ECAI 2024
U. Endriss et al. (Eds.)

doi:10.3233/FAIA240734

Preserving the Privacy of Reward Functions in MDPs Through Deception

Shashank Reddy Chirra [a,*,1], **Pradeep Varakantham**[a] **and Praveen Paruchuri**[b]

[a]Singapore Management University
[b]International Institute of Information Technology, Hyderabad

Abstract. Preserving the privacy of preferences (or rewards) of a sequential decision-making agent when decisions are observable is crucial in many physical and cybersecurity domains. For instance, in wildlife monitoring, forest rangers must conduct surveillance without revealing animal locations to poachers. This paper addresses privacy preservation in planning over a sequence of actions in MDPs, where the reward function represents the preference structure to be protected. Observers can use Inverse RL (IRL) to learn these preferences, making this a challenging task. Current research on Differential Privacy (DP) in this setting fails to ensure a lower bound on the minimum expected reward and offers theoretical guarantees that are inadequate against IRL-based observers. To bridge this gap, we propose a novel approach rooted in the theory of deception. Deception includes two models: dissimulation (hiding the truth) and simulation (showing the wrong). As our first contribution, we theoretically demonstrate a significant privacy leak in the current dissimulation-based method. Our second contribution is a novel RL-based planning algorithm that uses simulation to effectively address these privacy concerns while ensuring a guarantee on the expected reward. Through experimentation on multiple benchmark problems, we show that our proposed approach outperforms existing methods in preserving the privacy of reward functions. Code to reproduce the results can be found at: https://github.com/shshnkreddy/DeceptiveRL

1 Introduction

In the realm of decision-making, particularly in situations involving resource allocation in the context of security, agents face the complex task of making choices that are potentially observable by external entities. These choices can carry substantial implications, revealing critical insights into the preferences (or significance) over different targets (or states in general); which can be strategically harnessed by observers in potentially harmful ways. The central challenge lies in optimizing these decisions while safeguarding the privacy of the agents' underlying preferences. Our specific focus is on addressing this challenge within the context of Reinforcement Learning (RL) based planners where the reward function represents the preferences that must be kept private. In such a case, it is crucial to recognize that a significant portion of the reward is embedded in the agent's decision-making policy. Therefore the agent must take actions that preserve the privacy of the reward while still achieving good performance. Take, for instance, green security games (GSGs) [7], where forest rangers patrol to monitor various animal populations. Poachers observing these patrols could exploit the information to locate and target animals. Thus, rangers must conduct effective surveillance while simultaneously deceiving the poachers. Similarly, in urban policing, cities are divided into regions, with each region assigned a reward based on factors such as crime rates, wealth, etc [4, 8]. It is important for law enforcement to keep this reward function private for enhanced security.

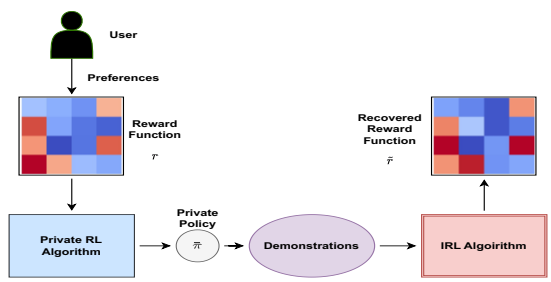

Figure 1. Flow of Information

The potential of reverse-engineering the agent's reward forms the basis for the field of Inverse Reinforcement Learning (IRL) [19] which poses a substantial privacy risk. IRL has demonstrated the remarkable ability to reconstruct high-quality reward functions across various environments [13, 9]. This concern underscores the importance of developing robust mechanisms to shield the agent's reward function, ensuring the integrity of decision-making and preventing potential privacy breaches. The problem of privacy preservation of the reward function is illustrated in Figure 1. First, the user defines the reward function r, which encodes their preferences. Next, a private RL algorithm learns a policy that maximizes the reward while simultaneously keeping the reward function private. An observer can then use an IRL algorithm to recover a reward function $\tilde{r}$ by observing demonstrations of the agent. If $\tilde{r}$ is of a high quality, it will have properties very similar to r making it feasible for an observer to estimate the preferences of the user. An ill-intentioned observer can use this information about the reward function to manipulate the agent to engage in undesired behaviours [10]. IRL covers the entirety of methods for recovering the reward function within our specific context, wherein the observer lacks additional information such as the nature of the reward function, domain knowledge, etc. However, a notable limitation of IRL lies in its assumption that demonstrations are not deceptive. To address this limitation, we introduce two modifications to IRL algorithms that

* Corresponding Author. Email: shashankc@smu.edu.sg.
[1] Work partially conducted when enrolled at the International Institute of Information Technology, Bangalore.

account for deceptive demonstrations.

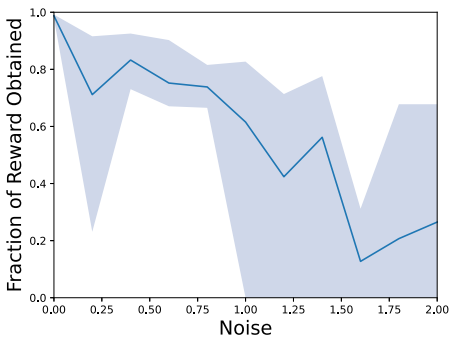

Figure 2. Expected Reward v/s Injected Noise of DQFN in the Four Rooms environment averaged over 5 seeds. The shaded region represents the max and min values. The large variance in the reward obtained underscores the difficulty in managing the privacy-reward tradeoff when using DQFN.

