A Mutual Information-based Assessment of Reverse Engineering on Rewards of Reinforcement Learning

Tong Chen, Jiqiang Liu, Thar Baker, Yalun Wu, Yingxiao Xiang, Yike Li,
Wenjia Niu*, Endong Tong*, Albert Zomaya

***Abstract*—Rewards are critical hyperparameters in reinforcement learning (RL), since in most cases different reward values will lead to greatly different performance. Due to their commercial value, RL rewards become the target of reverse engineering by the inverse reinforcement learning (IRL) algorithm family. Existing efforts typically utilize two metrics to measure the IRL performance: the expected value difference and the mean reward loss, which we call them EVD and MRL respectively. Unfortunately, in some cases, EVD and MRL can give completely opposite results, due to MRL focusing on whole state-space rewards while EVD only considering partly sampled rewards. Such situation naturally rises to one fundamental question: whether current metrics and assessment are sufficient and accurate for more general use. Thus, in this paper, based on the metric called normalized mutual information of reward clusters (C-NMI) we propose a novel IRL assessment; we aim to fill this research gap by considering a middle-granularity state space between the entire state space and the specific sampling space. We utilize the agglomerative nesting algorithm (AGNES) to control dynamical C-NMI computing via a 4-order tensor model with injected manipulated trajectories. With such a model, we can uniformly capture different-dimension values of MRL, EVD, and C-NMI, and perform more comprehensive and accurate assessment and analyses. Extensive experiments on several mainstream IRLs are experimented in Object World, hence revealing that the assessing accuracy of our method increases 110.13% and 116.59% respectively when compared with the EVD and MRL. Meanwhile, C-NMI is more robust than EVD and MRL under different demonstrations.**

***Impact Statement*—In this work, we pay attention to the inconformity problem of MRL- and EVD-based IRL assessment. There are two main challenges for us to address: (1) how to design a novel metric by combining the advantages of both MRL and EVD, and (2) how to construct a comprehensive assessment method for accurate comparison and analysis. To address such challenges, we craft a novel assessment of IRL based on the metric called normalized mutual information of reward clusters (C-NMI). Hence we attempt to fill the existing research gap by considering a middle-granularity state space between the entire state space and the certain sampled space. We list all of the notation and parameters used in the rest of this paper in Table I.**

TABLE I: Parameters and Notation

Notation	Meaning
π	policy, π^* optimal policy, π^u learned policy, π^{sub}/π^{inv} suboptimal/inverted policy
$\boldsymbol{r}$	reward set, $\boldsymbol{r}^e$ ground truth reward set, $\boldsymbol{r}^u$ learned reward set, $\boldsymbol{r}^{inv}$ inverted reward set
$\mathcal{D}$	demonstration, $\mathcal{D}^*$ expert demonstration, $\mathcal{D}^{s*}/\mathcal{D}^{i*}$ suboptimal/inverted manipulated demonstration
τ	trajectory, τ^* optimal trajectory, τ^{sub}/τ^{inv} suboptimal/inverted trajectory
$\mathcal{C}$	reward cluster, $\boldsymbol{\mathcal{C}} = \{\mathcal{C}_1, \mathcal{C}_2, \dots\}$ reward clusters set
S'	$S' = \{s' \mid r^e_{s'} \in \mathcal{C}^e_i\}$, $\mathcal{C}^e_i$ denotes any subset in $\mathcal{C}^e$, the sampled state space
k	cluster number, $\lvert\boldsymbol{k}\rvert = X$
o	top $\mathcal{K}$ of cluster ranking, $\lvert\boldsymbol{o}\rvert = Y$
m	percentage of manipulated trajectories, $\lvert\boldsymbol{m}\rvert = Z$
g	metrics, $\lvert\boldsymbol{g}\rvert = V$
p	percentage of manipulated trajectories injected
$\boldsymbol{B}$	4-order tensor, $\boldsymbol{B} \in \mathbb{R}^{X \times Y \times Z \times V}$
Q^v_1, Q^v_3	upper- and lower-quartiles of $\boldsymbol{B}_{i,j,z,v}$
Col	Col_1 the number of primary color, Col_2 the number of secondary color
DS	experimental datasets
A	IRL algorithm
ϵ	assessing accuracy of metric, $\overline{\epsilon}$ average assessing accuracy

***Index Terms*—reinforcement learning, reverse engineering, mutual information, assessment, tensor model**

I. Introduction

REINFORCEMENT learning (RL) together with unsupervised- and supervised-learning constitute the complete framework of machine learning in the AI field. Compared with the expert supervision that utilized in supervised learning, RL relies on step-by-step reward feedback from the interaction between the autonomous agent and the environment. RL algorithm can learn the corresponding optimal policy from a large number of trajectories via a series of action attempts of the agent during training time. Because of this unique characteristics, RL can be regraded as autonomous learning, and greatly drives AI development to grounding applications, including the Boston Dynamics robots [28] and AlphaGo Zero [27] software agent.

In RL, different rewards usually lead to performance's significant differences. As the critical hyperparameter in RL, the reward refers to the feedback given by the corresponding environment according to the agent's action selection performance at each state. Reward can guide the agent to learn a optimal policy which maximize the sum of expected rewards. Utilizing a simple car driving for example, we may simply design two rewards: +1 means taking an action into a state that has a predefined safe distance to the car in front of our car,

The work was supported by the National Natural Science Foundation of China [61972025, 61802389, 61672092, U1811264, and 61966009]; and the National Key R&D Program of China [2020YFB1005604 and 2020YFB2103802].

T. Chen, J. Liu, Y. Wu, Y. Xiang, Y. Li, W. Niu, and E. Tong are with the Beijing Key Laboratory of Security and Privacy in Intelligent Transportation, Beijing Jiaotong University, 3 Shangyuan Village, Haidian District, Beijing, China (e-mail: <tongchen, jqliu, wuyalun, yxxiang, yikeli, niuwj, edtong>@bjtu.edu.cn)

T. Baker is now with the University of Sharjah, Sharjah, United Arab Emirates (e-mail: tshamsa@sharjah.ac.ae)

A. Zomaya is now with the University of Sydney, School of Computer Science, Sydney, Australia (e-mail: albert.zomaya@sydney.edu.au)

This paragraph will include the Associate Editor who handled your paper.

while -1 means taking an action into a state not having a safe distance. However, artificial rewards for the given task do not work well in many cases, such as drone autonomous driving and Unmanned aerial vehicle. From this, we can see that well designed rewards have high commercial value, and need to be carefully predesigned according to the expert knowledge.

Unfortunately, RL rewards have become the target of reverse engineering through inverse reinforcement learning (IRL) [37]. IRL algorithms aim to learn rewards by imitating history expert trajectories, while IRL is designed to make the manually design of rewards become easier. More specifically, given a certain number of expert trajectories, IRL can obtain approximate rewards via model learning. Such a situation makes IRL become a double-edged sword, meaning that instead of providing help for RL training, IRL can also be used as a kind of reverse engineering to achieve approximated rewards. In recent years, with the development trend of online open AI, the reverse engineering vulnerability of IRL has become a serious challenge. For example, Facebook [26] has opened up a large number of expert trajectories of several RL models on Github to encourage researchers to retrain, and eventually upload their new models back to Facebook. Hence, it is necessary to explore accurate assessments on IRL performance, and further expose the threat of reward reverse engineering.

IRL methods are mainly classified into two categories: linear methods, including MMP [55], MWAL [56], MaxEnt [54], and AN [40]; and non-linear kernel function-based ones, including GPIRL [53], LEARCH [39], and FIRL [41]. Under the scenario of IRL performance assessing, there are generally two assessment metrics: expected value difference and mean reward loss, which we call them EVD and MRL respectively. EVD pay attention to the expected value of several sampling trajectories, while MRL focuses the entire state space and get the mean reward loss. However, as the motivating example shows in Section. III, EVD and MRL can give completely opposite assessing results. For this reason, which assessment metric should be trusted? This is the fundamental point that inspires us to rethink whether the current metrics are sufficient for accurate evaluation. Thus, it is necessary to develop a novel metric that can fill the gap with middle granularity between the entire state space and the specific sampling space.

In this work, we propose a novel assessing metric of reverse engineering on RL rewards, and develop a metric via normalized mutual information of reward clusters (C-NMI). We employ an agglomerative nesting algorithm (AGNES) [52] for dynamical C-NMI computing to quantify the reward clusters' similarity compared with the reward ground truth. We build a 4-order tensor model embedded with manipulated trajectories, which are formed from both suboptimal [43] and inverted [42] trajectories. Based on such a 4-order tensor model, we can uniformly capture and store different-dimension metric values of MRL, EVD, and C-NMI. We can also perform comprehensive assessment by computing a lower 1.5 $\times$ interquartile range (IQR) [33] whisker for MRL and EVD, and an upper 1.5 $\times$ IQR whisker for C-NMI.

In our experiment, we target Object World (OW) as the benchmark, and implement 7 mainstream IRL algorithms: MaxEnt, MMP, GPIRL, MWAL, FIRL, AN, and LEARCH. For setting the ground truth, the consistency between MRL and EVD is analyzed. In our datasets, statistical analysis shows that C-NMI has better assessing accuracy than MRL and EVD. In detail, C-NMI-based assessment can achieve the highest accuracy of 0.89, and is robust under different cases as well. We summarize our contributions as follows:

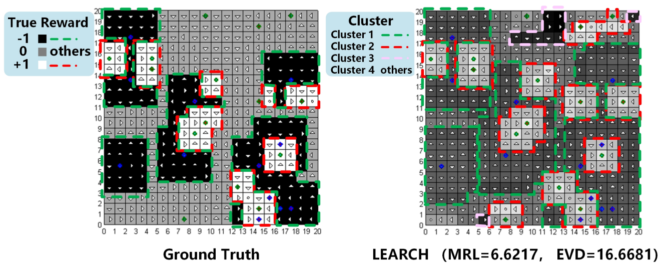
1) We make the first attempt to craft a novel assessment of IRL based on the metric called normalized mutual information of reward clusters (C-NMI), which can fill the existing research gap by considering a middle-granularity state space between the entire state space and the certain sampled space.
2) We first construct a 4-order tensor model with injected manipulate trajectories to realize the dynamically calculating of C-NMI.
3) We give extensive experiments on seven mainstream IRLs, which shows that C-NMI increases 110.13% and 116.59% respectively when compared with the EVD and MRL. Meanwhile, C-NMI is more robust than EVD and MRL under different demonstrations.

