# Outlier-robust Inverse Reinforcement Learning and Reward-based Detection of Anomalous Driving Behaviors

Dan Li[1], Mohamad Louai Shehab[2], Zexiang Liu[1], Nikos Aréchiga[3], Jonathan DeCastro[3], and Necmiye Ozay[1,2]

*Abstract*— **This paper considers an outlier detection problem for a collection of vehicles or *agents*. These agents are represented by Markov decision processes and the trajectory data are assumed available. The work aims to learn the intentions or *reward functions* of agents, and infer the anomalous agents whose intentions differ from the majority. To achieve this, we propose a joint inverse reinforcement learning framework, which enables learning of a common reward function that captures the behavior of the majority as well as individual rewards for normal and abnormal agents. An example on the detection and analysis of driving behaviors is provided, demonstrating the effectiveness of the proposed framework.**

## I. Introduction

The detection of anomalous or outlying behaviors can be useful in a broad range of applications, including healthcare [1], [2], finance [3], robotics [4], and more recently, transportation [5]–[7]. Transportation systems, composed of many vehicles or *agents*, are usually safety-critical. Thanks to the development of smart infrastructures and devices, large amounts of vehicle-driving data are made accessible to learn the nominal driving behaviors as well as to identify interesting or outlying behaviors. Detecting such outlying behaviors at run-time can be important to avoid interruptions to the traffic flow. Offline detection and characterization of such behaviors can also be useful, for instance, in generating interesting test cases for training and testing autonomous driving software [8] or directly reasoning about driver intents at runtime [9]. *Reward functions* have been considered a succinct, expressive and generalizable representation to characterize the system dynamical behavior [10]–[13] and, therefore, intentions of drivers can be effectively distinguished in the space of reward functions. Inverse Reinforcement Learning (IRL) aims to learn the reward function of a particular driver via the observation of its driving behavior. By characterizing the behavior via the reward function, it becomes easier to determine which drivers differ substantially from others. Nevertheless, a representative reward function for behaviors of the majority can be hard to learn when outliers prevail. To address this issue, we aim to adapt IRL to learn a reward function which represents the behavior of the majority, and to detect potential outlier behaviors in the collection due to different identified rewards.

*Related work:* Inverse Reinforcement Learning (IRL) has been successful at representing intentions for a variety of purposes in vehicle driving and traffic modeling applications. In learning driving behaviors, the work in [11] used IRL in a simple, 3-lane car driving simulation where a human expert demonstrated trajectories of five different driving styles, and the resulting algorithm learned reward functions that qualitatively match the demonstrated driving styles. To deal with more complex driving scenarios characterized by large state spaces, the work of [14] improved this method by integrating Deep Q-Learning (DQN) in the inner loop of IRL and generated safe and sound trajectories for that scenario. Similar questions were also addressed via a Maximum-Entropy Deep IRL algorithm proposed in [15], where the authors modeled the vehicles using finite-state Markov Decision Processes (MDPs) and the resulting reward functions were validated by the generated policies in comparison to the demonstrated ones. To achieve a more reactive, dynamic driver intention-prediction, the work in [16] used a hierarchical IRL algorithm and validated the methodology on a ramp-merging scenario. Various other applications of IRL were also considered, e.g. in [17], [18], [19], [20]. In need of the resilience and security of transportation systems, anomaly detection has also been considered recently. In [21], authors used a Bayesian IRL approach to reason about anomalies in the driving data from a single vehicle. In contrast, our work aims to jointly learn a representative reward function for a collection of vehicles in a way that is robust to outliers, while simultaneously detecting the anomalous agents and learning their rewards. There are also techniques beyond IRL for anomaly detection, for instance using temporal logics [22], [23], and applications of anomaly detection in assessment of driving behaviors [9].

*Statement of contributions:* We propose a novel joint optimization problem, using a finite-state/action MDP agent model, to distinguish a majority of agents with similar behaviors from a smaller set of agents whose behaviors are noticeably different from the majority. This problem is shown to be equivalent to a mixed integer linear program, which can be relaxed to a standard linear program for efficiency. Moreover, we extend the framework to handle incomplete policy information, that can simply be deduced from agent trajectories. To the best of our knowledge, this is the first implementation of a joint optimization problem which simultaneously learns the individual rewards of each

Toyota Research Institute ("TRI") provided funds to assist the authors with their research but this article solely reflects the opinions and conclusions of its authors and not TRI or any other Toyota entity.

