Model-Free Motion Planning of Complex Tasks Subject to Ethical Constraints

Shaoping Xiao(✉), Junchao Li, and Zhaoan Wang

The University of Iowa, Iowa City, IA 52242, USA
shaoping-xiao@uiowa.edu
https://xiao.lab.uiowa.edu

Abstract. Artificial Intelligence (AI) ethics establishes a moral framework to guide responsible AI technology use. This paper introduces a model-free Reinforcement Learning (RL) approach to address ethical constraints in motion planning problems, particularly in complex tasks within partially observable environments. Leveraging the Partially Observable Markov Decision Process (POMDP) for motion planning in environments with incomplete knowledge and Linear Temporal Logic (LTL) for task formulation, ethical norms are categorized as 'hard' and 'soft' constraints. Our approach involves generating a product of POMDP and LTL-induced automaton. An optimal policy is then learned, ensuring task completion while adhering to ethical constraints through model checking. To handle the situations where the agent lacks task awareness, we propose a novel modification to deep Q-learning. This model-free deep RL method employs a neural network architecture with environmental observations and recognized labels as inputs. An illustrative example showcases the applicability of our approach to motion planning problems. The flexibility and generality of this method make it suitable for addressing various ethical decision-making problems.

Keywords: Ethical constraints · Motion planning · Partially observable environments · Reinforcement learning

1 Introduction

Classical ethical theories, such as Utilitarianism [1], Deontology [2], virtue ethics [3], and consequentialism [4], significantly shape our daily lives, guiding ethical decision-making in various contexts. In the evolving landscape of Artificial Intelligence (AI), ethical considerations are paramount, given the potential societal impact on intelligent decision-making [5]. However, as AI technologies rapidly advance, the "black box" nature of their underlying models presents challenges in understanding the decision processes. It becomes necessary for AI systems to incorporate ethical considerations, ensuring their decisions not only meet the technical specifications but also align with social values and norms.

H. Degen and S. Ntoa (Eds.): HCII 2024, LNAI 14735, pp. 116–129, 2024.
https://doi.org/10.1007/978-3-031-60611-3_9

This paper introduces a model-free Reinforcement Learning (RL) approach to address ethical constraints in motion planning within complex tasks in partially observable environments. Specifically, we employ the Partially Observable Markov Decision Process (POMDP) to model motion planning in environments with incomplete knowledge and use Linear Temporal Logic (LTL) for task formulation. Ethical norms are classified into 'hard' and 'soft' categories. Our approach involves generating a product of POMDP and LTL-induced automaton. An optimal policy is learned, ensuring task completion while adhering to ethical constraints through model checking.

To address situations where agents lack awareness of assigned tasks, we present a novel solution utilizing modified Q-learning to learn optimal policies in partially observable environments. This model-free deep RL method employs a Q network architecture, incorporating Recurrent Neural Networks (RNNs) to process sequences of environmental observations and recognized labels as inputs.

In the following sections of this paper, we will provide a structured presentation. The initial part will cover essential preliminaries, followed by a detailed exploration of methodologies. This includes defining ethical constraints, outlining the process of generating a product POMDP, and presenting the algorithm of our proposed method. Subsequently, we will illustrate our approach through a practical example, demonstrating motion planning under diverse ethical constraints. Finally, we will conclude with a summary of key findings and potential avenues for future research.

2 Preliminary

2.1 Parially Observable Markov Decision Process (POMDP)

When an agent can not fully identify the state of its environment, POMDP [6] is typically employed to model the interaction between the agent and the environment.

Definition 1 (POMDP). *A tuple $\mathcal{P} = (S, A, s_0, T, R, O, \Omega)$ is utilized to denote a POMDP, including*

- *A set of states $S = \{s_1, \ldots, s_n\}$.*
- *A set of actions $A = \{a_1, \ldots, a_m\}$. Specifically, $A(s)$ is a set of available actions the agent can take at the current state s.*
- *An initial state $s_0 \in S$.*
- *A transition probability function $T : S \times A \times S \to [0, 1]$, satisfying $\sum_{s' \in S} T(s, a, s') = 1$. It defines the probability when the agent moves from the current state s to the next state s' after executing an action a.*
- *A reward function $R : S \times A \times S \to \mathcal{R}$. The reward function can sometimes be written as $R(s)$ or $R(s, a)$.*
- *A set of observations $O = \{o_1, \ldots, o_k\}$.*
- *An observation probability function $\Omega : S \times A \times O \to [0, 1]$, satisfying $\sum_{o \in O(s')} \Omega(s', a, o) = 1$. It represents the probability that the agent can perceive observation o at the next state s' after taking action a at the current state s.*

To address complex tasks, we utilize a set of atomic propositions Π to represent event occurrences. Additionally, we introduce a labeling function $L : S \rightarrow 2^{\Pi}$, where 2^{Π} is the power set of Π, to indicate events associated with individual states. This paper exclusively focuses on static events, meaning no probabilities are assigned to the occurrences of events.