The rest of this paper is organized as follows. The related work is organized in Section. II. Section. III describes the motivating examples. Section. IV demonstrates the preliminaries. Section. V proposes a mutual information-based assessment of reverse engineering on RL rewards. Section. VI reports experiments and the corresponding detailed analysis. Finally, Section. VII gives the conclusion.

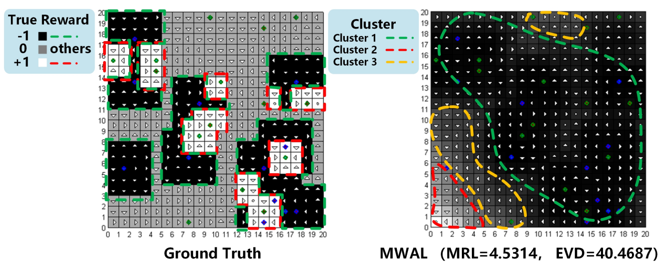
II. Related work

Critical Hyperparameter stealing Existing works have revealed hyperparameter stealing attack and more directly model stealing attack. Against open application programming interfaces (including Amazon machine learning and BigML [20]), researchers found such attacks that can steal the machine learning model almost perfectly. While our work make the first attempt to assess the reverse engineering threats against the open RL platform. Wang et al. [29] focused on revealing hyperparameter stealing, in which the stolen hyperparameter is highly important for model performance. Similarly, for RL model, reward function is the critical hyperparameter, which should be designed before model begins. However, comparing with the hyperparameters within the traditional machine learning model, the number of rewards are unknown, which is a challenge for IRL reverse engineering. Thus, we both pay attention to the unknown hyperparameters' number, and the specific values of reward function in RL model.

IRL assessment Many studies [53] [40] have shown different measures with various names on IRL performance, including the EVD, MRL, approximation error, percent misprediction, feature expected distance, policy loss, and learning score. In earlier works, the approximation error is usually utilized to assess the similarity between the original reward function and the reward function approximated by IRL. Currently, EVD and MRL are used to evaluate the reward function's similarity. In our work, we also employ them as a highly important measures. Some other measures, such as percent misprediction and feature expected distance, are used from a



(a) Ground Truth V.S. LEARCH-based IRL



(b) Ground Truth V.S. MWAL-based IRL

Fig. 1: The opposite results given by EVD and MRL under the Object World benchmark (a) LEARCH algorithm and (b) MWAL algorithm.

very different angle (state action value function) to indirectly reflect the reward function's similarity, that is, to compare the new learned policy's action with an expert's action of trajectories. Furthermore, a RL algorithm's learning score is obtained through using an approximated reward to actually interact with the environment. However, from the reverse engineering perspective, the percent misprediction, feature expected distance, and learning score, are far from direct and unsuitable. Therefore, we introduce mutual information to assess the discreteness similarity between original rewards and rewards by approximated IRL. [30] proposed a mutual information-based method for improving IRL, they just used the mutual information for features ranking based on the relevant evaluation results in the IRL training process; this is totally different from our use in assessing discrete reward clusters.

III. Motivating Example

We choose Object World, a popular game that has appeared in many IRL-related experiments. We choose two different IRL algorithms, LEARCH and MWAL, to analyze the confused IRL results. We use the metrics MRL and EVD.

Figure 1 presents two graphs for comparison. The left graph shows the ground truth of true discrete rewards. There are only three types of rewards: +1 means an agent reaching a white square, -1 for the black square, and 0 for the gray square. The IRL results are shown in the right part of each graph. For comparison, we use a red dotted line to surround each cluster of +1 rewards, and a green dotted line for -1 reward clusters. Moreover, for the right graph in Figure 1 (a) and (b) we utilize different color dotted lines to represents the clustering result given by IRL algorithm, LEARCH and MWAL respectively.

We can see that when assessing with MRL metric, it indicates that the MWAL's performance is better (MRL=4.5314); meanwhile, under the assessment of EVD shows that the LEARCH gives the better result (EVD=16.6681). Obviously, under such scenario, MRL and EVD give completely opposite assessing results, and which metric should we trusted to determine the effectiveness of IRL algorithm? This is the fundamental point that inspires us to rethink whether the current metrics are sufficient for accurate evaluation. One possible reason is that MRL focuses the entire state space and get the mean reward loss, while EVD pay attention to the expected value of several sampling trajectories. Thus, it is necessary to develop a novel metric that can fill the gap with middle granularity between the entire state space and the specific sampling space, further, to give an accurate evaluation result. In fact, from the aspect of clustering result, we can see that the result of LEARCH is much similar to the original ground truth, which inspires us that reward cluster-related features should be considered in an accurate assessment.

IV. Preliminaries

A. Reinforcement learning (RL)

The five-tuple $\mathcal{M} = \{\mathcal{S}, \mathcal{A}, P, \gamma, \boldsymbol{r}\}$, is generally be utilized to represented the markov decision process, in which $\mathcal{S}$ denotes the state space, and $\mathcal{A}$ represents the action space, $P(s'|s,a)$ denotes the corresponding transition probability that transfers from state s to s' $(s, s' \in \mathcal{S})$ with action selection $a \in \mathcal{A}$, γ denotes the discount factor and it ranges from 0 to 1, the fifth element $\boldsymbol{r}$ denotes the reward function. And the optimal policy π^* can be denoted as $\pi^* = \arg\max_\pi \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t r_{s_t}|\pi]$.

The purpose of RL is to obtain a policy, for which the input should be the observation s, and the output is the action selection probability under each state. At time t the action selecting is a_t, and the policy can be computed as the probability $p(a_t|s_t)$.

From the beginning to the end of a certain task, the agent can generate a trajectory $\tau = \{s_1, a_1, \ldots, s_T, a_T\}$. The state-action value function can be calculated as $Q_\pi(s_t, a_t) = \mathbb{E}(r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \ldots |s_t, a_t)$, and it an be simplified as $\mathbb{E}(\gamma^{t'-t} r^n_{t'+1}|s_t, a_t), t' \in \{t, t+1, \ldots, T\}$. According to the Bellman equation (dynamic programming equation), the state value function can be represented as $V_\pi(s_t) = \sum_{a_t} \pi(a_t|s_t) Q_\pi(s_t, a_t)$. Thus, the state-action value function can also be represented as $Q_\pi(s_t, a_t) = \mathbb{E}[r_t + V_\pi(s_{t+1})]$. In order to obtain a good policy, $\bar{R}_{\boldsymbol{\theta}} = \sum_\tau R(\tau) p_{\boldsymbol{\theta}(\tau)}$ should be maximized through a gradient ascent-based update $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \eta \nabla \bar{R}_{\boldsymbol{\theta}}$.

B. Inverse reinforcement learning (IRL)

IRL can be described as $\mathcal{M} \backslash \boldsymbol{r}$ together with the expert demonstration $\mathcal{D}^* = \{\tau_1, \tau_2, \ldots\}$, where τ_i is represented as $\tau_i = \{s_{i,1}, a_{i,1}, \ldots, s_{i,T}, a_{i,T}\}$. The purpose of IRL is to find a reward function r, and under r the optimal policy π^* can give the maximum probability when following the given expert trajectories. The probability of choosing action a at state s can be represented as $P(a|s) \propto \exp(Q_r(s,a))$, in which $Q_r(s,a) = \mathbb{E}[r + V_r(s')]$. While the state value function $V_r(s)$

can be calculated as $\sum_a p(a|s)Q_r(s,a)$, thus, $P(a|s)$ can be represented as $\exp(Q_r(s,a) - V_r(s))$. Under reward function $\boldsymbol{r}$ the log likelihood of the given expert trajectories can be represented as:

$$\begin{aligned}\log p(\mathcal{D}^*|\boldsymbol{r}) &= \sum_i \sum_t \log p(a_{i,t}|s_{i,t}), \\ &= \sum_i \sum_t (Q_{\boldsymbol{r}}(s_{i,t}, a_{i,t}) - V_{\boldsymbol{r}}(s_{i,t})),\end{aligned} \tag{1}$$

and maximizing Equation 1 directly to obtain reward function $\boldsymbol{r}$.

C. Classical Metrics: MRL and EVD

Mean reward loss (MRL) measures the average difference between the learned rewards and the true rewards, and MRL can be calculated as follows:

$$MRL(\boldsymbol{r}^e, \boldsymbol{r}^u) = \frac{1}{|\mathcal{S}|}\sum_{i=1}^{|\mathcal{S}|} |r^e_{s_i} - r^u_{s_i}|, s_i \in \mathcal{S}, \tag{2}$$

in which $\boldsymbol{r}^e$ denotes the true reward, and we use $\boldsymbol{r}^u$ to represent the learned reward.

Expected value difference (EVD) measures the difference between the expected cumulated reward under the ground truth rewards and the learned rewards, which can be represented as follows:

$$EVD(\boldsymbol{r}^e, \boldsymbol{r}^u) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t r^e_{s_t} | \pi^*] - \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t r^e_{s_t} | \pi^u], \tag{3}$$

where π^* denotes the optimal policy under the true rewards $\boldsymbol{r}^e$, and π^u is derived from the IRL's inverse rewards $\boldsymbol{r}^u$. $\mathbb{E}[\sum_{t=0}^{\infty} \gamma^t r^e_{s_t}|\pi^*] \approx \frac{1}{\zeta}\sum_{i=1}^{\zeta}\sum_{t=1}^{T} \gamma^t r^e_{s_{i,t}}, s \in \boldsymbol{\tau}^*$, $\mathbb{E}[\sum_{t=0}^{\infty} \gamma^t r^e_{s_t}|\pi^u] \approx \frac{1}{\zeta}\sum_{i=1}^{\zeta}\sum_{t=1}^{T} \gamma^t r^e_{s^u_{i,t}}, s^u \in \boldsymbol{\tau}^u$, where ζ is the number of selected trajectories; and $\boldsymbol{\tau}^*/\boldsymbol{\tau}^u$ denotes the T steps trajectory generated by policy π^*/π^u.