[1]D. Li, Z. Liu, and N. Ozay are with EECS, University of Michigan, Ann Arbor, MI 48109, USA {ecedanli;zexiang;necmiye}@umich.edu

[2]M.L. Shehab and N.Ozay are with the Robotics Institute, University of Michigan, Ann Arbor, MI 48109, USA {mlshehab}@umich.edu

[3] N. Aréchiga and J. DeCastro are with TRI, USA {nikos.arechiga,jonathan.decastro}@tri.global

agent, learns a representative reward for the majority, and identifies anomalies.

*Notation:* Let $\mathbb{R}$, $\mathbb{N}$, $\mathbb{R}^n$ and $\mathbb{R}^{m\times n}$ respectively denote the space of real numbers, natural numbers, $n$-dimensional vectors and $m$-by-$n$ matrices. For an $R \in \mathbb{R}^n$, we denote by $R^\top$ the transpose of $R$ and write $R > 0$ if all components are positive. We respectively denote 1-norm and infinity norm of the vector $R$ by $\|R\|_1$ and $\|R\|_\infty$, and we let $\|R\|_0$ denote the number of non-zero entries of $R$. For a $\mathcal{P} \in \mathbb{R}^{n\times n}$, we denote the value at an entry $(s, s')$ by $[\mathcal{P}]_{ss'}$, where $s$ and $s'$ are the row and column index, respectively. The cardinality of a finite set $\mathcal{N} \subset \mathbb{N}$ is written as $|\mathcal{N}|$. Specifically, the symbol $\mathbf{1}_n$ is a short-hand notation for the vector $(1, \dots, 1)^\top \in \mathbb{R}^n$ and $\mathbf{0}_n$ is that for $(0, \dots, 0)^\top \in \mathbb{R}^n$. Given a set $\mathcal{S} \subset \mathbb{R}^n$, we define an indicator function $\mathbb{1}_\mathcal{S} : \mathbb{R}^n \to \mathbb{R}$, where $\mathbb{1}_\mathcal{S}(s) = 1$ if $s \in \mathcal{S}$, or otherwise 0.

## II. Problem Formulation

Consider a collection of $N$ independent agents and denote by $\mathcal{N} := \{1, \dots, N\}$ their index set. We represent each agent $i \in \mathcal{N}$ by a finite-state and finite-action Markov Decision Process (MDP), denoted by a tuple $\text{MDP}_i := (\mathcal{S}, \mathcal{A}, \{\mathcal{P}_{sa}\}, R_i)$, where $\mathcal{S}$ and $\mathcal{A}$ respectively are the set of states and actions with the cardinality $|\mathcal{S}| = n$ and $|\mathcal{A}| = m$. The collection $\{\mathcal{P}_{sa}\}$ contains all state-transition probabilities from any state $s \in \mathcal{S}$ with any action $a \in \mathcal{A}$, and we define $\mathcal{P}_{sa}(s') := \mathbb{P}(s' \mid s,\, a)$ with $\mathbb{P}$ a conditional probability measure on $s' \in \mathcal{S}$. We assume that each agent $i$ holds a reward function $R_i : \mathcal{S} \to \mathbb{R}$ and operates under a stationary policy $\pi_i : \mathcal{S} \to \mathcal{A}$ which, for any $s_0 \in \mathcal{S}$, maximizes the expected, discounted and cumulative rewards

$$\max_{\pi_i} \left\{ V^\pi(s_0) := \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t R_i(s_t) | \pi, s_0] \right\},$$

where we call $V^\pi : \mathcal{S} \to \mathbb{R}$ the *value function* with the discount factor $\gamma \in (0, 1)$, and the expectation is taken over $s_{t+1} \sim \mathcal{P}_{s_t a_t}$ with the action selected as $a_t = \pi_i(s_t)$, $t = 0, 1, \dots$. Notice that each agent owns an *unknown* reward function as well as an optimal policy, which characterizes the behavior of the agent. In this work, we assume that the majority of the agents behave in a similar and *common* way, namely, they can be represented by similar, if not the same, reward functions. We denote the set of these agents by $\mathcal{M}^c \subset \mathcal{N}$ and, other than this, we denote by $\mathcal{M} = \mathcal{N} \setminus \mathcal{M}^c$ the set of *anomalous* agents who behave significantly different from the majority. In particular, we have $|\mathcal{M}^c| > |\mathcal{M}|$. To make it clear, let $\mathcal{V}$ be the space of reward functions and give a metric $d : \mathcal{V} \times \mathcal{V} \to \mathbb{R}$ for $\mathcal{V}$. We assume agents in $\mathcal{M}^c$ satisfy the following property.