2.2 Linear Temporal Logic (LTL)

Linear Temporal Logic [7], a formal language, is capable of expressing linear-time properties that represent the relation between state labels and sequential executions. In this study, we leverage LTL to articulate complex tasks. The basic operators encompass boolean connectors such as negation ($\neg$) and conjunction ($\wedge$), as well as temporal operators like "next" ($\bigcirc$) and "until" ($\mathcal{U}$). Assuming a word $\boldsymbol{w} = w_0 w_1 \ldots$ with $w_i \in 2^{\Pi}$, where $a \in \Pi$ is an atomic proposition, and ϕ, ϕ_1 and ϕ_2 are single LTL formulas, the grammar for forming an LTL formula and its semantics are expressed below [8].

$$\phi := \text{True} \mid a \mid \phi_1 \wedge \phi_2 \mid \neg\phi \mid \bigcirc\phi \mid \phi_1 \mathcal{U} \phi_2 \tag{1}$$

$$\begin{array}{ll}
\boldsymbol{w} \models \text{True} & \\
\boldsymbol{w} \models a & \Leftrightarrow a \in L(\boldsymbol{w}[0]) \\
\boldsymbol{w} \models \phi_1 \wedge \phi_2 & \Leftrightarrow \boldsymbol{w} \models \phi_1 \text{ and } \boldsymbol{w} \models \phi_2 \\
\boldsymbol{w} \models \neg\phi & \Leftrightarrow \boldsymbol{w} \not\models \phi \\
\boldsymbol{w} \models \bigcirc\phi & \Leftrightarrow \boldsymbol{w}[1:] \models \phi \\
\boldsymbol{w} \models \phi_1 \mathcal{U} \phi_2 & \Leftrightarrow \exists t \text{ s.t. } \boldsymbol{w}[t:] \models \phi_2, \forall t' \in [0, t), \boldsymbol{w}[t':] \models \phi_1
\end{array} \tag{2}$$

Other commonly-used temporal operators include "eventually" ($\Diamond\phi \equiv \text{True}\ \mathcal{U}$) and "always" ($\Box\phi \equiv \neg(\Diamond\neg\phi)$).

2.3 Limit-Deterministic Generalized Büchi Automaton (LDGBA)

Once complex tasks are expressed, an LTL formula can be transformed into a finite state automaton. This automaton takes a word as input and verifies temporal properties. In this study, we utilize LDGBA, which involves specific state transitions and allows the evaluation of task and constraint satisfaction through model checking [7].

Definition 2 (LDGBA). *A tuple $\mathcal{A} = (Q, \Sigma, \delta, q_0, \mathcal{F})$ is utilized to represent an LDGBA, which consists of*

- *A finite set of states Q, which can be decomposed into a deterministic set (Q_D) and a non-deterministic one (Q_N). The following relationships are satisfied: $Q_D \cup Q_N = Q$ and $Q_D \cap Q_N = \emptyset$.*
- *A finite alphabet $\Sigma = 2^{\Pi}$ where Π is a set of atomic propositions.*

- *A transition function $\delta\colon\ Q \times (\Sigma \cup \{\epsilon\}) \to 2^Q$, where ϵ-transitions do not take the input symbols. The state transitions satisfy the following requirements: (1) The transitions in Q_D are restricted, i.e., $\delta\,(q,\alpha) \subseteq Q_D$, for every state $q \in Q_D$ and $\alpha \in \Sigma$; (a) The state transitions in Q_D are total, i.e., $|\delta\,(q,\alpha)\,| = 1$; and (3) The ϵ-transitions are only valid from $q \in Q_N$ to $q' \in Q_D$.*
- *An initial state $q_0 \in Q$.*
- *A set of accepting sets $\mathcal{F} = \{\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_f\}$ where $\mathcal{F}_i \subseteq Q$, $\forall i \in \{1, \ldots, f\}$. It shall be noted that the accepting states in each accepting set belong to the deterministic set only, i.e., $\mathcal{F}_i \subseteq Q_D$ for every $\mathcal{F}_i \in \mathcal{F}$.*

After taking an input word $\boldsymbol{w} = w_0 w_1 \ldots$ where $w_i \in 2^{\Pi}$, the LDGBA generates a corresponding run $\boldsymbol{q} = q_0 q_1 \ldots$. This run is a sequence of automaton states determined by the transition function $\delta(q_i, w_i) = q_{i+1}$. The LDGBA accepts this word if the transitioned state eventually belongs to at least one of the accepting sets. Such a satisfaction condition can be mathematically expressed as $\inf\,(\boldsymbol{q}) \cap \mathcal{F}_i \neq \emptyset$, $\forall i \in \{1, \ldots f\}$ where $\inf\,(\boldsymbol{q})$ represents the infinite portion of $\boldsymbol{q}$. In other words, we can affirm that the run $\boldsymbol{q}$ satisfies the LDGBA's acceptance condition.