MRL and EVD both range from 0 to $+\infty$, the smaller their value, the better the IRL performance.

V. Mutual Information-Based IRL Assessment

In this section, as shown in Figure 2, we propose our mutual information-based assessment in detail. We define and compute the reward clustering-based metric, and present our comprehensive 4-order tensor model embedded in manipulated trajectory.

A. Generating reward clusters

Given the reward set $\boldsymbol{r}$, we utilize the he agglomerative nesting algorithm (AGNES) to obtain the reward-close clusters. As shown in Algorithm 1, which begins with $|\boldsymbol{r}|$ clusters. In line 5, we calculate the minimum distance between cluster $\mathcal{C}_\alpha$ and $\mathcal{C}_\beta$, which can be calculated as follows:

$$\begin{aligned}(\mathcal{C}_\alpha, \mathcal{C}_\beta) &= \underset{\mathcal{C}_i \in \boldsymbol{\mathcal{C}}_x, \mathcal{C}_j \in \boldsymbol{\mathcal{C}}_x}{\arg\min} \ Dal(\mathcal{C}_i, \mathcal{C}_j), \\ &= \underset{\mathcal{C}_i \in \boldsymbol{\mathcal{C}}_x, \mathcal{C}_j \in \boldsymbol{\mathcal{C}}_x}{\arg\min} \frac{1}{|\mathcal{C}_i||\mathcal{C}_j|}\sum_{r_i \in \mathcal{C}_i}\sum_{r_j \in \mathcal{C}_j} d(r_i, r_j),\end{aligned} \tag{4}$$

where $\mathcal{C}_i$ and $\mathcal{C}_j$ are clusters, and $\mathcal{C}_i \bigcap \mathcal{C}_j = \emptyset$, $d(r_i, r_j)$ is the Euclidean distance [35] between two rewards $r_i \in \mathcal{C}_i$ and $r_j \in \mathcal{C}_j$. In line 6, we merge two individual clusters whose distance is shortest into a larger cluster $New\_\mathcal{C}$, and reform the clustering $\boldsymbol{\mathcal{C}}_x$ as follows:

$$\boldsymbol{\mathcal{C}}_x = (\boldsymbol{\mathcal{C}}_x \setminus \{\mathcal{C}_\alpha, \mathcal{C}_\beta\}) \bigcup \{New\_\mathcal{C}\}. \tag{5}$$

We then repeat the operations from line 4 to line 8 until the number of cluster reaches $|\boldsymbol{k}|$.

Algorithm 1 Reward clustering

```
Input: reward set $\boldsymbol{r} = \{r_1, r_2, \dots, r_{|\boldsymbol{r}|}\}$, cluster number $\boldsymbol{k} = \{k_1, k_2, \dots, k_{|\boldsymbol{k}|}\}$
Output: $\{\boldsymbol{\mathcal{C}}_1, \boldsymbol{\mathcal{C}}_2, \dots, \boldsymbol{\mathcal{C}}_{|\boldsymbol{k}|}\}$
 1: repeat
 2:     $x \leftarrow 1$
 3:     // initial setting
        $\boldsymbol{\mathcal{C}}_x = \{\mathcal{C}_{x;1}, \dots, \mathcal{C}_{x;|\boldsymbol{r}|}\} = \{\{r_1\}, \dots, \{r_{|\boldsymbol{r}|}\}\}$
 4:     repeat
 5:         // individual clusters merging
            Calculating the minimum distance between cluster $\mathcal{C}_\alpha$ and $\mathcal{C}_\beta$ with Equation 4
 6:         $New\_\mathcal{C} = \mathcal{C}_\alpha \bigcup \mathcal{C}_\beta$
 7:         Reforming clustering $\boldsymbol{\mathcal{C}}_x$ with Equation 5
 8:     until $|\boldsymbol{\mathcal{C}}_x| \geq k_x$
 9:     $x \leftarrow x + 1$
10: until $x > |\boldsymbol{k}|$
11: return $\{\boldsymbol{\mathcal{C}}_1, \boldsymbol{\mathcal{C}}_2, \dots, \boldsymbol{\mathcal{C}}_{|\boldsymbol{k}|}\}$
```

B. C-NMI

Given the ground truth reward set $\boldsymbol{r}^e$, if the number of clusters is assumed as $k_x \in \boldsymbol{k}$ for clusters $\boldsymbol{\mathcal{C}}^e_x$, then we have $\boldsymbol{\mathcal{C}}^e_x = \{\mathcal{C}^e_{x;1}, \mathcal{C}^e_{x;2}, \dots, \mathcal{C}^e_{x;k_x}\}$; furthermore, $\mathcal{C}^e_{x;i} \bigcap \mathcal{C}^e_{x;j} = \emptyset$ $(i, j \in \{1, 2, \dots, k_x\})$, and $\bigcup_{i=1}^{k_x} \mathcal{C}^e_{x;i} = \boldsymbol{r}^e$.

In order to compute the mutual information of reward clusters, we first conduct state space sampling to make the cluster comparison under the same space. Figure 3 illustrates this state space sampling. We first descend all of the clusters in $\boldsymbol{\mathcal{C}}^e_x$ based on the size of each cluster $|\mathcal{C}^e_{x;i}|$. According to the top $\mathcal{K}$ of cluster ranking $o_y \in \boldsymbol{o}$, we form $\boldsymbol{\mathcal{C}}^e_{x,y} = \{\mathcal{C}^e_{x,y;1}, \dots, \mathcal{C}^e_{x,y;o_y}\}$, and then obtain the corresponding state space $\mathcal{S}'$ as follows:

$$\mathcal{S}' = \{s' | r^e_{s'} \in \mathcal{C}^e_{x,y;i}\}, i = \{1, \dots, o_y\}, \tag{6}$$

where $\mathcal{C}^e_{x,y;i}$ denotes any subset in $\boldsymbol{\mathcal{C}}^e_{x,y}$.

Next, we compute the mutual information of reward clusters (C-MI) between $\boldsymbol{\mathcal{C}}^u_{x,y}$ and $\boldsymbol{\mathcal{C}}^e_{x,y}$ by measuring how much reward information one cluster gives about another. $C-MI(\boldsymbol{\mathcal{C}}^u_{x,y}; \boldsymbol{\mathcal{C}}^e_{x,y})$ can be calculated as follows:

$$\sum_{i=1}^{|\boldsymbol{\mathcal{C}}^u_{x,y}|}\sum_{j=1}^{|\boldsymbol{\mathcal{C}}^e_{x,y}|} p(\mathcal{C}^u_{x,y;i}, \mathcal{C}^e_{x,y;j}) \log \frac{p(\mathcal{C}^u_{x,y;i}, \mathcal{C}^e_{x,y;j})}{p(\mathcal{C}^u_{x,y;i})p(\mathcal{C}^e_{x,y;j})}, \tag{7}$$

where $\boldsymbol{\mathcal{C}}^u_{x,y}$ is the clustering result for $\boldsymbol{r}^u$. $p(\mathcal{C}^u_{x,y;i}, \mathcal{C}^e_{x,y;j}) = \frac{|\mathcal{C}^u_{x,y;i} \bigcap \mathcal{C}^e_{x,y;j}|}{|\mathcal{S}'|}$ denotes the probability that a certain reward r belonging $\mathcal{C}^u_{x,y;i} \subset \boldsymbol{\mathcal{C}}^u_{x,y}$ and $\mathcal{C}^e_{x,y;j} \subset \boldsymbol{\mathcal{C}}^e_{x,y}$. Meanwhile,

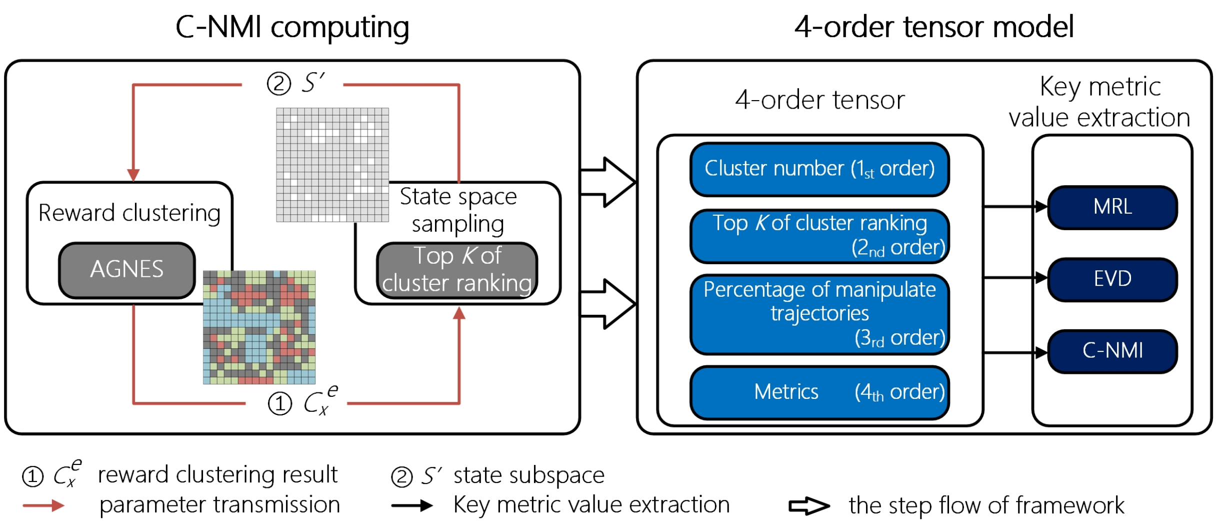

C-NMI computing
4-order tensor model
② S′
Reward clustering
AGNES
State space sampling
Top K of cluster ranking
① C_x^e
4-order tensor
Key metric value extraction
Cluster number (1st order)
Top K of cluster ranking (2nd order)
Percentage of manipulate trajectories (3rd order)
Metrics (4th order)
MRL
EVD
C-NMI
① C_x^e reward clustering result
② S′ state subspace
parameter transmission
Key metric value extraction
the step flow of framework

Fig. 2: Illustration of mutual information-based IRL assessment. The whole framework contains two parts, in which the first one is the C-NMI computing, and the second one is the 4-order tensor model.

