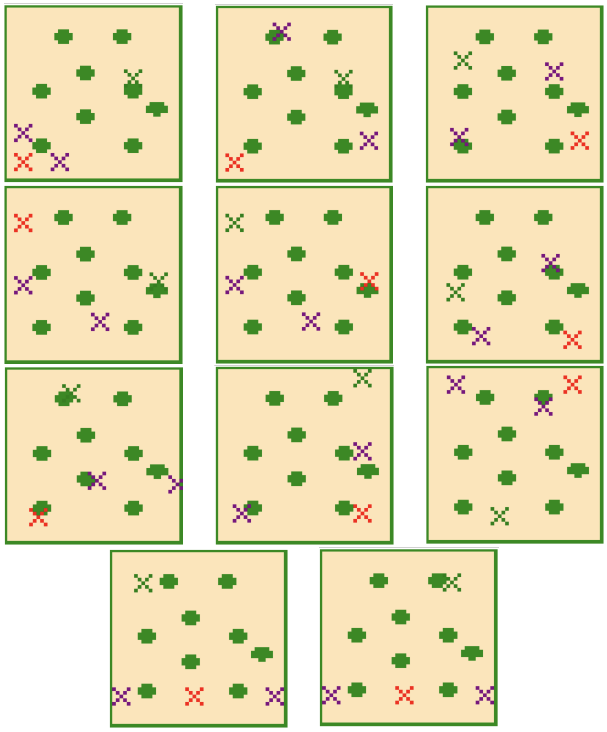
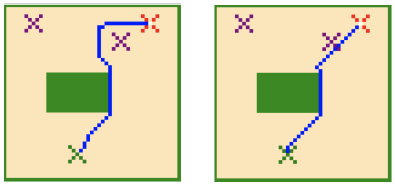


Figure 16: Map Sets

## 6.2 Human Experiment Maps



(a) Indifferent (b) Policy

Figure 17: Human Map Structures