1 System Model

Definition 1.1. We represent a probabilistic discrete event system (DES) as a labeled Markov decision process (MDP). A DES is a tuple $D = (S, E, P_T, P_E, s_{init}, AP, L)$, where S is a finite set of states; E is a finite set of events; $P_T : S \times S \times E \to [0,1]$ is a transition probability; $P_E : E \times S \times 2^E \to [0,1]$ is the probability of an event occurrence under a state $s \in S$ and a subset $\pi \in \mathcal{E}(s)$; for any $(s', s, \pi) \in S \times S \times 2^E$, we define the probability $P : S \times S \times 2^E \to [0,1]$ such that $P(s'|s,\pi) = \sum_{e \in \pi} P_E(e|s,\pi) P_T(s'|s,e)$ and $\Sigma_{s' \in S} P(s'|s,\pi) = 1$; $s_{init} \in S$ is the initial state; AP is a finite set of atomic propositions; and $L : S \times E \times S \to 2^{AP}$ is a labeling function that assigns a set of atomic propositions to each transition $(s, e, s') \in S \times E \times S$. Let $\mathcal{E}(s) = \{e \in E; \text{the event } e \text{ is feasible at the state } s\}$. Note that $\sum_{s' \in S} P_T(s'|s,e) = 1$ holds for any state $s \in S$ and any event $e \in \mathcal{E}(s)$, $P_T(s'|s,e) = 0$ for any $e \notin \mathcal{E}(s)$, and $\Sigma_{e \in \pi} P_E(e|s,\pi) = 1$ and we call the subset the control pattern. We assume that E can be partitioned into the set of controllable events E_c and the set of uncontrollable events E_{uc} such that $E_c \cup E_{uc} = E$ and $E_c \cap E_{uc} = \emptyset$. Note that each event e occurs probabilistically depending on only the current state and the subset of feasible events at the state given by a controller.

In the DES D, an infinite path starting from a state $s_0 \in S$ is defined as a sequence $\rho = s_0 \pi_0 e_0 s_1 \ldots \in S(2^E E S)^\omega$ such that $P_T(e_i|s_i,\pi_i) > 0$ and $P_T(s_{i+1}|s_i,e_i) > 0$ for any $i \in \mathbb{N}_0$. A finite path is a finite sequence in $S(2^E E S)^*$. In addition, we sometimes represent ρ as ρ_{init} to emphasize that ρ starts from $s_0 = s_{init}$. For a path $\rho = s_0 \pi_0 e_0 s_1 \ldots$, we define the corresponding labeled path $L(\rho) = L(s_0, e_0, s_1) L(s_1, e_1, s_2) \ldots \in (2^{AP})^\omega$. $InfPath^D$ (resp., $FinPath^D$) is defined as the set of infinite (resp., finite) paths starting from $s_0 = s_{init}$ in the DES D. For each finite path ρ, $last(\rho)$ denotes its last state.

We define the supervisor as a controller for the DES that restricts the behaviors of the DES to satisfy a given specification.

Definition 1.2. For the DES D, a supervisor $SV : FinPath^D \to 2^E$ is defined as a mapping that maps each finite path to a set of enabled events at the finite path and we call the set a control pattern. A supervisor is positional if $SV(\rho) = SV(last(\rho))$ for any $\rho \in InfPath^D$. Note that the relation $E_{uc} \subset SV(\rho) \subset E$ holds for any $\rho \in FinPath^M$.

Definition 1.3. A state $\bar{x}$ in the state set $\bar{X}$ of an augmented tLDBA $\bar{B}_\varphi = (\bar{X}, \bar{x}_{init}, \bar{\Sigma}, \bar{\delta}, \bar{\mathcal{F}})$ is a sink state if no accepting transition is defined for all states reachable from $\bar{x}$. We denote the set of sink states as $SinkSet$.