3 Methodologies

3.1 Ethical Constraints

This study categorizes various concepts of ethical norms into 'hard' and 'soft' constraints. Obligations and prohibitions are considered 'hard' ethical constraints and must be satisfied. These constraints can be formulated using LTL with temporal operators $\square$ ("always"). For instance, expressing the prohibition of event a can be done with the LTL formula $\square\neg a$, indicating that an acceptable sequence of agent's behaviors shall avoid all the actions leading to the occurrence of event a. Additionally, conditional obligations or prohibitions can be expressed by LTL formulas, such as $\square(a \to \bigcirc b)$, stating that the agent must take action to ensure event 'b' is true once event 'a' becomes true. It is important to note that LTL was employed to express complex tasks, as discussed above. Therefore, some atomic propositions labeled on POMDP states in this study represent task events, while others are associated with 'hard' ethical constraints.

When ethical norms, such as permission, are not strictly prohibited or obligated for agents, they fall into the 'soft' constraints category, which LTL cannot express. In this study, an additional reward function is introduced as below when the selected actions lead to permissible ethical events.

$$R_s(s,a,s') = \begin{cases} R_s \ a \in A, \ \text{and } L(s') \in L_p \\ 0 \qquad\quad \text{otherwise.} \end{cases} \tag{3}$$

where L_p denotes the set of labels indicating the permissible ethical events.

The reward R_s is assigned with positive or negative values to distinguish permissions with encouragement from discouragement. Additionally, a large or small positive (negative) reward signifies strong or weak encouragement (discouragement). It is worth noting that there is no necessity to assign a reward to the occurrence of events with simple permissions.

3.2 Product POMDP

We have outlined a POMDP to represent partially observable environments, incorporating an additional reward function for 'soft' ethical constraints and LTL specifications to express complex tasks and 'hard' ethical constraints. Subsequently, the original problem can be reformulated by creating a Cartesian product of the POMDP and the LTL-induced LDGBA, referred to as the product POMDP.

Definition 3 (Product POMDP). *Given a POMDP $\mathcal{P} = (S, A, s_0, T, R, O, \Omega)$ and an LDGBA $\mathcal{A} = (Q, \Sigma, \delta, q_0, \mathcal{F})$, the generated product POMDP can be represented by a tuple $\mathcal{P}^\times = \mathcal{P} \times \mathcal{A} = (S^\times, A^\times, s_0^\times, T^\times, R^\times, O, \Omega^\times, \mathcal{F}^\times)$, consisting of*

- *A finite set of product states, $S^\times = S \times Q$ or $s^\times = \langle s, q \rangle \in S^\times$ where $s \in S$ and $q \in Q$.*
- *A finite set of actions, $A^\times = A \cup \{\epsilon\}$.*
- *An initial product state $s_0^\times = \langle s_0, q_0 \rangle \in S^\times$ where $s_0 \in S$ and $q_0 \in Q$.*
- *A transition function, $T^\times = S^\times \times A^\times \times S^\times \to [0, 1]$.*
- *A reward function $R^\times = S^\times \times A^\times \times S^\times \to \mathcal{R}$.*
- *An observation function $\Omega^\times = S^\times \times A^\times \times O \to [0, 1]$.*
- *A set of accepting sets $\mathcal{F}^\times = \left\{\mathcal{F}_1^\times, \mathcal{F}_2^\times, ..., \mathcal{F}_f^\times\right\}$ where $\mathcal{F}_i^\times = \{\langle s, q \rangle | s \in S; q \in \mathcal{F}_i\}$ and $i = 1, ...f$.*

The transition function describes the state transition probabilities on the product POMDP as

$$T^\times\left(s^\times, a^\times, s^{\times'}\right) = \begin{cases} T(s, a^\times, s')\ q' = \delta\left(q, l\right), l \in L(s'),\ \text{and } a^\times \in A \\ \quad 1 \qquad s' = s, a^\times \in \{\epsilon\},\ \text{and } q' \in \delta(q, \epsilon) \\ \quad 0 \qquad\qquad \text{otherwise.} \end{cases} \tag{4}$$

where $s^{\times'} = \langle s', q' \rangle$. The reward function in the product POMDP comprises two terms, one for acceptance conditions and the other for 'soft' ethical constraints, defined below.

$$R^\times(s^\times, a^\times, s^{\times'}) = R_a^\times(s^\times, a^\times, s^{\times'}) + R_s^\times(s^\times, a^\times, s^{\times'}), \tag{5}$$

and

$$R_a^\times(s^\times, a^\times, s^{\times'}) = \begin{cases} R(s, a^\times, s')\ a^\times \in A, l \in L(s'), q' = \delta\left(q, l\right) \in \mathcal{F}_i \\ \quad 0 \qquad\qquad \text{otherwise.} \end{cases} \tag{6}$$

$$R_s^\times(s^\times, a^\times, s^{\times'}) = \begin{cases} R_s(s, a^\times, s')\ a^\times \in A, l \in L(s'),\ \text{and } l \in L_p \\ \quad 0 \qquad\qquad \text{otherwise.} \end{cases} \tag{7}$$

The available actions in the product POMDP include the physical actions on POMDP and ϵ-transitions on LDGBA. If the agent takes a physical action $a^{\times} \in A$, the observation probability is

$$\Omega^{\times}(s^{\times\prime}, a^{\times}, o) = \Omega(s', a^{\times}, o) \tag{8}$$

Otherwise, if ϵ-transitions are selected, the agent stays at the same POMDP state, $s' = s$, although the LDGBA state transitioned, i.e., $q' = \delta(q, \epsilon)$. In this case, no observation is perceived.