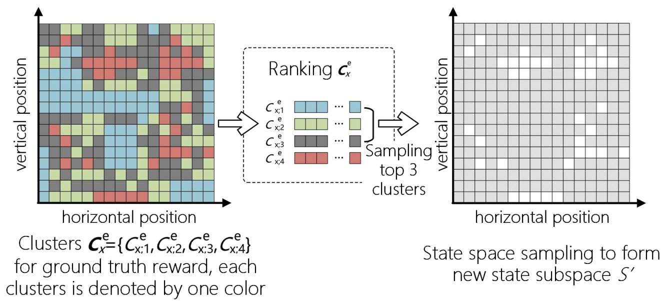

vertical position
horizontal position
Ranking C_x^e
Sampling top 3 clusters
Clusters C_x^e={C_{x1}^e, C_{x2}^e, C_{x3}^e, C_{x4}^e} for ground truth reward, each clusters is denoted by one color
State space sampling to form new state subspace S′

Fig. 3: Illustration of state space sampling. For the left map, different colors denote different clusters, in which "blue" represents cluster $\mathcal{C}^e_{x;1}$, "green" denotes cluster $\mathcal{C}^e_{x;2}$, "black" is cluster $\mathcal{C}^e_{x;3}$, and "red" is cluster $\mathcal{C}^e_{x;4}$. Then ranking $\boldsymbol{\mathcal{C}^e_x}$ based on each clusters' size, and sampling the top 3 clusters to form a new state space $\mathcal{S}'$ which colored by "gray", while the remaining states is colored with "white".

$p(\mathcal{C}^u_{x,y;i}) = \frac{|\mathcal{C}^u_{x,y;i}|}{|\mathcal{S}'|}$ represents the probability that r belonging $\mathcal{C}^u_{x,y;i}$. Similarly, $p(\mathcal{C}^e_{x,y;j}) = \frac{|\mathcal{C}^e_{x,y;j}|}{|\mathcal{S}'|}$.

The C-MI$(\boldsymbol{\mathcal{C}}^u_{x,y}; \boldsymbol{\mathcal{C}}^e_{x,y})$ ranges from 0 to $+\infty$, and the C-MI will increase with the higher dependence of clusters $\boldsymbol{\mathcal{C}}^u_{x,y}$ and $\boldsymbol{\mathcal{C}}^e_{x,y}$. C-MI=0 if and only if $\boldsymbol{\mathcal{C}}^u_{x,y}$ and $\boldsymbol{\mathcal{C}}^e_{x,y}$ are completely independent.

To rescale C-MI, we calculate the normalized mutual information of reward clusters (C-NMI) between $\boldsymbol{\mathcal{C}}^u_{x,y}$ and $\boldsymbol{\mathcal{C}}^e_{x,y}$, which can be represented as follows:

$$C-NMI(\boldsymbol{\mathcal{C}}^u_{x,y}, \boldsymbol{\mathcal{C}}^e_{x,y}) = \frac{C-MI(\boldsymbol{\mathcal{C}}^u_{x,y}; \boldsymbol{\mathcal{C}}^e_{x,y})}{\sqrt{H(\boldsymbol{\mathcal{C}}^u_{x,y}) \cdot H(\boldsymbol{\mathcal{C}}^e_{x,y})}}, \tag{8}$$

where $H(\boldsymbol{\mathcal{C}}^u_{x,y}) = -\sum_{i=1}^{|\boldsymbol{\mathcal{C}}^u_{x,y}|} p(\mathcal{C}^u_{x,y;i}) \log p(\mathcal{C}^u_{x,y;i})$, $H(\boldsymbol{\mathcal{C}}^e_{x,y}) = -\sum_{j=1}^{|\boldsymbol{\mathcal{C}}^e_{x,y}|} p(\mathcal{C}^e_{x,y;j}) \log p(\mathcal{C}^e_{x,y;j})$, and C-NMI$(\boldsymbol{\mathcal{C}}^u_{x,y}, \boldsymbol{\mathcal{C}}^e_{x,y}) \in [0, 1]$. C-NMI=1 means that the two clusters are exactly coincident.

C. 4-order tensor model

Towards dynamic compositions of variables including cluster number, top $\mathcal{K}$ of cluster ranking, and percentage of manipulated trajectories, we design a 4-order tensor model to uniformly capture and store different-dimension metric values of MRL, EVD, and C-NMI. Hence, we can conduct a more comprehensive assessment, as well as evaluate the accuracy for our proposed metric C-NMI.

Figure 4 depicts the overall architecture of the 4-order tensor model. This architecture contains three layers. The 4-order tensor layer is responsible for capturing and storing multi-dimensional metrics. The second layer is designed for extracting key metric values based on the interquartile range (IQR) of statistics, and the output layer outputs the final results for assessment.

4-order tensor We construct a 4-order tensor $\boldsymbol{B} \in \mathbb{R}^{X \times Y \times Z \times V}$, where $X = |\boldsymbol{k}|$, $Y = |\boldsymbol{o}|$, $Z = |\boldsymbol{m}|$, and $V = |\boldsymbol{g}|$. In the third order, we implement a method (see Algorithm 2) to inject suboptimal and inverted trajectories as manipulated ones. Randomly choosing a subspace $\boldsymbol{\pi'} \subset \boldsymbol{\pi}$, and $\boldsymbol{\pi'}$ contains the optimal policy π^*. The operation command $cmd == 0$ means injecting suboptimal trajectories as manipulated ones; else, it means injecting inverted trajectories. From line 2 to line 3, we compute the suboptimal policy π^{sub} and generate the corresponding trajectories τ_i^{sub}, which can be calculated as follows:

$$\pi^{sub} = \arg \max_{\boldsymbol{\pi'} \backslash \{\pi^*\}} \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t r^e_{s_t} | \pi], \tag{9}$$

$$\tau_i^{sub} = \{s_{i,1}, a_{i,1}, \ldots, s_{i,T}, a_{i,T} | \pi^{sub}\}. \tag{10}$$

In line 5, we invert $\boldsymbol{r}^e$, and obtain the inverted reward set $\boldsymbol{r}^{inv}$. From line 6 to line 7, we compute the inverted policy, and we generate the inverted trajectories as follows:

$$\pi^{inv} = \arg \max_{\pi} \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t r^{inv}_{s_t} | \pi], \tag{11}$$

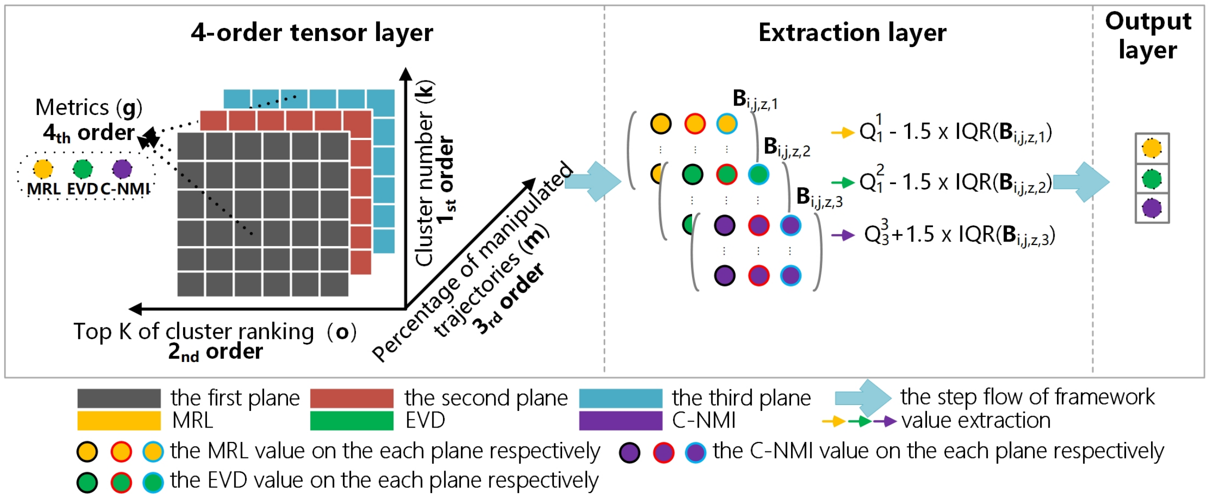

4-order tensor layer
Extraction layer
Output layer
Metrics (g) 4th order
MRL EVD C-NMI
Cluster number (k) 1st order
Percentage of manipulated trajectories (m) 3rd order
Top K of cluster ranking (o) 2nd order
B i,j,z,1
B i,j,z,2
B i,j,z,3
Q1^1 - 1.5 x IQR(B i,j,z,1)
Q1^2 - 1.5 x IQR(B i,j,z,2)
Q3^3+1.5 x IQR(B i,j,z,3)
the first plane
the second plane
the third plane
the step flow of framework
MRL
EVD
C-NMI
value extraction
the MRL value on the each plane respectively
the C-NMI value on the each plane respectively
the EVD value on the each plane respectively

Fig. 4: Framework of the 4-order tensor model. This framework contains three layers, in which the first layer is the 4-order tensor layer, the second one is the extraction layer, and the last one is the output layer.

$$\tau_i^{inv} = \{s_{i,1}, a_{i,1}, \ldots, s_{i,T}, a_{i,T} | \pi^{inv}\}. \tag{12}$$

We then repeat operations from line 9 to line 14 to inject τ_i^{sub} and τ_i^{inv} as manipulated trajectories until the control percentage reaches $p_{|\boldsymbol{p}|}$, which can be represented as follows:

$$\mathcal{D}_i^{s*} = \{\tau_1^{sub}, \ldots, \tau_{p_i \cdot |\mathcal{D}^{s*}|}^{sub}, \tau_1^*, \ldots, \tau_{(1-p_i)\cdot|\mathcal{D}^{s*}|}^*\}, \tag{13}$$

$$\mathcal{D}_i^{i*} = \{\tau_1^{inv}, \ldots, \tau_{p_i \cdot |\mathcal{D}^{i*}|}^{inv}, \tau_1^*, \ldots, \tau_{(1-p_i)\cdot|\mathcal{D}^{i*}|}^*\}. \tag{14}$$

Finally, we output the manipulated demonstrations $\boldsymbol{\mathcal{D}^{s*}}$ and $\boldsymbol{\mathcal{D}^{i*}}$.