Existing Work tackles the problem in two ways:
(1) *Through the use of Differential Privacy (DP) methods*: DP-based methods [22, 28, 30] and the Deep Q-learning with Functional Noise (DQFN) algorithm [29] introduce noise to computations such as Q-functions, Value functions, and Policy Gradients. This addition of noise guarantees that reward functions within l_∞-neighbourhood of each other return the same policy, making it difficult for an observer to reconstruct the exact reward function. In the context of reward reconstruction, these guarantees are ill-suited as (a) there are infinitely many reward functions that can explain the observed behaviour, (b) the l_∞ and l_p norms are not good metrics to use when comparing reward functions as two reward functions in the same l_∞ neighbourhood may possess several other properties that pose a privacy leak such as ordering of polices (explained in Section 3). This leads to a privacy leak in practice as highlighted in [22]. This is in addition to the fact that these approaches lack built-in reward constraints make it difficult for a user to balance the tradeoff between expected reward and privacy without resorting to time-consuming hyper parameter searches. As shown in Figure 2, it is difficult to achieve a good privacy/reward trade-off due to high variance in expected reward as noise is increased. As highlighted in [22], another drawback of these methods is that the quality of the reward function recovered by an observer is independent of the noise added, undermining their effectiveness as a private algorithm.
(2) *Through the use of deception*: Deception involves the act of intentionally creating or upholding false beliefs in the minds of others [3]. There are two primary approaches to deception: (a) "dissimulation" that relates to "hiding" the truth, and (b) "simulation" which entails providing false information to "mislead" the observer into believing something that is not true. The Max Entropy Intentional Randomization (MEIR) [21] algorithm developed to preserve the privacy of the reward function is an existing "dissimulation" based deception algorithm (as shown later in this paper). Although the MEIR algorithm satisfies constraints on the expected rewards, we show that it leaks significant information about the reward function when faced with IRL-based observers. Other deception-based planning algorithms such as those discussed in [14, 15, 18, 20, 17, 16], are limited in their ability to tackle this problem. This limitation arises from their focus on optimizing deception for a singular trajectory, inadvertently disclosing information about the reward function across multiple trajectories. These methods also consider a different problem as discussed in the Appendix [6].
Contributions: Our contributions are as follows:

- **Theoretical Analysis of Privacy Leak for MEIR**: We demonstrate theoretically and intuitively the significant privacy vulnerabilities of the MEIR algorithm when faced with an IRL observer.
- **Novel Max Misinformation Algorithm**: We introduce the innovative Max Misinformation (MM) Algorithm, designed to address the shortcomings of MEIR and DP-based methods. A key element is the introduction of an anti-reward function, enabling a balanced tradeoff between the expected value and the ability to deceive the observer.
- **Effectiveness Against IRL algorithms**: We provide insights into why MM can robustly counteract observers utilizing various IRL algorithms, demonstrating its superiority in preserving privacy of the reward function. In addition, we experiment against two additional algorithms based on IRL that an observer might use if they know they are being deceived, and demonstrate the robustness of the proposed algorithm in this case as well.
- **Comprehensive Evaluation**: To gauge the effectiveness of our algorithm, we rigorously evaluate it against IRL-based observers across diverse benchmark environments. We measure the quality of the recovered reward functions in comparison to the original reward using the Rollout method [11], Pearson Correlation, and the Equivalent-Policy Invariant Comparison (EPIC) distance [11]. Our conclusive findings highlight that the MM algorithm outperforms existing deception-based and DP-based algorithms, firmly establishing its efficacy in maintaining the privacy of the reward.

2 Background

We provide a brief overview of the relevant decision-making models (Markov Decision Process (MDP) and Deceptive RL), Max Casual Entropy Inverse Reinforcement Learning (MCE-IRL) and Max Entropy Reinforcement Learning (MERL) which is the backbone for MCE-IRL based algorithms as well as pre-existing Deceptive RL formulations.

Markov Decision Process We consider environments that can be expressed as Markov Decision Processes (MDP). An MDP M is defined by the tuple $(S, A, P, r, \gamma, \mu)$, where S is the set of states, A is the set of actions, $P(s'|s, a) \in [0, 1]$ is the transition probability, $r(s, a) \in \mathbb{R}$ is the reward function, $\gamma \in [0, 1]$ is the discount factor and μ is the initial state distribution. A policy $\pi(.|s)$ is a probability distribution over the set of valid actions for a given state. In this paper, we base our results on the assumption that both S and A are discrete, with every state reachable under μ and P. The cumulative γ-discounted value, or expected value, of the reward obtained by following π in M is denoted as $E_\pi[r(s, a)] = E[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)]$. The occupancy measure $\rho_\pi : S \times A \to \mathbb{R}$ of a policy π is defined as $\rho(s, a) = (1-\gamma)\pi(a|s) \sum_{t=0}^{\infty} \gamma^t P(s = s_t|\pi)$. The expected reward can be expressed in terms of occupancy measures as $E_\pi[r(s, a)] = \sum_s \sum_a \rho_\pi(s, a) r(s, a)$. For brevity, we sometimes use ρ to denote the occupancy measure of a policy π. It is worth mentioning that there exists a one-one mapping between a policy and its corresponding occupancy measure [26]. For rest of this paper, we rely on this result to use π and ρ interchangeably.

Deceptive Reinforcement Learning Let R be the set of all reward functions, then a *deceptive reinforcement learning* problem is defined by the tuple, $(S, A, P, r, \gamma, \mu, L_R^\pi)$, where S, A, P, r, γ, μ are the same as defined for a regular MDP, and $L_R^\pi(s, a)$ stands for

deception-inducted reward function [20] which combines the objective of reward maximization with deception.

A *deceptive* policy maximises the objective,

$$J_D = E_\pi[L_R^\pi(s,a)] \tag{1}$$

We do not make any assumptions about the knowledge of the observer similar to [21]. In such a case, L_R^π is a weighted mixture of the reward function and the deception level [17],

$$L_R^\pi(s,a) = \omega r(s,a) + d_R^\pi(s,a) \tag{2}$$

where $d_R^\pi(s,a) \in \mathbb{R}$ is a measure of deception, and $\omega \in \mathbb{R}$ controls the trade-off between reward maximization and deception. For example, $d_R^\pi(s,a)$ could be the entropy of the policy, i.e, $-\log \pi(a|s)$, leading to deception by dissimulation.

Maximum Entropy RL (MERL) The objective of Maximum Entropy Reinforcement Learning (MERL) [12] is to optimize both the value function and the entropy of the agent's policy. Formally,

$$\text{RL}(r) = \underset{\pi}{\text{argmax}} \ \Big\{ E_\pi\left[r(s,a)\right] + \mathcal{H}(\pi) \Big\} \tag{3}$$

where r is the reward function and $\mathcal{H}(\pi) \triangleq E_\pi[-\log(\pi(a|s))]$ is the γ-discounted casual entropy. MERL is an important algorithm in this paper as the MCE-IRL model (discussed below) is built on MERL. In addition, we showcase in subsequent sections that the MEIR algorithm (discussed below) is also an instance of MERL making it a "dissimulation" based Deceptive RL algorithm.