**Assumption II.1 (Common behaviors are endowed with similar reward functions)** *There exists a representative reward $R_0$ and a threshold parameter $\epsilon > 0$ so that, for any agents $i \in \mathcal{M}^c$, $d(R_i, R_0) \leq \epsilon$. Furthermore, for any $j \in \mathcal{M}$, $d(R_j, R_0) > \epsilon$.*

Assumption II.1 distinguishes in the reward-function space agents with common behaviors from those with different goals. In this work, we aim to learn a representative reward for the normal agents in $\mathcal{M}^c$ and meanwhile detecting candidates in $\mathcal{M}$, which are outliers. To do this, we consider two problems as follows:

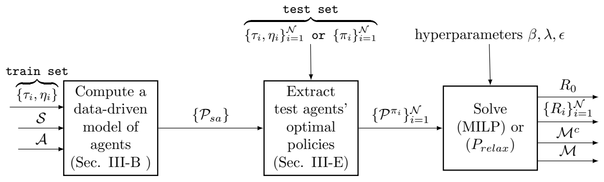

test set
$\{\tau_i, \eta_i\}_{i=1}^{\mathcal{N}}$ or $\{\pi_i\}_{i=1}^{\mathcal{N}}$
hyperparameters $\beta, \lambda, \epsilon$
train set
$\{\tau_i, \eta_i\}$
$\mathcal{S}$
$\mathcal{A}$
Compute a data-driven model of agents (Sec. III-B )
$\{\mathcal{P}_{sa}\}$
Extract test agents' optimal policies (Sec. III-E)
$\{\mathcal{P}^{\pi_i}\}_{i=1}^{\mathcal{N}}$
Solve (MILP) or $(P_{relax})$
$R_0$
$\{R_i\}_{i=1}^{\mathcal{N}}$
$\mathcal{M}^c$
$\mathcal{M}$

Fig. 1: Proposed anomaly detection framework

**Problem 1:** Given $\mathcal{S}$, $\mathcal{A}$, $\{\mathcal{P}_{sa}\}$ and policies $\{\pi_i\}_{i\in\mathcal{N}}$, find a representative reward $R_0$ and individual rewards $R_i$ such that for all $i \in \mathcal{M}$, $d(R_0, R_i) > \epsilon$ and for all $i \in \mathcal{M}^c$, $d(R_0, R_i) \leq \epsilon$ for some $\epsilon > 0$.

**Problem 2:** Given $\mathcal{S}$, $\mathcal{A}$, $\{\mathcal{P}_{sa}\}$, and some finite state and action trajectories $\{\tau_i := (s_0^{(i)}, s_1^{(i)}, \dots, s_{T_i}^{(i)}),\ \eta_i := (a_0^{(i)}, a_1^{(i)}, \dots, a_{T_i}^{(i)})\}_{i\in\mathcal{N}}$ for each agent $i$ obtained under optimal policies, find a representative reward $R_0$ and individual rewards $R_i$ such that for all $i \in \mathcal{M}$, $d(R_0, R_i) > \epsilon$ and for all $i \in \mathcal{M}^c$, $d(R_0, R_i) \leq \epsilon$ for some $\epsilon > 0$.

Notice that Problem 1 requires the knowledge of the policies while Problem 2 requires only state-action trajectories. A special case of Problem 2 is that all trajectories of agents are *complete*, i.e., for any agent $i \in \mathcal{N}$ and any $s \in \mathcal{S}$, there exists a time $t$ so that $s_t^{(i)} = s$. In such a scenario, two problems are identical. In the following, we start from a solution to Problem 1 and then further deal with the scenario when the sample trajectories $\{\tau_i, \eta_i\}$ are incomplete or the lengths $\{T_i\}_i$ are too small for policies to be fully recovered.

## III. Methodology

This section first presents our high-level approach, then provides the details of the proposed joint inverse reinforcement learning mechanism which learns the reward functions of agents as well as detects the anomalies.

### A. A General Framework

Our complete framework is demonstrated in Fig. 1. To solve Problems 1 and 2, our starting point is the availability of agent models. These models should be *well-defined* in the sense that the state space $\mathcal{S}$, action space $\mathcal{A}$, and the state-transition mappings $\{\mathcal{P}_{sa}\}$ together encode all possible behaviors independent of the reward. We build these models from training data according to section III-B. Given such models, test state trajectories $\{\tau_i\}$, as well as policies $\{\pi_i\}$ or action trajectories $\{\eta_i\}$, we obtain reward functions that these policies optimize, by solving a large-scale, joint Inverse Reinforcement Learning (IRL) problem. Then, the resulting representative reward function enables us to obtain a threshold-based anomaly detector, which identifies anomalies that belong to the set $\mathcal{M}$. Finally, we address information-fusion techniques when only incomplete trajectories are available. Such a scenario results in a similar and tractable optimization problem.