In addition, the expected return on the product POMDP can be written as below if an agent starts from the initial state and follows a policy $\xi^{\times}$.

$$U^{\xi^{\times}}(s_0^{\times}) = \mathbb{E}^{\xi^{\times}}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t^{\times}, a_t^{\times}, s_{t+1}^{\times}) \Big| s_{t=0}^{\times} = s_0^{\times}\right] \tag{9}$$

3.3 Problem Definition

If a path $s_0 s_1 \ldots$ exists on the POMDP, the corresponding path $q_0 q_1 \ldots$ on the LDGBA can be derived via the labeling function and then automaton state transitions. Those two paths can be integrated to generate a path on the product POMDP. It can be stated that any feasible path $\sigma^{\xi^{\times}} = (s_0, q_0)(s_1, q_1) \ldots$ generated by the learned policy $\xi^{\times}$ on product POMDP $\mathcal{P}^{\times}$ shares the intersections between an accessible path over the original POMDP $\mathcal{P}$ and a word accepted by the LTL-induced LDGBA $\mathcal{A}$. Furthermore, from an optimal policy $\xi^{\times *}$ on the product POMDP $\mathcal{P}^{\times}$, we can derive an optimal policy ξ^* on the POMDP $\mathcal{P}$. Additionally, as the product POMDP includes LTL specifications represented by LDGBA $\mathcal{A}$, the input word that corresponds to a path generated from ξ^* on POMDP is accepted by the LTL-induced LDGBA. In other words, the specifications are satisfied.

In this study, we adopt the strategy for solving MDP problems with LTL specifications [8–12]: generating a product MDP and applying the model-checking technique. Since the generated product POMDP is a type of POMDP, an optimal policy aims to maximize the expected return in (9). In our POMDP setting, the agent receives labels, which are input symbols to the LTL-induced automaton, as part of the feedback. In addition, the agent is unaware of the complex task (i.e., unknown to the automaton transitions). Consequently, the observations and labels can be grouped as the input of the policy $\xi^{\times}(\mathbf{o}_t, \mathbf{l}_t)$ on the product POMDP $\mathcal{P}^{\times}$.

Problem 1. A product POMDP $\mathcal{P}^{\times} = \mathcal{P} \times \mathcal{A}$ is formed by combining a POMDP $\mathcal{P}$ describing a partially observable environment with 'soft' ethical constraints and an LDGBA $\mathcal{A}$ expressing LTL specifications ϕ for a complex task and 'hard' ethical constraints. The objective is to discover an optimal policy $\xi^{\times *}(\mathbf{o}_t, \mathbf{l}_t)$, where $\mathbf{o}_t$ and $\mathbf{l}_t$ represent the sequences of observations and labels on POMDP states, respectively, on the product POMDP $\mathcal{P}^{\times}$ for maximizing the expected return of $\mathcal{P}^{\times}$.

3.4 Q-Learning

This study employs Q-learning [13], a model-free RL method where agents lack knowledge of the transition probability function, observation probability function, and reward function. In MDP problems, an agent learns state-action values or Q values, denoted as $Q(s, a)$, through interactions with the environment. For large or infinite state spaces, deep Q-learning (DQN) [14] is often utilized, where Q values are approximated via Deep Neural Networks (DNNs), referred to as Q networks. Deep Q-learning consists of two Q-networks. One is an evaluation Q-network $Q_e(s, a; \theta_e)$, usually trained and updated at each step. The other is a target Q-network $Q_t(s, a; \theta_t)$ with fixed weights periodically copied from the evaluation Q-network. Note that θ_e and θ_t represent the network weights.

Algorithm 1. Deep Recurrent Q-Network for Product POMDP Problems.

```
Define ‘hard’ and ‘soft’ ethical constraints
Initialize LTL formula $\phi$ expressing complex tasks with ‘hard’ ethical constraints
  and POMDP $\mathcal{P}$ with ‘soft’ ethical constraints.
Convert $\phi$ to an LDGBA $\mathcal{A}$.
Construct the product POMDP $\mathcal{P}^{\times} = \mathcal{P} \times \mathcal{A}$.
Initialize the evaluation network $Q_E^{\times}$, the target network $Q_T^{\times}$, the replay memory
  $D$, the length of observation sequence $p$, the length of label sequence $k$, the learning
  rate $\alpha$, the discount factor $\gamma$, the total number of episodes $E$, the total number
  of steps $N$, the batch size $M$, and the number of steps $K$ to update the target Q
  network $Q_T^{\times}$.
while The current episode $e$ in $E$ do
    Randomly select a start state $s_0^{\times}$.
    while The current step $i$ in $N$ do
        Select a random action $a_i^{\times}$ if $i < p$; otherwise, select an action via the
  $\epsilon$-greedy technique.
        Obtain observation and label.
        Generate $\mathbf{o}_{i+1}$ and $\mathbf{l}_{i+1}$.
        Collect the rewards $r_i^{\times}$.
        Store the experience $\langle \mathbf{o}_i, \mathbf{l}_i, a_i^{\times}, r_i^{\times}, \mathbf{o}_{i+1}, \mathbf{l}_{i+1} \rangle$ in $D$.
        if $i > 0$ and $i\%M{=}0$ then
            Randomly select $M$ data samples as U($D$) from the replay memory.
            Compute $Q_{new}^{\times}$ for each data sample.
            Train $Q_E^{\times}$ by the batch of samples.
        end if
        if $i > 0$ and $i\%K{=}0$ then
            Pass the weights of $Q_E^{\times}$ to $Q_T^{\times}$.
        end if
    end while
end while
Training end and save the evaluation network $Q_E^{\times}$
```