Maximum Entropy Intentional Randomization (MEIR) The MEIR algorithm is a private RL algorithm that is driven by the concept that maximizing the entropy of the policy $\mathcal{H}$ will keep the reward function private. MEIR solves the following optimization problem,

$$\begin{aligned} \text{MEIR}(r, E_{min}) = \ & \underset{\pi}{\text{argmax}} \ \ \mathcal{H}(\pi) \\ & \text{subject to} \ \ E_\pi[r(s,a)] \geq E_{min} \end{aligned} \tag{4}$$

where $E_{min} \in [\hat{E}, E^*]$ is the reward threshold. $\hat{E} = E_{\hat{\pi}}[r(s,a)]$ and $E^* = E_{\pi^*}[r(s,a)]$ where $\hat{\pi}$ and π^* correspond to the uniform random and optimal deterministic policies, respectively. The reward threshold E_{min} is used to control the privacy/reward tradeoff.

Maximum Causal Entropy IRL (MCE-IRL) The Maximum Causal Entropy Inverse Reinforcement Learning [31] model has emerged as the most prominent method to infer an unknown reward function from demonstrations. Given the occupancy measure $\tilde{\rho}$ of an agent (calculated from demonstrations) MCE IRL recovers a reward function based on the following formulation,

$$\text{MCE-IRL}(\tilde{\rho}) = \ \underset{r}{\text{argmax}} \ \min_{\rho} \ E_{\tilde{\rho}}[r] - E_\rho[r] - \mathcal{H}(\rho)$$

MCE-IRL is closely linked with MERL as the occupancy measures of the agent is obtained from the recovered reward function $\tilde{r}$ as,

$$\tilde{\rho} = RL(\tilde{r}) \tag{5}$$

3 Assessing Learnt Reward, $\tilde{r}$, from IRL

Before we describe our contributions, we describe mechanisms to evaluate whether the original preferences (reward) are captured by the reward learnt by an observer (using IRL). It is challenging to assess the quality of the recovered reward function due to the inherent ambiguity in the Inverse RL problem. The ambiguity is on account of multiple reward functions being able to explain the demonstrated behaviour. To evaluate the quality of the recovered reward function, we adopt the framework proposed in [25], which introduces three quality standards.

The first standard implies the preservation of the "ordering" of policies with respect to the true reward function, r in the recovered reward function, $\tilde{r}$ as an indicator of the highest quality. A policy π_1 is better ($\succeq$) than a policy π_2 if the expected value with policy π_1 is higher than the expected value of π_2.

$$\begin{aligned} &\pi_1 \succeq \pi_2 \iff \\ &\quad E_{\pi_1}[r(s,a)] \geq E_{\pi_2}[r(s,a)] \wedge E_{\pi_1}[\tilde{r}(s,a)] \geq E_{\pi_2}[\tilde{r}(s,a)] \end{aligned}$$

The second standard implies that the recovered reward function, $\tilde{r}$ shares the same set of optimal policies as the true reward function.

$$\underset{\pi}{\text{argmax}} E_\pi[r(.,.)] = \underset{\pi}{\text{argmax}} E_\pi[\tilde{r}(.,.)]$$

Lastly, the third criterion implies that the recovered reward function fails to preserve any of these desirable properties, indicating a lower level of quality in learning.

By applying these three quality standards, we can assess and compare the effectiveness of the recovered reward function. Matching optimal policies between the true and recovered reward functions signifies valuable insights into the agent's optimal trajectories. If policy ordering is preserved, the observer not only gains trajectory insights but also learns their order, increasing the chances of unveiling the agent's encoded preferences in the reward function.

Formally, Let R be the set of all possible reward functions $r : S \times A \to \mathbb{R}$ with state space S and action space A, and $M < S, A, P, , \gamma, \mu >$ be an MDP without a reward function. Let partitions OPT^M and ORD^M be defined on R as follows: given two reward functions r_1 an r_2, we say that $r_1 \equiv_{OPT^M} r_2$ if $< S, A, P, r_1, \gamma, \mu >$ and $< S, A, P, r_2, \gamma, \mu >$ have the same set of *optimal policies*, and $r_1 \equiv_{ORD^M} r_2$ if $< S, A, P, r_1, \gamma, \mu >$ and $< S, A, P, r_2, \gamma, \mu >$ have the same *ordering over policies* [3].

If two reward functions have the same *ordering* of policies, then they have the same set of *optimal* policies (Section 2.4 in [25]).

4 Privacy Leak in MEIR

We study the privacy leak of MEIR in two situations: (i) Observer has access to the agent's true occupancy measure; (ii) Observer obtains a few demonstrations instead of the true occupancy measure. For (i), we can theoretically prove that there exists a privacy leak. For (ii), we provide a bound on the quality of the recovered reward function $\tilde{r}$ w.r.t r in terms of the distribution over policies they induce.

4.1 True Occupancy Measure: MEIR = MERL

We show this by hypothesizing that any policy that is a solution for MEIR can be computed by solving the MERL problem, where the rewards are multiplied by a positive scalar.

Lemma 1. *Any policy $\bar{\pi}$ that is the solution of a Max Entropy Intentional Randomization formulation MEIR(r, E_{min}) with a reward constraint $E_{min} \in [\hat{E}, E^*]$, can be expressed as the solution of the Maximum Entropy RL problem as,*

$$\bar{\pi} = RL(\lambda^* r) \tag{6}$$

for some $\lambda^* \geq 0$.

[3] **Notation**: $x \equiv_P y$ denotes x and y belong to the same partition P.

Lemma 1 shows that MEIR algorithm implicitly solves the RL objective with an additional temperature parameter λ (dual optimum) to intentionally control the trade-off between reward and entropy maximization. The RL objective can also be viewed as a belief inducted reward function L_R^π of the form 2, where the measure of deception $d_R^\pi(s,a) = -\log \pi(a|s)$. As $\lim_{\lambda \to 0}$, the entropy term dominates the reward term and we obtain a uniform random policy. Similarly, as $\lim_{\lambda \to \infty}$, reward dominates the entropy term and we obtain the optimal deterministic policy. Proof for Lemma 1 is provided in the Appendix [6]. We now prove that MEIR suffers a privacy leak when used against an MCE-IRL-based observer.