Definition 1.4. Given an augmented tLDBA $\bar{B}_\varphi = (\bar{X}, \bar{x}_{init}, \bar{\Sigma}, \bar{\delta}, \bar{\mathcal{F}})$ and a DES D, a tuple $D \otimes \bar{B}_\varphi = D^\otimes = (S^\otimes, E^\otimes, s^\otimes_{init}, P^\otimes_T, P^\otimes_E, \delta^\otimes, \mathcal{F}^\otimes)$ is a product DES, where $S^\otimes = S \times \bar{X}$ is the finite set of states and we represent s and $\bar{x}$ corresponding with $s^\otimes = (s, \bar{x}) \in S^\otimes$ as $[\![s^\otimes]\!]_s$ and $[\![s^\otimes]\!]_q$, respectively; $E^\otimes = E \cup \{\varepsilon_{x'}; \exists x' \text{s.t.} (x, \varepsilon_{x'}, x') \in \delta\}$ is the finite set of events, where $\varepsilon_{x'}$ is the action that represents an ε-transition to $x' \in X$; $s^{\otimes}_{init} = (s_{init}, \bar{x}_{init})$ is the initial states, $P^{\otimes}_{T} : S^{\otimes} \times S^{\otimes} \times E^{\otimes} \to [0,1]$ is the transition probability defined as

$$P^{\otimes}_{T}(s^{\otimes\prime} | s^{\otimes}, e) = \begin{cases} P_T(s'|s,e) & \text{if } (\bar{x}, L((s,e,s')), \bar{x}') \in \bar{\delta}, a \in \mathcal{A}(s) \\ 1 & \text{if } s = s', v = v', (x, \varepsilon_{x'}, x') \in \delta, a = \varepsilon_{x'} \\ 0 & \text{otherwise,} \end{cases}$$

where $s^{\otimes} = (s,(x,v))$ and $s^{\otimes\prime} = (s',(x',v'))$. $P^{\otimes}_{E} : E^{\otimes} \times S^{\otimes} \times 2^{E^{\otimes}} \to [0,1]$ is the probability of the occurrence of the event defined as $P^{\otimes}_{E}(e|s^{\otimes}, \pi) = P_E(e|s,\pi)$, $\delta^{\otimes} = \{(s^{\otimes}, e, s^{\otimes\prime}) \in S^{\otimes} \times E^{\otimes} \times S^{\otimes}; P^{\otimes}_{T}(s^{\otimes\prime}|s^{\otimes}, e) > 0\}$ is the set of transitions, and $\mathcal{F}^{\otimes} = \{\bar{F}^{\otimes}_1, \ldots, \bar{F}^{\otimes}_n\}$ is the acceptance condition, where $\bar{F}^{\otimes}_i = \{((s,\bar{x}), e, (s', \bar{x}')) \in \delta^{\otimes}\ ;\ (\bar{x}, L(s,e,s'), \bar{x}') \in \bar{F}_i\}$ for each $i \in \{1, \ldots, n\}$.

2 Objective function for Control patterns

From the view point of reinforcement learning, the DES can be interpreted as the environment controlled by the supervisor and the supervisor can be interpreted as the agent. We introduce the two following assumptions.

1. The relative frequency of occurrence of each event does not depend on the control pattern.
2. We define a reward function $\mathcal{R} : S \times 2^E \times E \times S \to \mathbb{R}$ and the reward $\mathcal{R}$ can be decomposed into $\mathcal{R}_1$ and $\mathcal{R}_2$. The first reward $\mathcal{R}_1 : S \times 2^E \to \mathbb{R}$ is determined by the control pattern selected by the supervisor, which depends on only the control pattern and the current state. The second reward $\mathcal{R}_2 : S \times E \times S \to \mathbb{R}$ is determined by the occurrence of an event and the corresponding state transition. For any $(s, \pi, e, s') \in S \times 2^E \times E \times S$, we then have

$$\mathcal{R}(s, \pi, e, s') = \mathcal{R}_1(s, \pi) + \mathcal{R}_2(s, e, s'). \tag{1}$$

Under the above assumptions, we have the following *Bellman optimality equation.*

$$\begin{aligned} Q^*(s,\pi) = & \sum_{s' \in S} P(s'|s,\pi) \\ & \left\{ \mathcal{R}(s,\pi,e,s') + \gamma \max_{\pi' \in 2^{\mathcal{E}(s')}} Q^*(s',\pi') \right\} \\ = & \sum_{s' \in S} \sum_{e \in \pi} P_E e|s,\pi) P_T(s'|s,e) \left\{ \mathcal{R}_1(s,\pi) + \mathcal{R}_2(s,e,s') + \gamma \max_{\pi' \in 2^{\mathcal{E}(s')}} Q^*(s',\pi') \right\} \\ = & \mathcal{R}_1(s,\pi) + \sum_{e \in \pi} P_E(e|s,\pi) \sum_{s' \in S} P_T(s'|s,e) \left\{ \mathcal{R}_2(s'|s,e) + \gamma \max_{\pi' \in 2^{\mathcal{E}(s')}} Q^*(s',\pi') \right\}, \end{aligned} \tag{2}$$

where $\gamma \in [0,1)$.