Thus, the 3rd order $\boldsymbol{m}$ contains $|\boldsymbol{p}|$-dimension features, which can be represented as $(m_1, \ldots, m_{|\boldsymbol{p}|}) = (p_1, p_2, \ldots, p_{|\boldsymbol{p}|})$; meanwhile, the 4th order $\boldsymbol{g}$ has three-dimension features to separately represent three metrics, and it can be represented as $(g_1, g_2, g_3) = (\text{MRL, EVD, C-NMI})$. By expanding tensor $\boldsymbol{B}$ along $\boldsymbol{g}$, we can obtain the matrix $\boldsymbol{B}_{(\boldsymbol{g})}$:

$$\boldsymbol{B}_{(\boldsymbol{g})} = \begin{bmatrix} \boldsymbol{B}_{x,y,z,1} \\ \boldsymbol{B}_{x,y,z,2} \\ \boldsymbol{B}_{x,y,z,3} \end{bmatrix}, \tag{15}$$

where $\boldsymbol{B}_{x,y,z,1} = (B_{1,1,1,1}, \ldots, B_{|\boldsymbol{k}|,1,1,1}, \ldots, B_{1,|\boldsymbol{o}|,|\boldsymbol{m}|,1}, \ldots, B_{|\boldsymbol{k}|,|\boldsymbol{o}|,|\boldsymbol{m}|,1})$, $\boldsymbol{B}_{x,y,z,2} = (B_{1,1,1,2}, \ldots, B_{|\boldsymbol{k}|,|\boldsymbol{o}|,|\boldsymbol{m}|,2})$, and $\boldsymbol{B}_{x,y,z,3} = (B_{1,1,1,3}, \ldots, B_{|\boldsymbol{k}|,|\boldsymbol{o}|,|\boldsymbol{m}|,3})$.

Key metric value extraction Through the 4-order tensor, we obtain the multi-dimensional sequence for each metric. $\boldsymbol{B}_{x,y,z,1}$ denotes the sequence for MRL, $\boldsymbol{B}_{x,y,z,2}$ denotes the sequence of EVD and $\boldsymbol{B}_{x,y,z,3}$ represents the corresponding sequence C-NMI. Then, we extract the key metric values of MRL, EVD, and C-NMI based on the IQR of the corresponding multi-dimensional sequence, because it is a commonly used robust measure of scale [21]. The IQR of $\boldsymbol{B}_{x,y,z,v}$ can be computed as follows:

$$IQR(\boldsymbol{B}_{x,y,z,v}) = Q_3^v - Q_1^v, \tag{16}$$

Algorithm 2 Manipulated trajectory injection

```
Input: expert demonstration $\mathcal{D}^*$, ground truth rewards $\boldsymbol{r}^e$,
    policy subspace $\boldsymbol{\pi'} = \{\pi^*, \pi_2, \ldots, \pi_{|\boldsymbol{\pi'}|}\}$, percentage
    controlling coefficient $\boldsymbol{p} = \{p_1, \ldots, p_{|\boldsymbol{p}|}\}$, $p_i \in [0,1]$,
    operation command $cmd$
Output: manipulated demonstrations $\boldsymbol{\mathcal{D}^{s*}}$,$\boldsymbol{\mathcal{D}^{i*}}$
 1: if $cmd$==0 then
 2:     Computing the suboptimal policy $\pi^{sub}$ with Equation 9
 3:     Generating the suboptimal trajectory $\tau_i^{sub}$ with Equation 10
 4: else
 5:     $\boldsymbol{r}^{inv} = -\boldsymbol{r}^e$
 6:     Computing the inverted policy $\pi^{inv}$ with Equation 11
 7:     Generating the inverted trajectory $\tau_i^{inv}$ with Equation 12
 8: end if
 9: repeat
10:     // control percentage of manipulated trajectories
        $i \leftarrow 1$
11:     Infecting $\tau_i^{sub}$ as manipulated trajectory and obtain $\mathcal{D}_i^{s*}$ with Equation 13
12:     Infecting $\tau_i^{inv}$ as manipulated trajectory and obtain $\mathcal{D}_i^{i*}$ with Equation 14
13:     $i \leftarrow i+1$
14: until $i > |\boldsymbol{p}|$
15: return $\boldsymbol{\mathcal{D}^{s*}} = \{\mathcal{D}_1^{s*}, \ldots, \mathcal{D}_{|\boldsymbol{p}|}^{s*}\}$, $\boldsymbol{\mathcal{D}^{i*}} = \{\mathcal{D}_1^{i*}, \ldots, \mathcal{D}_{|\boldsymbol{p}|}^{i*}\}$
```

where Q_3^v and Q_1^v indicate the upper- and lower-quartiles of $\boldsymbol{B}_{x,y,z,v}$, respectively. Thus, the upper 1.5× IQR whiskers can be represented as $Q_3^v + 1.5 \times IQR(\boldsymbol{B}_{x,y,z,v})$, and the lower 1.5×IQR whiskers is $Q_1^v - 1.5 \times IQR(\boldsymbol{B}_{x,y,z,v})$. These can separately characterize the highest and lowest occurring values of $\boldsymbol{B}_{x,y,z,v}$, thereby avoiding the influence of outliers.

According to the definitions of MRL, EVD, and C-NMI, we set the lower 1.5× IQR whiskers of $\boldsymbol{B}_{x,y,z,1}$ and $\boldsymbol{B}_{x,y,z,2}$ as the key metric values of MRL and EVD, respectively. We

TABLE II: Experimental environment configuration

Platform	Experiment environment	Environment configuration
IRL algorithms	GPU	MSI GeForce RTX 2070 VENTUS
	RAM	32GB
	Graphic Memory	151MiB
	Software	Matlab R2017b

set the upper 1.5× IQR of $\boldsymbol{B}_{x,y,z,3}$ as the key metric value for C-NMI.

VI. Experiments

A. Experimental setup

Benchmark The parameters for the experiment platform are shown in Table II. Since the Object World is most commonly used in IRL experiments, we utilize this environment as our experimental benchmark. Object World is a gridworld where dots of *primary colors*, e.g., blue and green, and *secondary colors*, e.g., red and cyan, which are placed on the grid randomly, as shown in Figure 1. In Object World, the agent maximizes its expected discounted reward by following a policy that provides the probabilities of actions (moving up/down/left/right or staying still) at each state, with each subject to a transition probability. Each state, i.e., grid block, is described by the shortest distances to dots among each color group. The reward is assigned such that if a block is 1 step within a green dot and 3 steps within a blue dot, the reward is $+1$; if it is 3 steps within a blue dot only, the reward is -1; and 0 otherwise. Considering the number of *primary colors* $Col_1 = 2$, and the number of *secondary colors* $Col_2 = 2$. For constructing the benchmark of Object World, we use a common Matlab function named *round(rand(1))* to assign each grid a value of 0 or 1; if it is 1, then we place an object on this grid. Meanwhile, each object is randomly assigned one of the Col_1 primary and Col_2 secondary colors.

Parameter setting For the 4-order tensor, we vary the cluster number k in the range $\{2, 3, \ldots, 10\}$, and the top $\mathcal{K}$ of cluster ranking in the range $\{1, \ldots, k\}$. We also vary the percentage controlling coefficient p in $\{10\%, 30\%, 50\%\}$. The trajectory length is set to be 8, and $|\mathcal{D}| = 128$ following the literature [53], thereby ensuring satisfied training converge of the IRL algorithm. For the 7 state-of-the-art IRL algorithms, we directly utilize the standard setting follow the corresponding works: MMP [55], MWAL [56], MaxEnt [54], AN [40], GPIRL [53], LEARCH [39], and FIRL [41].

Datasets According to the grid map size, we build six datasets for experiment analysis, which can be recorded as $\boldsymbol{DS} = \{DS_{5\times5}, \ldots, DS_{50\times50}\}$. Moreover, for each training dataset DS we utilize the IRL algorithm toolkit which is developed by Levine et al. [53] to generate 20 grid map samples randomly, which can be represented as $DS_{i\times i} = \{ds_i^1, ds_i^2, \ldots, ds_i^{20}\}$.

Baseline In the field of IRL algorithm assessing, there is not existing one specific "ground truth" to give an expert evaluation, as different researchers utilize different evaluation metrics, such as only use EVD, MRL, or use both of them.

TABLE III: Definition of function F_{bl}

l_{mrl}	l_{evd}	baseline l
1	1	1
-1	-1	-1
1	0/-1	0
-1	0/1	0
0	1/-1	0

TABLE IV: Definition of function F_{acc}

$l(A)$	$l_{metric}(A)$	$F_{acc}(l(A) - l_{metric}(A))$
1	1	1
0	0	1
-1	-1	1
1	0/-1	0
0	1/-1	0
-1	1/0	0

Thus, in order to evaluate the assessing accuracy of C-NMI, we utilize MRL together with EVD to design a baseline for accuracy comparison among different metrics. Through ranking MRL and EVD values in ascending order, we can obtain two ordered sets $\boldsymbol{mrl}$ and $\boldsymbol{evd}$. Then, we divide $\boldsymbol{mrl}$ and $\boldsymbol{evd}$ into three subsets with same sizes, and use $tertiles_1()$ and $tertiles_2()$ to represent the lower- and upper-tertiles [25], respectively.

Using the IRL algorithm A, we define the function $MRL(A) = Q_1^1 - 1.5 \times IQR(\boldsymbol{B}_{i,j,z,1})$. If $MRL(A) \leq tertiles_1(\boldsymbol{mrl})$, A has a good performance, and tagging A with $l_{mrl}(A) = 1$; while $MRL(A) \geq tertiles_2(\boldsymbol{mrl})$ shows that A performs poor, thus tag A with $l_{mrl}(A) = -1$; lastly, $l_{mrl}(A) = 0$ indicates moderate performance. Similarly, we also have $l_{evd}(A) = 1, -1, or 0$. Finally, we define the baseline as follows:

$$l(A) = F_{bl}(l_{mrl}(A) + l_{evd}(A)). \tag{17}$$

F_{bl} is defined in Table III. If $l_{mrl}(A) + l_{evd}(A) = 2$, then $F_{bl}(l_{mrl}(A) + l_{evd}(A)) = 1$; if $l_{mrl}(A) + l_{evd}(A) = -2$, the value of F_{bl} equals -1; and 0 otherwise.