Theorem 1. *For an MDP M let $\bar{\pi} = MEIR(r, E_{min})$ for any reward constraint $E_{min} > \hat{E}$, and $\rho_{\bar{\pi}}$ be its corresponding occupancy measure. If $\tilde{r} = MCE\text{-}IRL(\rho_{\bar{\pi}})$ is the reward function recovered by an observer using Maximum Entropy IRL, then $r \equiv_{ORD^M} \tilde{r}$.*

Theorem 1 states that the observer will recover a reward function that respects the *ordering* of policies in the true reward function *irrespective* of the reward threshold. This indicates that $\tilde{r}$ and r share the same set of optimal policies as well. The proof for theorem 1 is built on the following lemma:

Lemma 2. *(Based on Theorem 3.4 in [25]) For any two scalars, $\lambda_1, \lambda_2 \in \mathbb{R}^+$ and two reward functions r_1, r_2, if we have $RL(\lambda_1 r_1) = RL(\lambda_2 r_2)$, then $r_1 \equiv_{ORD} r_2$.*

The proof for lemma 2 is given in Appendix [6]. Lemma 2 states that if two reward functions yield the same policy when optimizing the objective 3 with any positive weight assigned to the reward term, then the two reward functions have the same ordering over policies.

Proof of Theorem 1. Let $\bar{\pi} = \text{MEIR}(r, E_{min})$ be the randomized policy for any reward threshold $E_{min} > \hat{E}$. From Lemma 1, we know that $\bar{\pi} = RL(\lambda^* r)$, where $\lambda^* > 0$. Let $\tilde{r} = \text{MCE-IRL}(\rho_{\bar{\pi}})$ be the reward function recovered by the observer using MCE-IRL. From Equation 5, we know that $\bar{\pi} = \text{RL}(\tilde{r})$. Hence,
$\text{RL}(\lambda^* r) = \text{RL}(\tilde{r}) \implies r \equiv_{ORD^M} \tilde{r}$ (Lemma 2) □

4.2 Limited Demonstrations:

In section 4.1, we showed that the policy generated by the MEIR algorithm can be represented as a solution to the MERL objective, i.e, $\text{MEIR}(r, E_{min}) = RL(\lambda^* r)$. The solution of the MERL objective can also be interpreted as a mixture policy over the set of all stochastic policies Π, with the weight given to each policy proportional its value. Formally, a mixture policy π^{mix} contains a set of policies $\{\pi_1, ..., \pi_n\}$, and a distribution w over these policies. Before each episode, a policy is sampled according to w and executed for the entire trajectory. The probability of a sampling policy $\pi \in \Pi$ is given by [31],

$$P^{MERL}(\pi|r) \propto E_\pi[r] \tag{7}$$

Thus, learning this distribution (or a reward function that *induces* this distribution over policies) will give an observer insight into the ordering over policies in r. During execution however, from Lemma 1, we know that the agent samples a policy according to the distribution,

$$P^{MERL}(\pi|\lambda^* r) \propto E_\pi[\lambda^* r]$$

where $\lambda \geq 0$ which the observer learns via maximum likelihood (MCE-IRL). This distribution preserves the same ordering of policies in r and hence the accurate estimation of this distribution by an observer poses a significant privacy leak.

The quality of the distribution learned (Total Variation (TV) distance from the true distribution) is highlighted in Proposition 1.

Proposition 1. *For any MDP M, let $\bar{\pi} = MEIR(r, E_{min})$ for any reward constraint $E_{min} > \hat{E}$, i.e, $\bar{\pi} = RL(\lambda r)$ for some $\lambda > 0$ (Lemma 1) and let $\rho_{\bar{\pi}}$ be the empirical occupancy measure of n demonstrations obtained by executing π in M. If $\tilde{r} = MCE\text{-}IRL(\rho_{\tilde{\pi}})$, then,*

$$Pr(TV(P^{MERL}(\pi|\lambda^* r), P^{MERL}(\pi|\tilde{r})) > \epsilon) \leq \delta \tag{8}$$

where, $\delta = \Theta(e^{|\Pi| - n\epsilon^2})$

Proposition 1 directly follows from [2] given that $P(\pi|\tilde{r})$ is the maximum likelihood estimator of $P(\pi|\lambda r)$.

5 The Max Misinformation Algorithm

To address the privacy leak of MEIR, we introduce a novel algorithm referred to as the Max Misinformation (MM) algorithm. MM uses a deceptive measure called an anti-reward to incentivize the agent to *stay away from* the optimal trajectories. That is to say, MM intentionally leads the agent to take sub-optimal trajectories, in a bid to fool the observer into believing that these trajectories are highly rewarding. This makes MM a *simulation* based Deceptive RL algorithm.

Formally, let $r^-(s,a)$ be an anti-reward that induces the agent to take sub-optimal trajectories, then the MM algorithm solves the following constrained optimization problem,

$$\begin{aligned} \text{MM}(r, r^-, E_{min}) = \ & \underset{\pi}{\text{argmax}} \ E_\pi[r^-(s,a)] \\ & \textbf{s.t.} \ E_\pi[r(s,a)] \geq E_{min} \end{aligned} \tag{9}$$

where $E_{min} \in [E^-, E^*]$ is the reward threshold that is used to control the privacy-expected reward tradeoff. $E^- = E_{\pi^-}[r(s,a)]$ and $E^* = E_{\pi^*}[r(s,a)]$ where π^- and π^* correspond to the optimal policies with respect to the anti-reward, r^- and actual reward, r. We describe mechanisms for computing anti-reward in Section 5.2.

The MM formulation is a linear program (Appendix [6]) and hence the primal optimum $\bar{\pi}$ can be uniquely recovered from the dual optimum λ^* [1, Section 5.5.5] as,

$$\bar{\pi} = \underset{\pi}{\text{argmax}} \ E_\pi[\lambda^* r(s,a) + r^-(s,a)] \tag{10}$$

Equation 10 is a form of the Deceptive RL objective 2, where $d_R^\pi(s,a) = r^-(s,a)$ and the dual variable λ^* acts as a temperature parameter controlling the trade-off between reward and deception maximization. As $\lim_{\lambda^* \to 0}$, the anti-reward dominates the reward resulting in π^-, and as $\lim_{\lambda^* \to +\infty}$ the reward dominates the anti-reward resulting in π^* as the solution to Equation 10.