In a partially observable environment, determining the current state solely from instant observations is not possible for the agent. However, the agent can make informed decisions based on the history of observations. In other words, the policy, representing the agent's function, maps a sequence of observations to the selected action. In such cases, Recurrent Neural Networks (RNNs), including Long Short-Term Memory (LSTM) [16], can replace DNNs in Q-networks to approximate Q values in DQN [15].

To address the problems defined in this study, an agent needs to collect information for decision-making in the product POMDP, encompassing both POMDP and LDGBA. Assuming the agent is unaware of the assigned task, i.e., lacking knowledge of LDGBA's transitions, its decisions rely on the observation and label sequences, serving as inputs to the Q-networks.

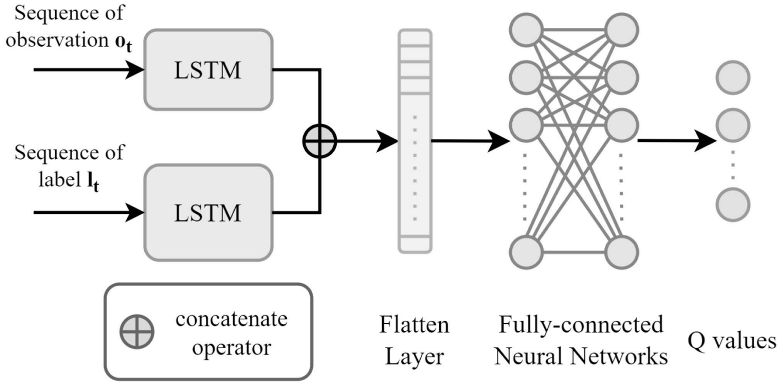
Sequence of observation $\mathbf{o_t}$
Sequence of label $\mathbf{l_t}$
LSTM
LSTM
concatenate operator
Flatten Layer
Fully-connected Neural Networks
Q values

Fig. 1. The architecture of Q networks taking $\mathbf{o}_t$ and $\mathbf{l}_t$ as inputs.

Denoting $\mathbf{o}_t = o_1 \ldots o_p$ as the sequence of observations at time step t and the corresponding sequence of state labels as $\mathbf{l}_t = l_1 \ldots l_k$, it is important to note that the sequence lengths k and p may not the same since not every state is labeled. Therefore, two input sequences are pre-processed by one-hot-encoding before entering LSTMs. The hidden states are then concatenated and flattened into a fully connected neural network to predict Q values. The Q-network architecture in our DQN is illustrated in Fig. 1. We employ two Q networks: the evaluation network $Q_E^{\times}(\mathbf{o}_t, \mathbf{q}_t, a_t^{\times}; \theta_E^{\times})$ and the target network $Q_T^{\times}(\mathbf{o}_t, \mathbf{q}_t, a_t^{\times}; \theta_T^{\times})$ where $a_t^{\times} \in A$. The details of the training process are provided in Algorithm 1.

4 Example

A company is in the process of constructing a nuclear power plant, facing opposition from local residents who are protesting for a permanent closure. The construction site, depicted in Fig. 2, is represented as a 10 × 10 grid. States 'a' and 'b' denote the inventory of construction materials and plant-building areas,

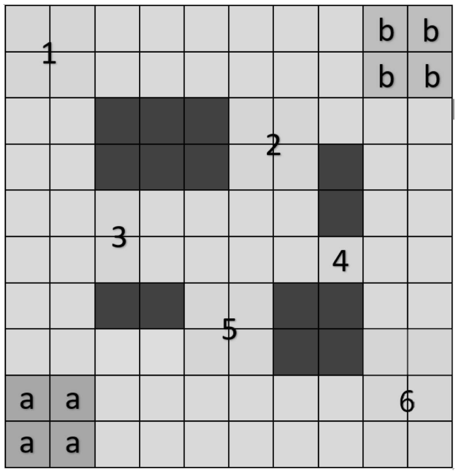

Fig. 2. Grid-world model of the environment of a power-plant construction.

respectively. An autonomous truck, acting as an agent in this example, transports materials between the inventory and the construction areas. Protesters, concentrated in locations labeled with numbers '1'−'6,' aim to impede the truck's route. Black-colored states represent company buildings, serving as impassible obstacles for the truck. The agent's primary goal is to deliver materials, navigating around protesters or cautiously passing them based on ethical constraints dictated by high-level decisions. These decisions, influenced by factors like local legislation, public safety, and economic impacts, reflect the ethical perspectives of protesters, the government, and the company. The example explores the agent's motion planning (pathfinding) in three scenarios with varying ethical constraints.