We introduce the following function. $T^* : S \times E \to \mathbb{R}$ such that

$$T^*(s,e) = \sum_{s' \in S} P_T(s'|s,e) \left\{ \mathcal{R}_2(s'|s,e) + \gamma \max_{\pi' \in 2^{\mathcal{E}(s')}} Q^*(s',\pi') \right\}. \tag{3}$$

We then have

$$Q^*(s,\pi) = \mathcal{R}_1(s,\pi) + \sum_{e \in \pi} P_E(e|s,\pi) T^*(s,e). \tag{4}$$

Definition 2.1. We define an optimal supervisor SV^* as follows. For any state $s \in S$,

$$SV^*(s) = \pi^* \in \arg\max_{\pi \in \mathcal{E}(s)} Q^*(s,\pi). \tag{5}$$

Definition 2.2. The two reward functions $\mathcal{R}_1 : S^{\otimes} \times 2^{E^{\otimes}} \to \mathbb{R}$ and $\mathcal{R}_2 : S^{\otimes} \times E^{\otimes} \times S^{\otimes} \to \mathbb{R}$ are defined as follows.

$$\mathcal{R}_1(s^{\otimes},\pi) = \begin{cases} r_n|\pi| & \text{if } [\![s^{\otimes}]\!]_q \notin \mathit{SinkSet}, \\ 0 & \text{otherwise,} \end{cases} \tag{6}$$

where $|E|$ means the number of elements in the set E and r_n is a positive value.

$$\mathcal{R}_2(s^{\otimes},e,s^{\otimes\prime}) = \begin{cases} r_p & \text{if } \exists j \in \{1,\ldots,n\},\ (s^{\otimes},e,s^{\otimes\prime}) \in \bar{F}_j^{\otimes}, \\ r_{sink} & \text{if } [\![s^{\otimes\prime}]\!]_q \in \mathit{SinkSet}, \\ 0 & \text{otherwise,} \end{cases} \tag{7}$$

where r_p and r_{sink} are the positive and negative value, respectively.

3 Learning Algorithm

We make the supervisor learn how to give the control patterns to satisfy an LTL specification while keeping costs associated with disabled events low. We use Q-learning to estimate the function T^*. We then use Bayesian inference to robustly estimate the probability P_E. For the inference, we model P_E as Categorical distribution as $p^k_{s,\pi,e}$, where $p^k_{s,\pi,e}$ represents the estimated probability of $P_E(e|s,\pi)$ at the time step k and the prior distribution $\phi^k_{s,\pi}$ for the distribution of the parameter of $p^k_{s,\pi,e}$ is defined as Dirichlet.

In the following, we distinguish events by numbering them as $\{e^1,\ldots,e^{|E|}\}$. In order to reflect the events disabled by the supervisor on the estimated probability of an event occurrence, we introduce the function $\mathit{RestProb} : (0,1)^{|E|} \times 2^E \to [0,1]^{|E|}$ defined as

$$RestProb(\phi^k_{s,\pi},\pi)_i = \begin{cases} \dfrac{\phi^{k,i}_{s,\pi}}{\sum_{e^j \in \pi} \phi^j_{s,\pi}} & \text{if } e^i \in \pi, \\ 0 & \text{otherwise}, \end{cases} \tag{8}$$

where $\phi^{k,i}_{s,\pi}$ is the i-th element of $\phi^k_{s,\pi}$ and $RestProb(\phi^k_{s,\pi},\pi)_i$ is the i-th element of $RestProb(\phi^k_{s,\pi},\pi)$.

We denote the probability vector of an event occurrence at the time step k as $p^k_{s,\pi} = (p^k_{s,\pi,e^1}, \ldots, p^k_{s,\pi,e^{|E|}})$, where $s \in S$ and $\pi \in \mathcal{E}(s)$ is the state and the control pattern at the time step k. Let $n^k_{s,\pi,e}$ be the number of the occurrence of the event $e \in E$ up to the time step k at the state $s \in S$ under the control pattern $\pi \in \mathcal{E}(s)$ and let $n^k_{s,\pi} = (n^k_{s,\pi,e_1}, \ldots, n^k_{s,\pi,e_{|E|}})$. We sample the parameter $\phi^k_{s,\pi}$ of the posterior distribution of an event occurrence from the Dirichlet distribution $Dir(\cdot|n^k_{s,\pi})$. Then, we obtain the estimated probability vector $p^k_{s,\pi}$ of an event occurrence by $RestProb$ from the sampled parameter $\phi^k_{s,\pi}$ and the control pattern π.