The linear program formulation of MM presented in Appendix [6] relies on known model dynamics. However, this assumption is often impractical in real-world scenarios. To address this limitation, we introduce Algorithm 1, which demonstrates how to solve Equation 9 using primal-dual descent without requiring explicit knowledge of the model dynamics. This formulation is particularly useful in the context of large MDPs with continuous state and (or) action spaces. In such scenarios, solving the primal problem to convergence (Line 4 of Algorithm 1) can be very time-consuming. Instead, one could take a few steps towards maximizing the objective function: Line 4 and then incrementally optimize λ and so on.

In the case of discrete MDPs, where solving the primal problem is much faster, we can use binary search to speed up the optimization procedure significantly as highlighted in Appendix [6].

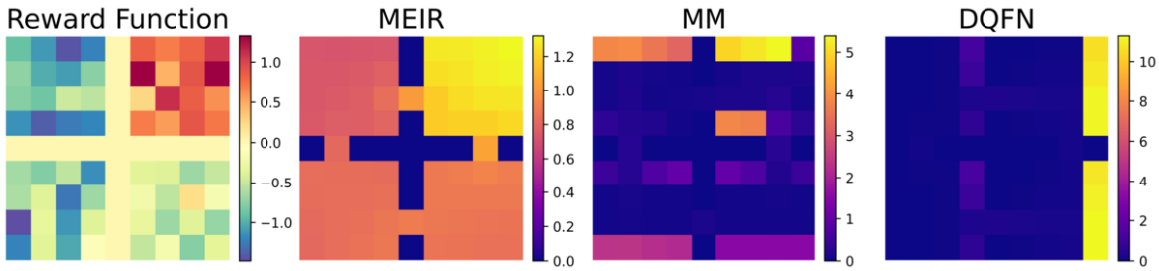

Figure 3. Occupancy measures of different private policies satisfying the same reward constraint in the Four Rooms environment. The MM algorithm leads to policies that visit a diverse mix of high reward and low reward states.

Algorithm 1 Max Misinformation via Primal-Dual Descent

Input: Anti-reward r^-, $E_{min} \in [E^-, E^*]$
Initialize temperature parameter $\lambda >= 0$, learning rate α
for t in {0, 1, 2, . . .} **do**
  $\bar{\pi}$ = max $E_\pi[\lambda_t r(s,a) + r^-(s,a)]$
  $\lambda_{t+1} \leftarrow \lambda_t - \alpha \nabla_\lambda [\lambda[E_{\bar{\pi}} r(s,a) - E_{min}]]$

5.1 *Security of the Max Misinformation Algorithm*

In Section 4, we highlighted that the privacy leakage in the MEIR algorithm was due to the fact that MEIR is an instance of MERL, which preserves the ordering of policies which can be learnt efficiently by the observer. The proposed MM algorithm addresses this privacy leakage as the addition of an anti-reward does not preserve the ordering over policies as it assigns a high value to sub-optimal trajectories. Consequently, it does not preserve the set of optimal policies either.

The difference between the occupancy measures of the MEIR and the MM algorithms are highlighted in Figure 3. Despite the MEIR policy exhibiting higher entropy, it readily reveals locations with high rewards. In contrast, the MM policy visits a nuanced mix of high and low reward states, making it more challenging to discern important locations.

5.2 *Generating anti-reward functions*

We now describe mechanisms for generating anti-reward functions that maximize deception by steering an agent away from the optimal trajectories.

Ideally, we would like to maximize the distance between the true reward function and the recovered reward function, but this would make the deceptive policy specific to an IRL algorithm (as recovered reward function is dependent on the algorithm). Instead, to ensure robustness against the observer reward recovery methods (IRL or some other mechanism), we propose a mechanism that is agnostic to the specific algorithm utilized to recover the reward function.

Intuitively, we compute an anti-reward that maximizes the distance between a distribution/statistic corresponding to the optimal policy for the original reward and optimal policy for the anti-reward. This will ensure that observer receives minimal information about the optimal policy for the original reward. Let o be a distribution/statistic that can be computed from the agent's reward function r. Two examples of o would be the policy computed or occupancy distribution corresponding to the reward function r. We can generate an anti-reward function r^- by maximizing the distance between o^* and o^-, where o^* is observed when behaving optimally according to r and o^- is observed when behaving optimally according to r^-. The algorithm for computing the anti-reward is provided in Algorithm 2. Let C be the function that maps from r to o. We iteratively do the following steps by starting from a randomly initialised o^-: (1) Set r^- as the anti-reward function that maximises the distance between o^* and o^- (2) Compute the new o^- from the new value of r^-. We repeat this process for a set number of iterations. An intuition behind why this approach works is highlighted in Appendix [6].

Algorithm 2 Generating Anti-Reward functions

Input: Distance Metric D, c, o^*
Initialize o^-
for t in {0, 1, 2, . . .} **do**
  $r^- = \underset{r}{\text{argmax}}\ D(o^*, o^-)$
  $o^- = C(r^-)$

We will now outline the various forms of o (occupancy measures and trajectory distributions) and the corresponding distance measures, denoted as D, applied in each case.

Occupancy Measures Since IRL algorithms try to match occupancy measures, one could try to directly maximize the distance between the occupancy measures of the optimal policy of r, i.e, ρ^* and ρ^-. Hence, in this case, $o = \rho$. We use f-divergences and Integral Probability Metrics (IPMs) to measure the distance between ρ^* and ρ^-.