Evaluation Metrics According to the baseline, under a specific demonstration $\mathcal{D} \in \boldsymbol{\mathcal{D}} = \{\mathcal{D}^*, \mathcal{D}_1^{s*}, \ldots, \mathcal{D}_{|\boldsymbol{p}|}^{s*}, \mathcal{D}_1^{i*}, \ldots, \mathcal{D}_{|\boldsymbol{p}|}^{i*}\}$, given A, we can calculate the assessing accuracy of metric, which can be represented as follows:

$$\epsilon_{metric}(A^{\mathcal{D}}) = \frac{\sum_{i=1}^{|\boldsymbol{DS}|} \sum_{j=1}^{|DS_{i\times i}|} F_{acc}(l(A_{ds_i^j}^{\mathcal{D}}) - l_{metric}(A_{ds_i^j}^{\mathcal{D}}))}{|\boldsymbol{DS}| \cdot |DS_{i\times i}|}, \tag{18}$$

in which ds_i^j denotes the jth sample in $DS_{i\times i}$. F_{acc} is defined in Table IV. If $l(A_{ds_i^j}^{\mathcal{D}}) - l_{metric}(A_{ds_i^j}^{\mathcal{D}}) = 0$, then $F_{acc}(l(A_{ds_i^j}^{\mathcal{D}}) - l_{metric}(A_{ds_i^j}^{\mathcal{D}})) = 1$; otherwise, F_{acc} equals 0. The average assessing accuracy across different demonstrations can be calculated as follows:

$$\bar{\epsilon}_{metric}(A) = \frac{\sum^{|\boldsymbol{\mathcal{D}}|} \epsilon_{metric}(A^{\mathcal{D}})}{|\boldsymbol{\mathcal{D}}|}, \tag{19}$$

in which $metric \in \{mrl, evd, c\text{-}nmi, ws\}$, and ws represents the weighted sum of metric MRL and EVD which can be denoted as $ws = \omega_1 \cdot \text{MRL} + \omega_2 \cdot \text{EVD}$, $(\omega_1, \omega_2) \in \{(\frac{1}{2}, \frac{1}{2}), (\frac{2}{5}, \frac{3}{5}), (\frac{3}{5}, \frac{2}{5})\}$.

For $l_{c-nmi}(A)$, similarly, ranking C-NMI values by descending order and obtaining ordered set ***c-nmi***. We make comparison of *C-NMI*$(\mathcal{C}^e, \mathcal{C}^A)$ and $tertiles_{1/2}($***c-nmi***$)$ under the given learned result $\mathcal{C}^A$. Thus, we can tag A with $l_{c-nmi}(A) = 1, -1$ or 0. In the same way, we can obtain $l_{ws}(A)$. Noting that $\epsilon, \bar{\epsilon}$ both range from 0 to 1, and the larger $\epsilon, \bar{\epsilon}$ represent the performance is better.

B. Average assessing accuracy analysis of C-NMI

In this section, to compare the average assessing accuracy of different metrics, we take MRL, EVD, C-NMI, and the weighted sum metrics $\omega_1 \cdot \text{MRL} + \omega_2 \cdot \text{EVD}$ into consideration. As shown in Table V, we can obtain several observations: under LEARCH algorithm, C-NMI increases 116.95% and 132.69% respectively when comparing with EVD and MRL metric. In addition, the average metric value among different IRLs has increased 110.13% and 116.59% respectively when compared with EVD and MRL. Under GPIRL algorithm, C-NMI will give the highest accuracy.

For ws metric, when the combination of $(\omega_1, \omega_2) = (\frac{2}{5}, \frac{3}{5})$ results in the highest accuracy (0.89) with GPIRL. Under most algorithms, C-NMI's performance is better than the ws. Furthermore, when the combination is $(\frac{2}{5}, \frac{3}{5})$ which can give the best performance, our the average assessing accuracy of C-NMI improves the mean by up to 72.69%. Thus, generally speaking, compared with all other assessing metrics C-NMI has the best performance.

In order to verify the above observations more intuitively, we visualize the rewards learned through 7 mainstream IRLs for a case study. We target the same 40×40 grid map for comparison. Eight subgraphs are presented in Figure 5 for comparison: subfigure (a) shows the ground truth rewards, and subfigures (b) to (h) reveal the learned rewards under different IRL algorithms. Table VI compares the corresponding key metric values of MRL, EVD, and C-NMI with the baseline. We mark the key metric value with an underline if $l_{metric}(A) = -1$, and bold if $l_{metric}(A) = 1$. It can be seen that MRL gives a wrong assessment of MWAL, and EVD makes a mistake with MaxEnt; conversely, C-NMI gives exactly the correct assessment for all of the IRL algorithms compared to the baseline.

C. Hyperparameters analysis

Cluster number (k) Under different cluster numbers, the performance of C-NMI metric is shown in Figure 7(a). Varying the cluster number k in the range of $\{2, 3, \ldots, 10\}$, and record the average C-NMI assessing accuracy for all k. It can be seen that a higher average assessing accuracy is obtained for a larger k, which reveals that, with the increase of cluster number k, the rewards division is finer, and the C-NMI can give a better evaluating performance of IRL algorithms. When $k \geq 8$, C-NMI can reach the optimal $\bar{\epsilon}$. Thus, 8 is a good choice for k in the C-NMI computing.

Top $\mathcal{K}$ of cluster ranking (o) In order to verify o's influence on the average assessing accuracy of C-NMI, we vary o in the range of $\{2, 3, \ldots, 10\}$, then make comparison among 7 mainstream IRLs. As shown in Figure 7(b), $o = 7$ is a good choice, since the optimal average assessing accuracy can be achieved with all 7 IRL algorithms. As the top $\mathcal{K}$ of cluster ranking o increases, the performance of C-NMI increases as well, which indicates that the larger the state space sampling, the better the performance of C-NMI.

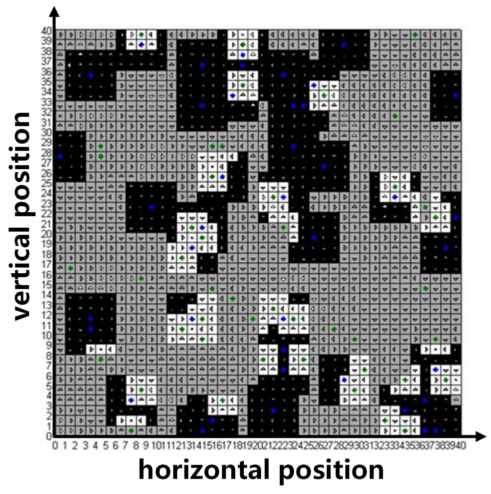
D. Robustness comparison

To validate the robustness of the different assessment metrics, we compare the performance of MRL, EVD, C-NMI, and $\omega_1 \cdot \text{MRL} + \omega_2 \cdot \text{EVD}$ under different demonstrations. The results are shown in Figure 6, in which we use a dotted red line to surround the smallest accuracy variance among different demonstrations. It can be seen that for all of the algorithms, the assessing accuracy of C-NMI varies in the smallest range. This indicates that C-NMI has the highest robustness compared to the other metrics. In other words, C-NMI will give a more stable high assessing accuracy for any $\mathcal{D} \in \boldsymbol{\mathcal{D}}$. One possible reason for this is that when $\mathcal{D}$ is mixed with manipulated trajectories, IRL algorithms may give "unbalanced" policies that learn well in some states but poorly in others. Our C-NMI samples multiple state spaces in the whole state space to calculate the reward clusters' similarity, which can resolve the problem of the "unbalanced" policy to some extent.

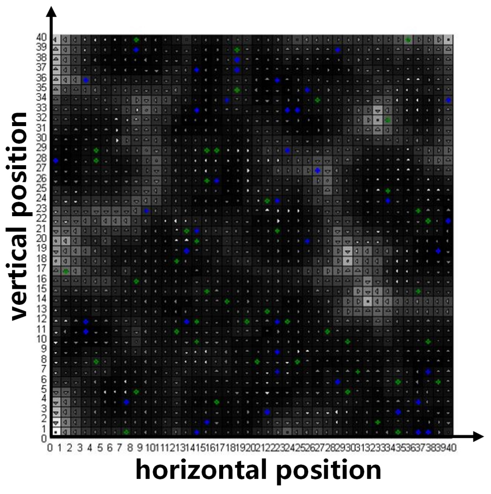
In addition, we also find that the robustness of MRL and EVD is not good enough. The reason for this is that MRL computes the mean reward loss on the global space, and thus it can not handle the "unbalanced" policy in extreme situations. For instance, the learned policy gives extremely close rewards in a few states, and very far rewards in others. Meanwhile, EVD may randomly sample initial states that locate in the well-learned space, and give a good assessment under the "unbalanced" policy.

VII. Conclusion

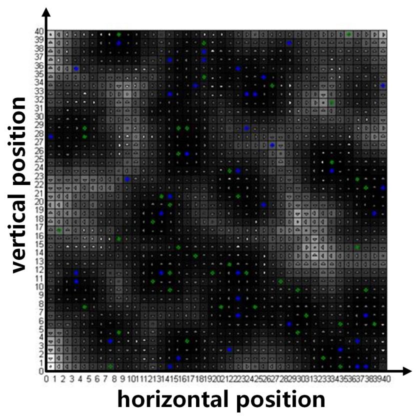
In this work, we pay attention to the inconformity problem of MRL- and EVD-based IRL assessment. There are two main challenges for us to address: (1) how to design a novel metric by combining the advantages of both MRL and EVD, and (2) how to construct a comprehensive assessment method for accurate comparison and analysis. To address such challenges, we craft a novel assessment of IRL based on the metric called normalized mutual information of reward clusters (C-NMI). Hence we attempt to fill the existing research gap by considering a middle-granularity state space between the entire state space and the certain sampled space. We utilize the agglomerative nesting algorithm (AGNES) to controlling dynamical C-NMI computing via a 4-order tensor model with injected manipulated trajectories. Furthermore, we give extensive experiments on seven mainstream IRLs. We analyze the experimental results on various aspects, including accuracy and robustness. The experimental results show that our method increases 110.13% and 116.59% respectively when compared with the EVD and MRL. Meanwhile, C-NMI is more robust than EVD and MRL under different demonstrations.