The overall procedure of the inference is shown in Algorithm 1.

Algorithm 1 P_E inference.

```
Input: the event occurrence count $n^k_{s,\pi}$, a threshold $\xi^k_{s,\pi}$ for $p^k_{s,\pi}$
Output: the posterior distribution $p^k_{s,\pi}$
$\phi^k_{s,\pi} \sim Dir(\cdot|n^k_{s,\pi})$
$p^k_{s,\pi} = RestProb(\phi^k_{s,\pi}, \pi)$
```

Under the estimation of P_E, we use TD-learning to estimate Q^* with the TD-error defined as $\mathcal{R}_1(s^\otimes,\pi) + \sum_{e\in\pi} p_{[\![s^\otimes]\!]_s,\pi,e} T(s^\otimes, e) - Q(s^\otimes,\pi)$.

We show the overall procedure of the learning algorithm in Algorithm 2.

Algorithm 2 RL-based synthesis of a supervisor satisfying the given LTL specification.

```
Input: LTL formula $\varphi$, DES $M$
Output: optimal supervisor $SV^*$ on the product DES $M^{\otimes}$
Convert $\varphi$ into tLDGBA $B_{\varphi}$.
Augment $B_{\varphi}$ to $\bar{B}_{\varphi}$.
Construct the product DES $M^{\otimes}$ of $M$ and $\bar{B}_{\varphi}$.
Initialize $T : S^{\otimes} \times E^{\otimes} \to \mathbb{R}$.
Initialize $Q : S^{\otimes} \times 2^{E^{\otimes}} \to \mathbb{R}$.
Initialize $n : S \times 2^{E} \times E \to \mathbb{R}$.
initialize $\xi : S \times 2^{E} \to \mathbb{R}$.
Initialize episode length $L$.
while $Q$ is not converged do
    $s^{\otimes} \leftarrow (s_{init}, (x_{init}, \mathbf{0}))$.
    $t \leftarrow 0$
    while $t < L$ and $[\![s^{\otimes}]\!]_q \notin SinkSet$ do
        Choose the control pattern $\pi \in 2^{\mathcal{E}(s^{\otimes})}$ by the supervisor $SV$.
        Observe the occurrence of the event $e \in E$.
        Observe the next state $s^{\otimes\prime}$.
        $T(s^{\otimes}, e) \leftarrow (1 - \alpha)T(s^{\otimes}, e) + \alpha\{\mathcal{R}_2(s^{\otimes}, e, s^{\otimes\prime}) + \gamma \max_{\pi' \in 2^{\mathcal{E}(s^{\otimes\prime})}} Q(s^{\otimes\prime}, \pi')\}$
        $n([\![s^{\otimes}]\!]_s, \pi, e) \leftarrow n([\![s^{\otimes}]\!]_s, \pi, e) + 1$
        Obtain $p_{[\![s^{\otimes}]\!]_s, \pi}$ from $n$ by the $P_E$ inference.
        $Q(s^{\otimes}, \pi) = (1 - \beta)Q(s^{\otimes}, \pi) + \beta\{\mathcal{R}_1(s^{\otimes}, \pi) + \sum_{e \in \pi} p_{[\![s^{\otimes}]\!]_s, \pi, e} T(s^{\otimes}, e)\}$
        $s^{\otimes} \leftarrow s^{\otimes\prime}$
        $t \leftarrow t + 1$
    end while
end while
```

4 Example

We evaluate the algorithm by the maze of the cat and the mouse shown in Fig. 1. At the beginning, we define the settings for the example. The corresponding DES is as follows. The state set is $S = \{(s^{cat}, s^{mouse}); s^{cat}, s^{mouse} \in \{s_0, s_1, s_2, s_3\}\}$. The set of events (to open the corresponding door) is $E = \{m_0, m_1, m_2, m_3, c_0, c_1, c_2, c_3\}$, where $E_c = \{m_0, m_1, m_2, m_3, c_0, c_1, c_2\}$ and $E_{uc} = \{c_3\}$ and $\mathcal{E}(s) = E$ for any $s \in S$. The initial state is $s_{init} = (s_0, s_2)$. If the door of the room with the cat (resp., mouse) opens, the cat (resp., mouse) moves, with probability 0.95, to the room next to the room through the door or stays in the same room with probability 0.05. Otherwise, the cat (resp., mouse) stays in the same room with probability 1. The labeling function is

$$L((s,a,s')) = \begin{cases} \{a\} & \text{if } s'_c = s_1, \\ \{b\} & \text{if } s'_m = s_1, \\ \{c\} & \text{if } s'_c = s'_m, \\ \emptyset & \text{otherwise,} \end{cases}$$

where s'_c and s'_m is the next room where the cat and the mouse is, respectively, i.e., $s' = (s'_c, s'_m)$.