The f-divergence between ρ and ρ^* is defined using the convex conjugate f^* as,

$$D_f(\rho^* || \rho^-) = \sup_{g: D \to R} E_{\rho^*}[g(s,a)] - E_{\rho^-}[f^*(g(s,a))]$$

setting $g = -r^-$ and $\phi(u) = -f^*(-u)$, so that E_{ρ_-} is maximized,

$$D_f(\rho^* || \rho^-) = \sup_{r^-: D \to R} E_{\rho_-}[\phi(r^-(s,a))]] - E_{\rho^*}[r^-(s,a)] \tag{11}$$

IPMs that are parameterized by a family of functions $\mathcal{F}$ are defined,

$$\gamma_{\mathcal{F}}(\rho^-, \rho^*) = \sup_{f \in \mathcal{F}} |E_{\rho^-}[f(s,a)] - E_{\rho^*}[f(s,a)]| \tag{12}$$

We can see that in both IPMs (Equation 12) and f-divergences (Equation 11), the anti-reward function gives a high reward to the state-action pairs visited by π^- (due to the sup and expectation over ρ^- being the first term) and a low reward to the ones visited by π^*. In both these cases, the anti-reward function can be represented using a function approximator and solved using gradient ascent, or in the case of discrete environments, Equation 11 can be solved using the closed form solution (described in the Appendix [6]).

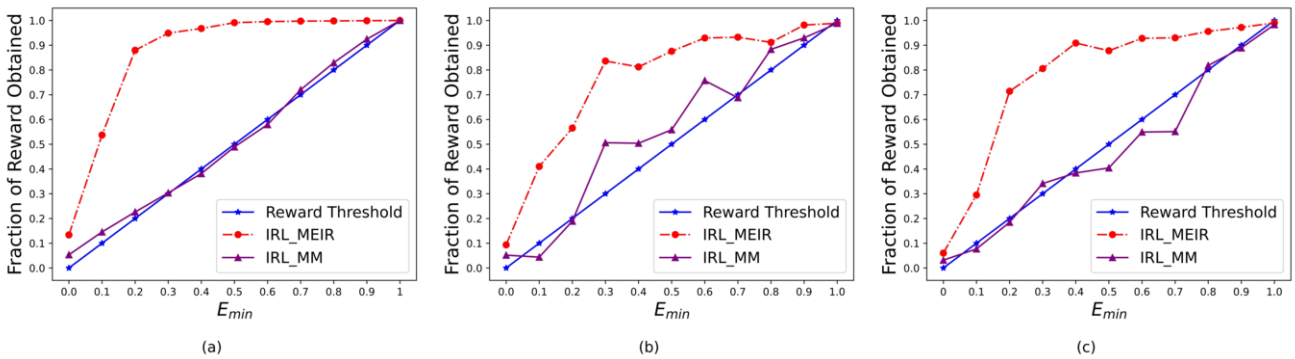

Figure 4. MM and MEIR against IQ-Learn given 10 demonstrations. Figures correspond to (a) Cyber security domain, (b) Frozen Lake and (c) Random MDPs.

Trajectory Distributions We can observe from Equation 7 and Figure 3 that the MEIR algorithm suffers from the problem of preferring highly rewarding trajectories (policies in the case of stochastic dynamics). To avoid this, we can generate an anti-reward function that maximizes the distance between the trajectory distributions of π^* and π^-. Consider the objective of maximizing the KL-divergence, between the distribution over trajectories induced by a policy π and the optimal policy π^*,

$$\begin{aligned} J^- &= \underset{\pi}{\text{argmax}} \; KL\left(p(\tau)||p(\tau^*)\right) \\ &= \underset{\pi}{\text{argmax}} \; \int p(\tau) \log \left(\frac{\prod_t p(s_{t+1}|s_t,a_t)\pi(a_t|s_t)}{\prod_t p(s_{t+1}|s_t,a_t)\pi^*(a_t|s_t)} \right) d\tau \\ &= \underset{\pi}{\text{argmax}} \; -\mathcal{H}(\pi) + E_{\tau\sim\pi}[\sum_{t=0}^{T} -\log \pi^*(a_t|s_t)] \\ &= \underset{\pi}{\text{argmax}} \; E_{\tau\sim\pi}[r^-_{KL}(s_t,a_t)] \end{aligned} \tag{13}$$

This formulation is the standard RL objective with the anti-reward function $r^{KL}_- \triangleq -\log(\pi^*(a|s))$ [4]. We can drop the entropy term as the formulation calls for entropy *minimization* given that an MDP has an optimal deterministic solution, i.e., a policy with 0 entropy.

6 Experiments and Results

In this section, we intend to answer the following key questions: (1) Does the MEIR algorithm as described in Section 4 suffer a significant privacy leak in the case of limited demonstrations, and how does the MM algorithm perform in comparison? (2) How does the MM algorithm fare in comparison to the MEIR and DQFN algorithms in preserving the privacy of the reward function when the observer has access to the true occupancy measures of the agent? (3) How does MM perform against observers that know they are being deceived?

6.1 Environments and Evaluations Metrics

We conduct our experiments in the following environments: Cyber Security which is based on Moving Target Defence [24], Frozen Lake [27], Four Rooms [5] and randomly generated MDPs. In the Cyber Security [24] environment, the state space consists of network configurations generated by the CyberBattleSim library [23] that emulates real-world active directory networks. The value associated with a network configuration corresponds to the ease with which malicious actors could compromise and gain control over the network. Agent's objective is to dynamically switch between network configurations to bolster security. The intention behind using Frozen Lake [27] and Four Rooms [5] from the standard Gym environment draws inspiration from tangible real-world security challenges such as Police Patrolling [4] and Green Security Games (GSGs) [7]. Randomly generated MDPs can be representative of a broad set of domains in general. A more detailed description is provided in Appendix [6]. The reward function in the above domains reflects the user's preferences, encompassing critical aspects such as the valuation of patrol locations, density of animals in various regions, and significance of distinct network configurations, all of which must be kept private.

We consider Inverse Reinforcement Learning (IRL) as the main method for reward function reconstruction due to two primary considerations: (a) Observer does not know that they are being deceived, which is the assumption made in the prior deception based methods as well [17, 16] (b) Recovering the reward function from deceptive demonstrations is not a trivial task. Consider the example in Figure 3 - the agent visits multiple diverse locations and deducing the high reward states from them is not easy for the observer. In addition, there does not exist any prior work in this space to benchmark our algorithms against. In such a case, irrespective of whether the observer knows that they are being deceived, learning a reward function that maximises the likelihood of the demonstrations is a strong strategy for the observer given that they are guaranteed at least E_{min} reward.