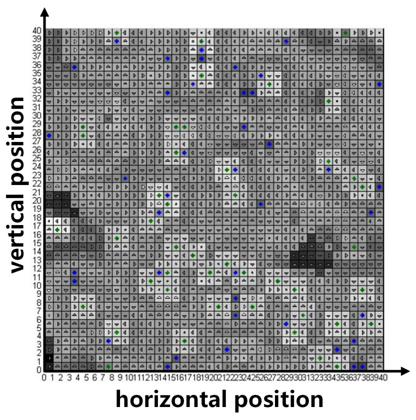
(a) Ground Truth



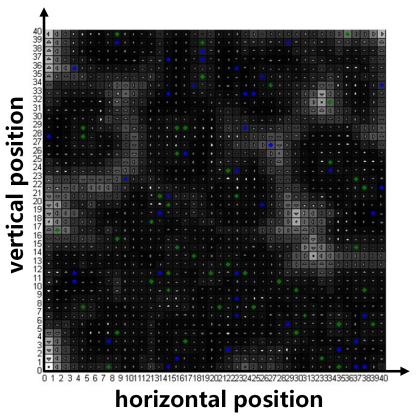
(b) MMP



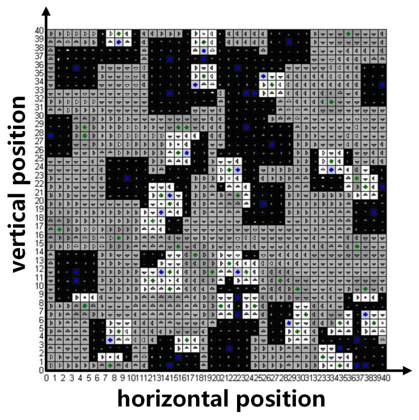
(c) MWAL



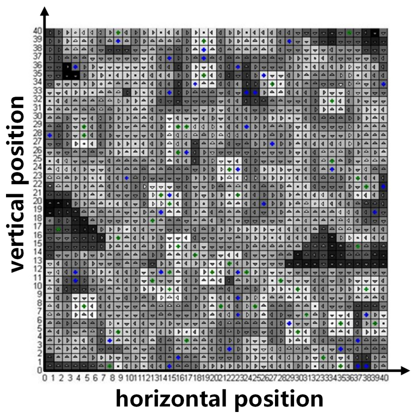
(d) MaxEnt



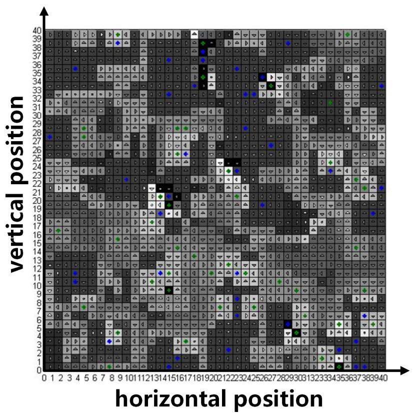
(e) AN



(f) GPIRL



(g) LEARCH



(h) FIRL

Fig. 5: Rewards visualization for the case study of (a) ground truth, (b) MMP, (c) MWAL, (d) MaxEnt, (e) AN, (f) GPIRL, (g) LEARCH, and (h) FIRL under a 40×40 grid map. For each subfigure the x-axis denotes the horizontal position of the given grid map, and the y-axis represents the vertical position of the given grid map. Moreover, for the 40×40 grid map, the range of x-axis and the y-axis are all in $\{0, 1, \dots, 39\}$.

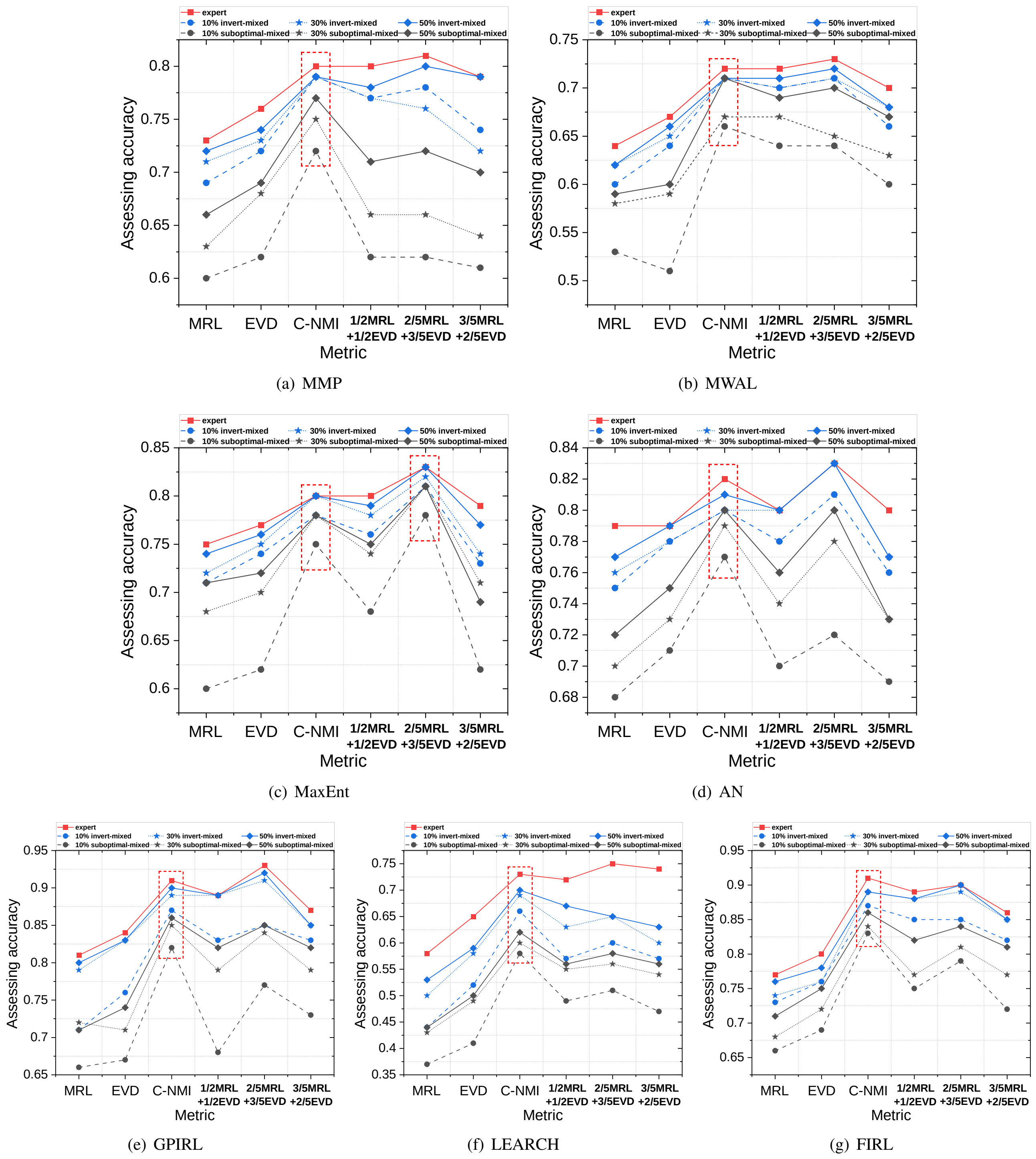


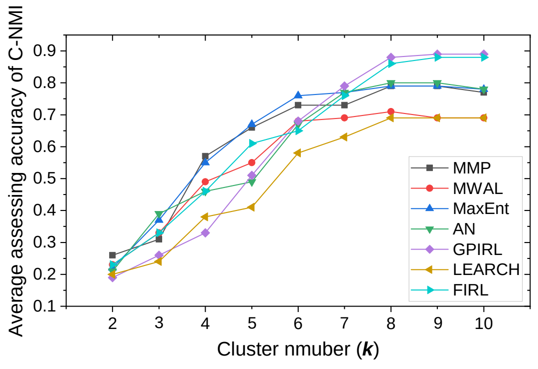
Fig. 6: Robustness comparison of different metrics for IRL algorithms MMP, MWAL, MaxEnt, AN, GPIRL, LEARCH, and FIRL.

TABLE V: Comparison of $\bar{\epsilon}_{metric}(A)$ for different metrics

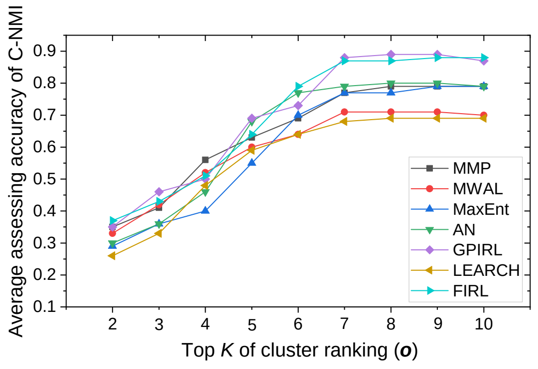
Metric	**IRL Algorithm**						
	MMP	MWAL	MaxEnt	AN	GPIRL	LEARCH	FIRL
MRL	0.69	0.61	0.72	0.75	0.77	<u>0.52</u>	0.73
EVD	0.72	0.64	0.74	0.77	0.80	0.59	0.76
C-NMI	0.79	0.71	0.79	0.80	**0.89**	0.69	0.88
$\frac{1}{2}$MRL+$\frac{1}{2}$EVD	0.76	0.70	0.77	0.78	0.86	0.65	0.85
$\frac{2}{5}$MRL+$\frac{3}{5}$EVD	0.77	0.71	0.78	0.81	**0.89**	0.67	0.87
$\frac{3}{5}$MRL+$\frac{2}{5}$EVD	0.74	0.68	0.75	0.77	0.84	<u>0.61</u>	0.83

TABLE VI: Comparison of the corresponding key metric values of the case study with the baseline under different IRL algorithms

Key Metric Value	**IRL Algorithm**						
	MMP	MWAL	MaxEnt	AN	GPIRL	LEARCH	FIRL
MRL	<u>19.3472</u>	8.9254	7.9420	<u>21.3749</u>	**1.6033**	10.0911	**4.3376**
EVD	<u>70.4981</u>	<u>61.2417</u>	**8.9814**	<u>80.7064</u>	**0.9338**	24.7630	**2.4415**
C-NMI	<u>1.6758</u>	<u>1.0037</u>	4.0712	<u>0.9984</u>	**9.9471**	5.8319	**8.0213**
Baseline	<u>-1</u>	<u>-1</u>	0	<u>-1</u>	**1**	0	**1**



(a) cluster number analysis



(b) top $\mathcal{K}$ cluster ranking analysis

Fig. 7: Comparison of $\bar{\epsilon}_{c-nmi}(A)$ under (a) various cluster number k, and (b) various top $\mathcal{K}$ of cluster ranking o.