In the example, we want the supervisor to learn to give control patterns satisfying that the cat and the mouse take the food in the room 1 (s_1) avoiding they come across. This is formally specified by the following LTL formula.

$$\varphi = \mathbf{GF}a \wedge \mathbf{GF}b \wedge \mathbf{G}\neg c.$$

The tLDGBA $B_\varphi = (X, x_{init}, \Sigma, \delta, \mathcal{F})$ corresponding to φ is shown in Fig. 2. B_φ has the acceptance condition of two accepting sets.

We use ε-greedy policy and gradually reduce ε to 0 to learn an optimal supervisor asymptotically. We set the rewards $r_p = 10$, $r_n = 0.1, 0.7, and 1.2$, and $r_{sink} = -1000$; the epsilon greedy parameter $\varepsilon = \frac{1}{\sqrt{episode}}$, where *episode* is the number of the current episode; and the discount factor $\gamma = 0.99$. The learning rate α and β vary in accordance with *the Robbins-Monro condition.* We train supervisors 5000 iterations and 15000 episodes.

Fig. 3 shows the estimated optimal state values at the initial state $V(s^{\otimes}_{init})$ with $r_n = 0.1, 0.7$, and 1.2, respectively, for each episode when learning 5000 iterations and 15000 episodes by the algorithm 2. Fig. 4 shows the average rewards from $\mathcal{R}_2$ and the average rewards from $\mathcal{R}_1$ with $r_n = 0.1, 0.7$, and 1.2, respectively, of 5000 iterations and 1000 episodes by the supervisor obtained from the learning.

Fig. 3 suggests the three supervisors becomes optimal as the episode progresses. Fig. 4 suggests the three supervisors obtained from the learning satisfy φ and there is no sink recurrent class under the supervisors. The latter is implied by the stable average rewards. Furthermore, Fig. 4 suggests that there is a trade-off between the frequency of visits to accepting sets of a augmented tLDGBA corresponding to a given LTL formula and the number of enabling events. Moreover, we can consider how much we see it important to enable events and how often the event occurs that leads to the satisfaction of a given LTL formula by changing the magnitude of the reward for control patterns and the reward for the LTL formula relatively.

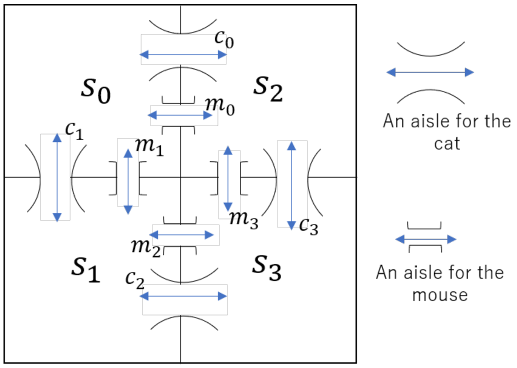

Figure 1: The maze of the cat and the mouse. the initial state of the cat and the mouse is s_0 and s_2, respectively. the food for them is in the room 1 (s_1).

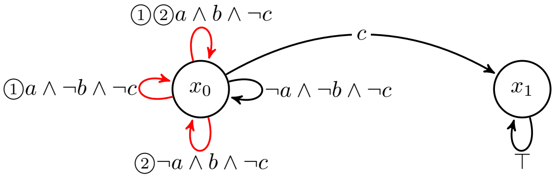
①②$a \wedge b \wedge \neg c$
c
①$a \wedge \neg b \wedge \neg c$
x_0
$\neg a \wedge \neg b \wedge \neg c$
x_1
②$\neg a \wedge b \wedge \neg c$
$\top$

Figure 2: The tLDGBA recognizing the LTL formula $\mathbf{GF}a \wedge \mathbf{GF}b \wedge \mathbf{G}\neg c$, where the initial state is x_0. Red arcs are accepting transitions that are numbered in accordance with the accepting sets they belong to, e.g., ①$a \wedge \neg b \wedge \neg c$ means the transition labeled by it belongs to the accepting set F_1.