During the simulations, the agent's observation of the current state is assumed with a probability of 0.9 after taking an action. Alternatively, adjacent states (excluding those colored black) can be observed with a total probability of 0.1 uniformly distributed. Upon the agent's first visit to states 'a' or 'b' during a trip, a reward of 1 is received. In addition, there is an action cost of 0.01. Every simulation consists of 25,000 episodes, each with 800 steps. The observation sequence has a length of $p = 5$, and the label sequence length is $k = 3$. The batch size is set to $M = 32$ for training the evaluation Q-network at every time step. The target Q-network is updated by copying the weights of the evaluation Q-network every 50 time steps. Moverover, the discount factor $\gamma = 0.95$.

4.1 Scenario 1: Deontological Government and Company but Utilitarian Protesters

This scenario operates under the assumption that both the government and the company strictly adhere to local environmental and public safety legislation. According to those regulations, roads must always remain clear for public transportation. However, the protesters hold the opposite view, prioritizing environmental safety over temporary disruptions and economic losses for greater well-being. They have chosen to fully block streets in areas 1 and 2, while keeping areas 3 to 6 clear. To model this situation, we adopt the concept of 'hard' constraints and use the atomic proposition 'c' to represent the agent passing through areas 1 and 2. Consequently, the LTL formula for the request task with ethical constraints can be formulated as follows.

$$\phi_1 = \Box\Diamond(a \wedge \Diamond b) \wedge \Box\neg c \tag{10}$$

The LTL formula described above specifies that the agent is required to visit states 'a' and then 'b' in a repeated manner. Importantly, it also enforces the constraint that the agent should never pass through areas 1 and 2.

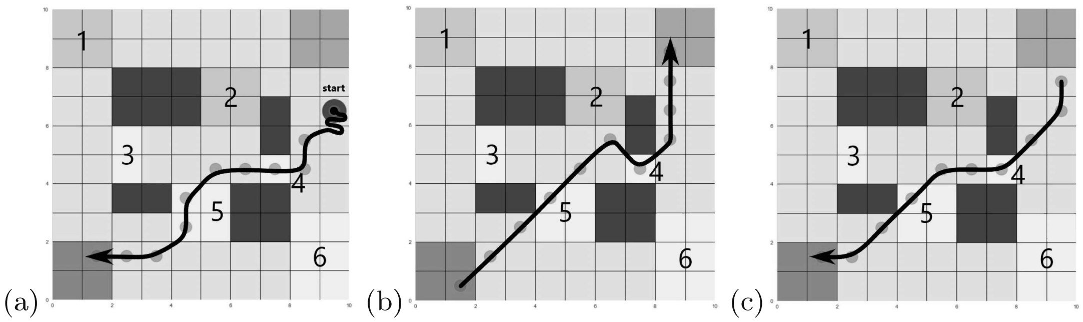

Fig. 3. A path to accomplish task ϕ_1 in case 1.

After the convergence of the learning process is converged and the acquisition of the optimal policy, we generate paths, as depicted in Fig. 3, to visually showcase the agent successfully completing the task. The agent initiates its journey near the construction area, taking random actions in 5 steps before generating the first observation and label sequences. Subsequently, the agent adopts a greedy approach in action selection based on the predicted Q values.

In Fig. 3(a), the agent navigates through areas 4 and 5, highlighted in light beige, where no protesters are present in these particular areas. It then reaches the inventory areas. After loading the materials, the agent follows the path outlined in Fig. 3(b), arriving at the construction areas and unloading the materials. In Fig. 3(c), the agent retraces its steps back to the inventory location for the second round. All paths traverse through areas 4 and 5, minimizing the total cost.

4.2 Scenario 2: Utilitarian Government and Company but Deontological Protesters

In the second scenario, both the government and the company believe that the construction project brings significant benefits to the community, outweighing the limited environmental impact. Consequently, they focus on prioritizing the construction while still permitting a certain degree of lawful protest. Unlike the first scenario, protesters in this case strictly adhere to the law, avoiding street blockages but potentially assembling near the company site within areas 3 to 6, prompting the agent to navigate those zones with caution. It is essential to note that areas 1 and 2 remain clear. As a result, soft constraints come into play, imposing negative rewards on the agent when passing through areas 3 to 6. Various reward values are considered for these soft constraints, illustrating permission with different levels of discouragement. Given that only soft constraints are at play in this scenario, the LTL formula for the assigned task is defined below.

$$\phi_2 = \Box\Diamond(a \wedge \Diamond b) \tag{11}$$

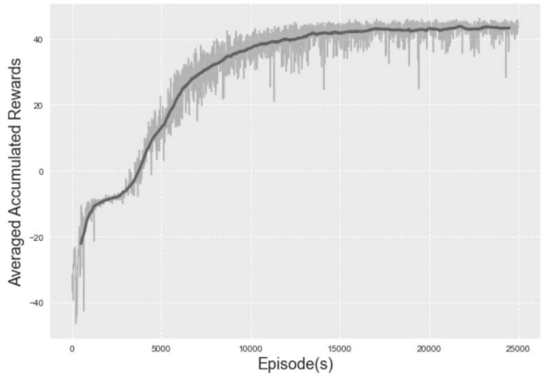

Fig. 4. The evolution of the cumulative reward.