REFERENCES

[18] Fabisch A, Petzoldt C, Otto M, et al. A Survey of Behavior Learning Applications in Robotics–State of the Art and Perspectives[J]. arXiv preprint arXiv:1906.01868, 2019.

[19] Silver D, Huang A, Maddison C J, et al. Mastering the game of Go with deep neural networks and tree search[J]. nature, 2016, 529(7587): 484-489.

[20] Tramèr F, Zhang F, Juels A, et al. Stealing Machine Learning Models via Prediction APIs[C]//25th USENIX security symposium (USENIX Security 16). 2016: 601-618.

[21] Wada K. Outliers in official statistics[J]. Japanese Journal of Statistics and Data Science, 2020, 3(2): 669-691.

[22] Kendall M G. The advanced theory of statistics[J]. The advanced theory of statistics., 1946 (2nd Ed).

[23] Xin L, Li S E, Wang P, et al. Accelerated inverse reinforcement learning with randomly pre-sampled policies for autonomous driving reward design[C]//2019 IEEE Intelligent Transportation Systems Conference (ITSC). IEEE, 2019: 2757-2764.

[24] Kokoska S, Zwillinger D. CRC standard probability and statistics tables and formulae[M]. Crc Press, 2000.

[25] Streit M, Gehlenborg N. Bar charts and box plots: creating a simple yet effective plot requires an understanding of data and tasks[J]. Nature methods, 2014, 11(2): 117-118.

[26] Gauci J, Conti E, Liang Y, et al. Horizon: Facebook's open source applied reinforcement learning platform[J]. arXiv preprint arXiv:1811.00260, 2018.

[27] Silver D, Huang A, Maddison C J, et al. Mastering the game of Go with deep neural networks and tree search[J]. nature, 2016, 529(7587): 484-489.

[28] Fabisch A, Petzoldt C, Otto M, et al. A Survey of Behavior Learning Applications in Robotics–State of the Art and Perspectives[J]. arXiv preprint arXiv:1906.01868, 2019.

[29] Wang B, Gong N Z. Stealing hyperparameters in machine learning[C]//2018 IEEE Symposium on Security and Privacy (SP). IEEE, 2018: 36-52.

[30] Li D C, He Y Q, Fu F. Nonlinear inverse reinforcement learning with mutual information and Gaussian process[C]//2014 IEEE International Conference on Robotics and Biomimetics (ROBIO 2014). IEEE, 2014: 1445-1450.

[31] Tramèr F, Zhang F, Juels A, et al. Stealing Machine Learning Models via Prediction APIs[C]//25th USENIX security symposium (USENIX Security 16). 2016: 601-618.

[32] Rousseeuw P J, Croux C. Explicit scale estimators with high breakdown point. In" L1 Statistical Analysis and Related Methods"(Y. Dodge, ed.)[J]. 1992.

[33] KOCAT$\ddot{u}$RK A, ALTUNKAYNAK B. A New Method Based on Interquartile Range to Feature Selection for Classification in Big Data[J].

[34] Ratner B. Pythagoras: Everyone knows his famous theorem, but not who discovered it 1000 years before him[J]. Journal of Targeting, Measurement and Analysis for Marketing, 2009, 17(3): 229-242.

[35] Huang R, Cui C, Sun W, et al. Poster: Is Euclidean Distance the best Distance Measurement for Adaptive Random Testing?[C]//2020 IEEE 13th International Conference on Software Testing, Validation and Verification (ICST). IEEE, 2020: 406-409.

[36] Ng A Y, Russell S J. Algorithms for inverse reinforcement learning[C]//Icml. 2000, 1: 2.
[37] Arora S, Doshi P. A survey of inverse reinforcement learning: Challenges, methods and progress[J]. Artificial Intelligence, 2021, 297: 103500.
[38] Dvijotham K, Todorov E. Inverse optimal control with linearly-solvable MDPs[C]//ICML. 2010.
[39] Ratliff N D, Silver D, Bagnell J A. Learning to search: Functional gradient techniques for imitation learning[J]. Autonomous Robots, 2009, 27(1): 25-53.
[40] Abbeel P, Ng A Y. Apprenticeship learning via inverse reinforcement learning[C]//Proceedings of the twenty-first international conference on Machine learning. 2004: 1.
[41] Levine S, Popovic Z, Koltun V. Feature construction for inverse reinforcement learning[J]. Advances in Neural Information Processing Systems, 2010, 23.
[42] Shiarlis K, Messias J, Whiteson S A. Inverse reinforcement learning from failure[J]. 2016.
[43] Wang X, Klabjan D. Competitive multi-agent inverse reinforcement learning with sub-optimal demonstrations[C]//International Conference on Machine Learning. PMLR, 2018: 5143-5151.
[44] Agarwal N, Singh K. The price of differential privacy for online learning[C]//International Conference on Machine Learning. PMLR, 2017: 32-40.
[45] Tossou A C Y, Dimitrakakis C. Achieving privacy in the adversarial multi-armed bandit[C]//Thirty-First AAAI Conference on Artificial Intelligence. 2017.
[46] Tossou A C Y, Dimitrakakis C. Algorithms for differentially private multi-armed bandits[C]//Thirtieth AAAI Conference on Artificial Intelligence. 2016.
[47] Sajed T, Sheffet O. An optimal private stochastic-mab algorithm based on optimal private stopping rule[C]//International Conference on Machine Learning. PMLR, 2019: 5579-5588.
[48] Shariff R, Sheffet O. Differentially private contextual linear bandits[J]. Advances in Neural Information Processing Systems, 2018, 31.
[49] Balle B, Gomrokchi M, Precup D. Differentially private policy evaluation[C]//International Conference on Machine Learning. PMLR, 2016: 2130-2138.
[50] Wang B, Hegde N. Private q-learning with functional noise in continuous spaces[J]. 2019.
[51] Kaufman L, Rousseeuw P J. Finding groups in data: an introduction to cluster analysis[M]. John Wiley & Sons, 2009.
[52] Shetty P, Singh S. Hierarchical Clustering: A Survey[J]. International Journal of Applied Research, 2021, 7(4): 178-181.
[53] Levine S, Popovic Z, Koltun V. Nonlinear inverse reinforcement learning with gaussian processes[J]. Advances in neural information processing systems, 2011, 24.
[54] Ziebart B D, Maas A L, Bagnell J A, et al. Maximum entropy inverse reinforcement learning[C]//Aaai. 2008, 8: 1433-1438.
[55] Ratliff N D, Bagnell J A, Zinkevich M A. Maximum margin planning[C]//Proceedings of the 23rd international conference on Machine learning. 2006: 729-736.
[56] Syed U, Schapire R E. A game-theoretic approach to apprenticeship learning[J]. Advances in neural information processing systems, 2007, 20.

Tong Chen received her M.S. degree in cyber security from Beijing Jiaotong University, Beijing, in 2018. She is currently a Ph.D. candidate of cyber security in Beijing Jiaotong University, Beijing. Her main research interests are cyber security and reinforcement learning security.

Jiqiang Liu received his B.S. and Ph.D. degrees in fundamental mathematics from Beijing Normal University, Beijing, in 1994 and 1999, respectively. He is currently a professor with Beijing Jiaotong University, Beijing. His main research interests are trusted computing, cryptographic protocols, privacy preserving, and network security.

Thar Baker (Senior Member, IEEE) received the Ph.D. degree in autonomic cloud applications from Liverpool John Moores University (LJMU), U.K., in 2010 and became a Senior Fellow of Higher Education Academy in 2018. He is an Associate Professor with the Department of Computer Science, University of Sharjah (UoS), UAE. Before joining UoS, he was a Reader of Cloud Engineering and the Head of the Applied Computing Research Group, Faculty of Engineering and Technology, LJMU. Prior to that, he was a Lecturer of Computer Science with the Department of Computing and Mathematics, Manchester Metropolitan University, U.K. He has published numerous refereed research papers in multidisciplinary research areas, including parallel and distributed computing, algorithm design, green and sustainable computing, and energy routing protocols.

Yalun Wu received Bachelor degree from Qingdao Agricultural University in 2017. He is currently a graduate student of Beijing Jiaotong University. His current research interests include AI security and information security.

Yingxiao Xiang received the BS degree in computer science and technology from Taiyuan University of Technology in 2016, Master degree from Beijing student of information security in Beijing Jiaotong University. Her research interests are in information security and AI security.

Yike Li received Bachelor degree from Hefei University of Technology in 2018. She is currently a graduate student of Beijing Jiaotong University. Her current research interests include AI security and Image generation.

Wenjia Niu received his B.S. degree in computer application from Beijing Jiaotong University, Beijing, in 2005, Ph.D. degree in computer software and theory from Chinese Academy of Sciences, Beijing, in 2010, both in computer science. Now he is currently a professor in Beijing Jiaotong University, Beijing. His research interests are AI security, agent and data mining.

Endong Tong received his Ph.D. degree in information and signal processing from Chinese Academy of Sciences, Beijing, in 2013. He is currently an assistant professor of Beijing Jiaotong University, Beijing. His current research interests include AI security services computing and data mining. He has published more than 30 research papers in refereed international conferences and journals.

Albert Y. Zomaya is currently the Chair Professor of High Performance Computing & Networking in the School of Computer Science, University of Sydney. He is also the Director of the Centre for Distributed and High Performance Computing which was established in late 2009. Professor Zomaya was an Australian Research Council Professorial Fellow during 2010-2014 and held the CISCO Systems Chair Professor of Internetworking during the period 2002–2007 and also was Head of school for 2006–2007 in the same school.