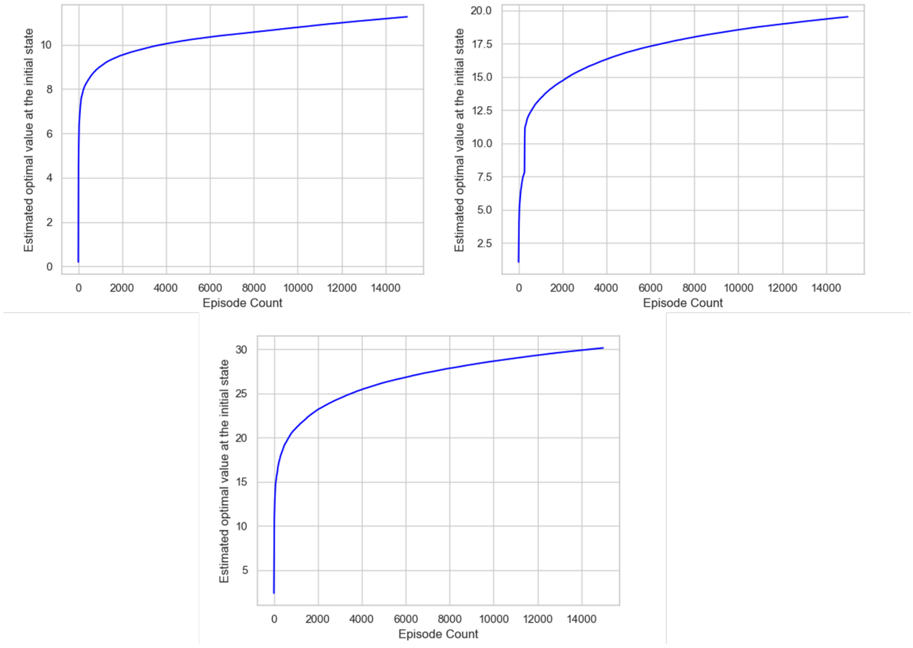

Figure 3: The estimated optimal state values at the initial state $V(s_{init}^{\otimes})$ with $r_n = 0.1$ (left above), $r_n = 0.7$ (right above), and $r_n = 1.2$ (below) when using Algorithm 2.

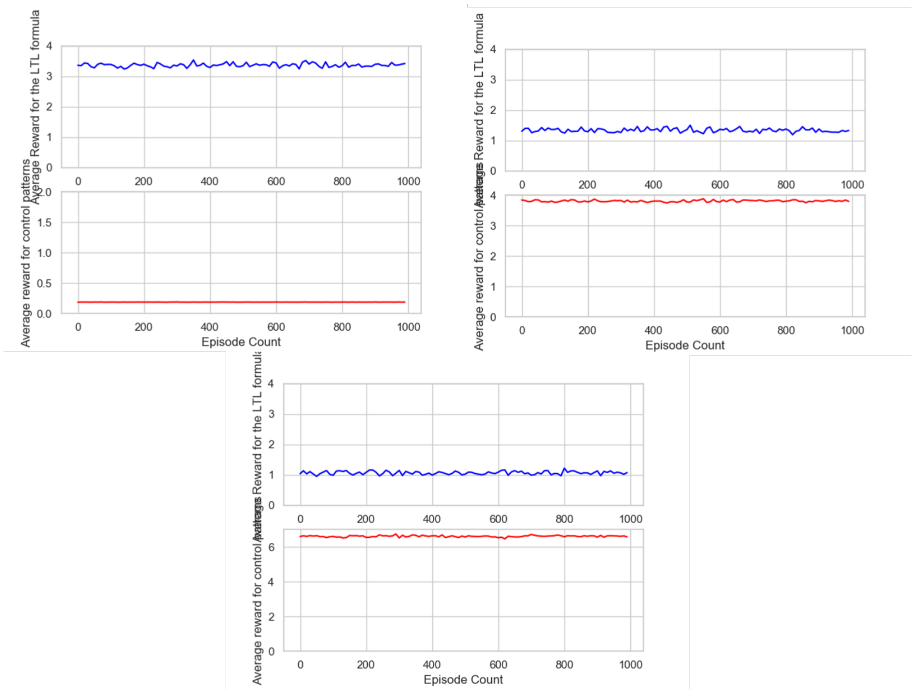

Figure 4: The average rewards of $\mathcal{R}_1$ and average rewards of $\mathcal{R}_2$ by the supervisor obtained from the learning with $r_n = 0.1$ (left above), $r_n = 0.7$ (right above), and $r_n = 1.2$ (below).