In the IRL space, MCE-IRL based methods have demonstrated superior performance in recent times for reward reconstruction [13, 9] and hence we use MCE-IRL and IQ-learn [9] (a variant of MCE-IRL that performs well in the limited demonstrations setting) in our experiments. Furthermore, we introduce two additional baselines that try to account for observers that are aware that they are being deceived by the usage of the MM algorithm. Given that the observer knows that the agent is intentionally visiting sub-optimal states along with the optimal states, the observer can cluster the occupancy measures of the agent and selectively recover a reward function that matches just one (or more) of them. See Figure 3 - each cluster either contains the optimal states or the misleading states intentionally visited by the agent. Based on how we select the cluster(s), we split these methods into: (a) IRL$^{\text{random}}$ that picks a cluster at random and (b) IRL$^{\text{max}}$ that greedily picks the cluster that has the highest occupancy measure.

We use three metrics to evaluate the quality of the reward function learned by the IRL algorithms, namely: (1) Pearson Correlation: high

[4] If π^* is deterministic $\log(\pi^*(a|s))$ is not defined when $\pi^*(a|s) = 0$. This can be avoided by setting π^* as the a solution of MERL which ensures all actions have non-zero support.

Table 1. MM and the DQFN algorithms against an MCE IRL based observer with access to the true occupancy measure.

Env	Algorithm	Avg. Pearson	Avg. EPIC
Random MDP	DQFN	0.31	0.58
	MM (τ-based)	**0.30**	**0.58**
	MM (ρ-based)	0.67	0.39
Four Rooms	DQFN	0.09	0.67
	MM (τ-based)	**0.03**	**0.69**
	MM (ρ-based)	0.22	0.62
Frozen Lake	DQFN	0.11	0.67
	MM (τ-based)	**0.05**	**0.72**
	MM (ρ-based)	0.34	0.57

value indicates better learning; (2) EPIC distance [11]: low value indicates better learning); and (3) Evaluation of the optimal policy of the recovered reward function in the original reward function: higher expected reward implies better learning.

6.2 *Analysis of Results*

Privacy Leak in MEIR and the efficacy of MM Figures 4, 5 and 6 can be interpreted as follows: for a given private RL algorithm. First, we specify a reward constraint E_{min} that is represented on the x-axis. Next, the algorithms return a private policy with a return $\geq E_{min}$ that is plotted on the y-axis. The Reward Threshold line indicates the return of the generated policy. Next, an observer takes the occupancy measure of the private policy as input and recovers a reward function $\tilde{r}$. The return of the optimal policy $\pi^*_{\tilde{r}}$ of $\tilde{r}$ when evaluated in r is represented using the IRL line.

From Figure 4, we can infer that the reward recovered by $\pi^*_{\tilde{r}}$ is very high, almost the same as the optimal policy π^*_r, even when E_{min} is very low and when the observer is given just 10 trajectories to learn from. Therefore validating the insights presented in Section 4.2. MM algorithm does a much better job in preserving the reward privacy in this case. Figure 5 shows that MM significantly outperforms MEIR in the worst case, i.e when an observer has access to the true occupancy measures of the agent. A quantitative analysis for this case is present in the Appendix [6].

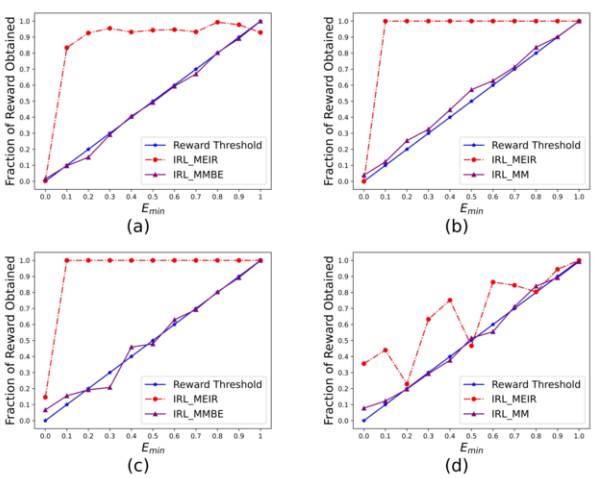

Figure 5. MM and MEIR against MCE IRL with true occupancy measures. Figures correspond to (a) Four Rooms (b) Frozen Lake (c) Random MDPs (d) Cyber Security domain.

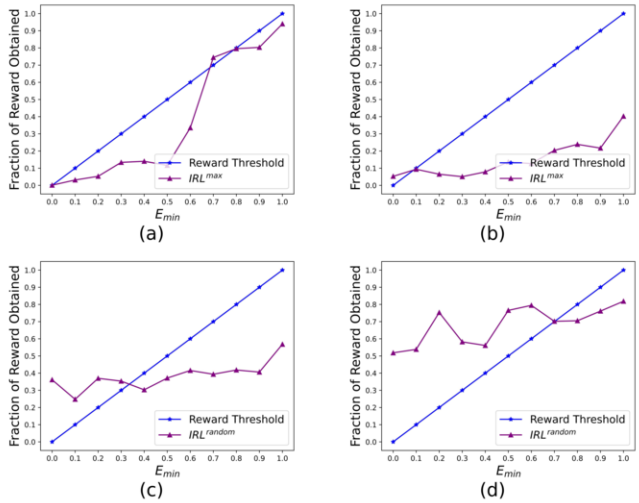

Figure 6. MM against IRL^{max} in (a) Four Rooms (b) Frozen Lake and $\text{IRL}^{\text{random}}$ in (c) Random MDP (d) Cyber Security domain.

Advantage over DP-based methods The quantitative comparison between the Deep Q-learning with Functional Noise (DQFN) [29], a state-of-the art algorithm in DP-based privacy methods and the MM algorithm in Table 1 demonstrates that the MM algorithm with a trajectory-based anti-reward outperforms the DQFN algorithm. Additionally, a qualitative comparison (Figure 3) shows that the MM algorithm visits a significantly more diverse set of misleading states compared to the DQFN algorithm. In addition, as discussed in Section 1, the MM algorithm has an advantage over existing DP-based methods based on the following: (1) There is a direct relationship between the privacy budget E_{min} and the reward obtained by the corresponding policy. The noise parameter in DQFN has no such relationship (Figure 2), making it challenging to control the privacy/reward tradeoff and (2) The quality of the recovered reward function is proportional to the privacy budget E_{min} as seen in Figures 4 and 5, another important property that DP-based algorithms do not posses [22].