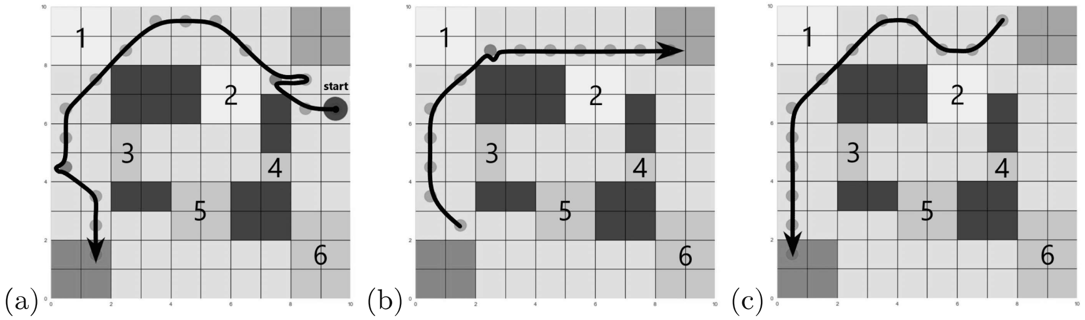

Fig. 5. A path to accomplish task ϕ_2 when permitted with strong encouragement.

In the initial variation, the agent is permitted but strongly discouraged from entering protest areas 3 through 6. Navigating through these areas incurs a negative reward of -0.3. During the learning process, the progression of accumulated rewards averaged every 10 episodes is depicted in Fig. 4. The darker color represents the Simple Moving Average (SMA) of rewards computed every 50 episodes.

The optimal policy emerges upon the convergence of cumulative rewards. Then, a path is generated and illustrated in Fig. 5. Commencing from the same location as in case 1, the agent systematically traverses the inventory and construction locations in order through areas 1 and 2, both devoid of protests. Despite the potential efficiency of passing through protest areas, specifically areas 4, 5, and 6, where fewer steps would be required, imposing high penalties steers the agent towards a route through areas 1 and 2. This strategic decision is influenced by the relatively higher costs incurred due to the strong discouragement associated with the occupied protest areas.

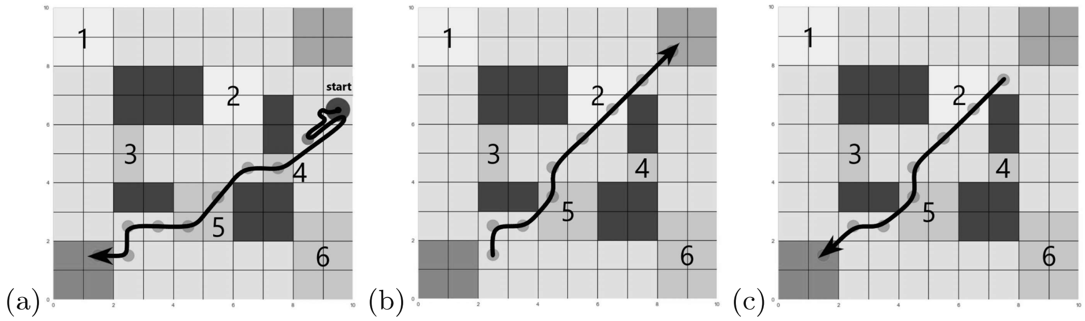

Fig. 6. A path to accomplish task ϕ_2 when permitted with weak encouragement.

When the penalties are reduced to -0.01 in the second variation, the agent experiences only weak discouragement from passing through protest areas 4 and 5. Consequently, prioritizing a considerably shorter route becomes the agent's focus to minimize the total cost, as shown in Fig. 6.

4.3 Scenario 3: Utilitarian Government, Company, and Protesters

In the final scenario, we assume that all stakeholders prioritize social well-being. The protesters attempted to block all the streets across areas 1 to 6. Contrarily, government law enforcement refrains from dispersing them entirely but opts to clear several main streets in areas 3 to 6, engaging in persuasive dialogue to encourage the protesters to disperse voluntarily. Recognizing the situation's complexity, the company requests the agent to avoid areas 1 and 2 and exercise caution when navigating through areas 3 to 6. Consequently, this scenario involves both 'hard' and 'soft' ethical constraints. The LTL formula remains consistent with the expression presented in Eq. (10). Permissions with strong and weak discouragement are implemented, introducing a negative reward of -0.3 when the agent traverses areas 3 to 5, and -0.01 for area 6.

After acquiring the optimal policy, the path generated for the agent to accomplish the task is illustrated in Fig. 7. As area 6 is permitted with weak encouragement, the agent opts to navigate through it to minimize the total cost, even though the path is longer than traversing areas 4 and 5.