Effectiveness against observers who are aware of the use of deception by the agent Figure 6 contains evaluations of MM against an observer that is aware of the use of deception and subsequently uses IRL^{max} / $\text{IRL}^{\text{random}}$ to recover the reward function. From Figure 6, we can infer (1) robustness of the MM algorithm in preserving the privacy of the reward function in this scenario and (2) weakness of these reward recovery mechanisms as compared to IRL indicating the difficulty of reward recovery when faced with deceptive demonstrations and the dominance of IRL as a strategy for the observer.

7 Conclusion and Future Work

RL-based planning algorithms have found applications in many domains including security related where protection of the reward function from potential observers becomes critical. Our study identifies vulnerabilities and limitations in existing methodologies and proposes the Max Misinformation (MM) algorithm as a solution. While our experiments are limited to discrete state/action spaces, MM can be used in continuous settings, making for future research endeavors. Furthermore, our research underscores the limitations of Inverse Reinforcement Learning (IRL) when confronted with deceptive demonstrations, prompting further exploration into Deceptive IRL.

Acknowledgments

This research/project is supported by the National Research Foundation Singapore and DSO National Laboratories under the AI Singapore Programme (AISG Award No: AISG2- RP-2020-016).

References

[1] S. P. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2014.
[2] C. L. Canonne. A short note on learning discrete distributions. *arXiv preprint arXiv:2002.11457*, 2020.
[3] T. L. Carson. *Lying and Deception: Theory and Practice*. Oxford University Press, 2010.
[4] X. Chen. Police patrol optimization with security level functions. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2013.
[5] M. Chevalier-Boisvert, B. Dai, M. Towers, R. D. L. Perez-Vicente, L. Willems, S. Lahlou, S. Pal, P. S. Castro, and J. K. Terry. Minigrid & miniworld: Modular & customizable reinforcement learning environments for goal-oriented tasks. In *Neurips Datsets and Benchmarks Track*, 2023.
[6] S. R. Chirra, P. Varakantham, and P. Paruchuri. Preserving the privacy of reward functions in MDPs through deception. *arXiv preprint arXiv:2407.09809*, 2024.
[7] F. Fang, P. Stone, and M. Tambe. When security games go green: Designing defender strategies to prevent poaching and illegal fishing. In *IJCAI*, 2015.
[8] M. Franchi, J. Zamfirescu-Pereira, W. Ju, and E. Pierson. Detecting disparities in police deployments using dashcam data. In *Proceedings of the ACM Conference on Fairness, Accountability, and Transparency*, 2023.
[9] D. Garg, S. Chakraborty, C. Cundy, J. Song, and S. Ermon. Iq-learn: Inverse soft-q learning for imitation. In *NeurIPS*, 2021.
[10] A. Gleave, M. Dennis, C. Wild, N. Kant, S. Levine, and S. Russell. Adversarial policies: Attacking deep reinforcement learning. In *ICLR*, 2020.
[11] A. Gleave, M. Dennis, S. Legg, S. Russell, and J. Leike. Quantifying differences in reward functions. In *ICLR*, 2021.
[12] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *ICML*, 2018.
[13] J. Ho and S. Ermon. Generative adversarial imitation learning. In *NeurIPS*, 2016.
[14] S. Keren, A. Gal, and E. Karpas. Privacy preserving plans in partially observable environments. In *IJCAI*, 2016.
[15] A. Kulkarni, M. Klenk, S. Rane, and H. Soroush. Resource bounded secure goal obfuscation. In *AAAI Fall Symposium on Integrating Planning, Diagnosis and Causal Reasoning*, 2018.
[16] A. Lewis and T. Miller. Deceptive reinforcement learning in model-free domains. In *ICAPS*, 2023.
[17] Z. Liu, Y. Yang, T. Miller, and P. Masters. Deceptive reinforcement learning for privacy-preserving planning. In *AAMAS*, 2021.
[18] P. Masters and S. Sardiña. Deceptive path-planning. In *IJCAI*, 2017.
[19] A. Y. Ng and S. J. Russell. Algorithms for inverse reinforcement learning. In *ICML*, 2000.
[20] M. Ornik and U. Topcu. Deception in optimal control. In *2018 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2018.
[21] P. Paruchuri, M. Tambe, F. Ordóñez, and S. Kraus. Security in multiagent systems by policy randomization. In *AAMAS*, 2006.
[22] K. Prakash, F. Husain, P. Paruchuri, and S. Gujar. How private is your rl policy? an inverse rl based analysis framework. In *AAAI*, 2022.
[23] C. Seifert, M. Betser, W. Blum, J. Bono, K. Farris, E. Goren, J. Grana, K. Holsheimer, B. Marken, J. Neil, N. Nichols, J. Parikh, and H. Wei. Cyberbattlesim. https://github.com/microsoft/cyberbattlesim, 2021.
[24] S. Sengupta, A. Chowdhary, A. Sabur, A. Alshamrani, D. Huang, and S. Kambhampati. A survey of moving target defenses for network security. *IEEE Communications Surveys and Tutorials*, 2020.
[25] J. Skalse and A. Abate. Misspecification in inverse reinforcement learning. In *AAAI*, 2023.
[26] U. Syed, M. Bowling, and R. E. Schapire. Apprenticeship learning using linear programming. In *ICML*, 2008.
[27] M. Towers, J. K. Terry, A. Kwiatkowski, J. U. Balis, G. d. Cola, T. Deleu, M. Goulão, A. Kallinteris, A. KG, M. Krimmel, R. Perez-Vicente, A. Pierré, S. Schulhoff, J. J. Tai, A. T. J. Shen, and O. G. Younis. Gymnasium. *Zenodo*, Mar. 2023. doi: 10.5281/zenodo.8127026.
[28] G. Vietri, B. Balle, A. Krishnamurthy, and Z. S. Wu. Private reinforcement learning with pac and regret guarantees. In *ICML*, 2020.
[29] B. Wang and N. Hegde. Privacy-preserving q-learning with functional noise in continuous spaces. In *NeurIPS*, 2019.
[30] X. Zhou. Differentially private reinforcement learning with linear function approximation. *Proc. ACM Meas. Anal. Comput. Syst.*, 2022.
[31] B. D. Ziebart, J. A. Bagnell, and A. K. Dey. Modeling interaction via the principle of maximum causal entropy. In *ICML*, 2010.