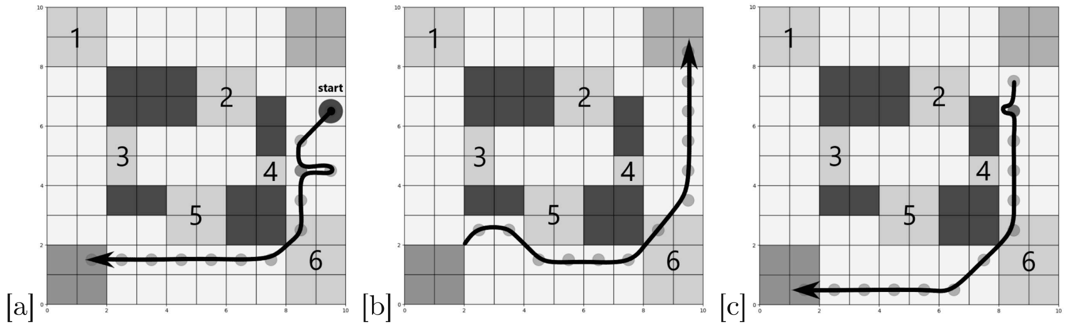

Fig. 7. A path to accomplish task ϕ_1 in case 3.

5 Conclusion

We propose a model-free RL approach to tackle motion planning challenges in partially observable environments while considering ethical constraints. Our framework classifies ethical norms into 'hard' and 'soft' constraints. Complex tasks and 'hard' ethical constraints are expressed using LTL, while an additional reward function enforces 'soft' ethical constraints. To address the defined problems, we employ an RNN-based DQN. The Q networks use the observation history and label sequences as inputs to estimate Q values, enabling the agent to make optimal decisions. We conduct a simulation example to showcase the effectiveness and flexibility of our proposed approach. Future research directions include handling dynamic ethical constraints and exploring multi-objective RL approaches.

Acknowledgments. This study was funded by US Department of Education (ED #P116S210005) and NSF (#2226936).

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Mill, J.-S.: Utilitarianism. Crips, Roger (ed.). Oxford University Press, Oxford, England (1998)
2. Davis, N.-A.: Contemporary Deontology. Blackwell, Malden, Massachusetts, United States (1991)
3. Crisp, R., Slote, M.: Virtue Ethics. Oxford University Press, Oxford, England (1997)
4. Sinnott-Armstrong, W.: Consequentialism. Stanford Encyclopedia of Philosophy (2019)
5. Slavkovik, M.: Automating moral reasoning. In: Bourgaux, C., Ozaki, A., Penaloza, R. (eds.) International Research School in Artificial Intelligence in Bergen, Open Access Series in Informatics (OASIcs), vol. 99, pp. 6:1 – 6:13. University of Bergen, Norway (2022)

6. Chadès, I., Pascal, L.-V., Nicol, S., Fletcher, C.-S., Ferrer-Mestres, J.: A primer on partially observable Markov decision processes (POMDPs). Methods Ecol. Evol. **12**, 2058–2072 (2021). https://doi.org/10.1111/2041-210X.13692
7. Baier, C., Katoen, J.-P.: Principles of Model Checking, 1st edn. MIT press, Cambridge, Massachusetts (2008)
8. Bozkurt, A.-K., Wang, Y., Zavlanos, M.-M., Pajic, M.: Control synthesis from linear temporal logic specifications using model-free reinforcement learning. In: Proceedings - IEEE International Conference on Robotics and Automation, pp. 10349–10355. IEEE, Paris, France (2020)
9. Cai, M., Hasanbeig, M., Xiao, S., Abate, A., Kan, Z.: Modular deep reinforcement learning for continuous motion planning with temporal logic. IEEE Robot. Autom. Lett. **6**(4), 7973–7980 (2021). https://doi.org/10.1109/LRA.2021.3101544
10. Cai, M., Xiao, S., Li, B., Li, Z., Kan, Z.: Reinforcement learning based temporal logic control with maximum probabilistic satisfaction. In: Proceedings - IEEE International Conference on Robotics and Automation, pp. 806–812, IEEE, Xi'an, China (2021). https://doi.org/10.1109/ICRA48506.2021.9561903
11. Cai, M., Xiao, S., Li, Z., Kan, Z.: Optimal probabilistic motion planning with potential infeasible LTL constraints. IEEE Trans. Autom. Control **68**(1), 301–316 (2023). https://doi.org/10.1109/TAC.2021.3138704
12. Cai, M., Xiao, S., Li, J., Kan, Z.: Safe reinforcement learning under temporal logic with reward design and quantum action selection. Sci. Rep. **13**, 1925 (2023). https://doi.org/10.1038/s41598-023-28582-4
13. Watkins, C., Dayan, P.: Q-Learning. Mach. Learn. **3–4**, 279–292 (1992). https://doi.org/10.1007/bf00992698
14. Mnih, V., et al.: Human-level control through deep reinforcement learning. Nature **7540**, 14764687 (2015). https://doi.org/10.1038/nature14236
15. Hausknecht, M., Stone, P.: Deep recurrent q-learning for partially observable MDPs. In: Technical Report - AAAI Fall Symposium, (2015)
16. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 08997667 (1997). https://doi.org/10.1162/neco.1997.9.8.1735