### B. The Data-driven Model of Agents

Suppose that the actual system has a state space $\mathcal{X}$ and an action space $\mathcal{U}$ that can contain infinitely many elements. To model the actual system by an MDP with $n$ states in $\mathcal{S}$ and $m$ actions in $\mathcal{A}$, we partition the state space $\mathcal{X}$ into $n$ subsets $\{\mathcal{X}_i\}_{i=1}^n$, where any state $x$ in $\mathcal{X}_i$ is mapped to the $i$th state of the MDP in $\mathcal{S}$. Similarly, the action space $\mathcal{U}$ is partitioned into $m$ subsets $\{\mathcal{U}_i\}_{i=1}^m$, where any action $u$ in $\mathcal{U}_i$ is mapped to the $i$th action in $\mathcal{A}$.

Assume that a set $\{(x_i, u_i, x_i')\}_{i=1}^p$ of sampled transitions of the actual system is available. Given the partitions $\{\mathcal{X}_i\}_{i=1}^n$ and $\{\mathcal{U}_i\}_{i=1}^m$, we map each transition $(x_i, u_i, x_i')$ of the actual system into a transition $(s_i, a_i, s_i')$ of the MDP. Then, for each tuple $(s, a, s')$ in $\mathcal{S} \times \mathcal{A} \times \mathcal{S}$, the state-transition probability $\mathcal{P}_{sa}(s')$ of the MDP is estimated by

$$\mathcal{P}_{sa}(s') = \frac{\mathcal{D}(s, a, s')}{\sum\limits_{s'' \in \mathcal{S}} \mathcal{D}(s, a, s'')},$$

where $\mathcal{D}(s, a, s')$ is the number of times that the transition $(s, a, s')$ occurs in the mapped data set $\{(s_i, a_i, s_i')\}_{i=1}^p$. To have a well-defined state-transition probability $\{\mathcal{P}_{sa}\}$, we require that the data set $\{(s_i, a_i, s_i')\}_{i=1}^p$ visits all the states with all possible actions for at least once.

### C. Learning a Representative Reward Function

To learn a representative reward function, first of all, we need to assume that *non-trivial*[1] reward functions of all agents exist. Notice that such an assumption depends on the selected models and the available trajectory data. For simplicity, let us first characterize the space of reward functions for a single agent following ideas in [10]. Then, we propose an optimization problem which learns a representative reward function for possibly all, common agents.

Consider an agent MDP $:= (\mathcal{S}, \mathcal{A}, \{\mathcal{P}_{sa}\}, R)$ with the optimal policy $\pi$, $|\mathcal{S}| = n$ and $|\mathcal{A}| = m$. By the Bellman's principle of optimality, the following relation [24] holds:

$$V^\pi(s) = R(s) + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{s\pi(s)}(s') V^\pi(s'),\ \forall s \in \mathcal{S},$$
$$V^\pi(s) \geq V^\mu(s),\ \forall s \in \mathcal{S},\ \forall \mu : \mathcal{S} \to \mathcal{A},$$

where $\gamma \in (0, 1) \subset \mathbb{R}$. With the abuse of notation, let us respectively denote by $V^\pi \in \mathbb{R}^n$ and $R \in \mathbb{R}^n$ the vector form of $V^\pi(s)$ and $R(s)$, $s \in \mathcal{S}$, where $V^\pi := [V^\pi(s_1), \ldots, V^\pi(s_n)]^\top$ and $R := [R(s_1), \ldots, R(s_n)]^\top$. In particular, we write the state-transition matrix under a policy $\mu$ as $\mathcal{P}^\mu \in \mathbb{R}^{n \times n}$, where, for an entry $(s, s')$, we select $[\mathcal{P}^\mu]_{ss'} := \mathcal{P}_{s\mu(s)}(s')$. As a result, we have

$$V^\pi = R + \gamma \mathcal{P}^\pi V^\pi \text{ and } V^\pi \geq V^\mu,\ \forall\ \mu. \qquad (1)$$

To derive an equivalent condition on the existence of the reward functions, we have two observations. First, the mapping between the reward function $R$ and the *optimal* value function $V^\pi$ is a bijection: $R = (I - \gamma \mathcal{P}^\pi) V^\pi$ because $\mathcal{P}^\pi$ is a stochastic matrix and $\gamma \in (0, 1)$, thus $(I - \gamma \mathcal{P}^\pi)$ is invertible. Then, the existence of $R$ is equivalent to that of $V^\pi$. Second, the inequality in (1) holds for any policy $\mu$. Now, consider a policy $\mu$ to be an one-step extension of $\pi$ at the initial state $s_0$, i.e. $\mu(s_0) \neq \pi(s_0)$ and $\mu(s_t) = \pi(s_t)$ for all $s_t \in \mathcal{S}$ and any $t > 0$. We consider such perturbations over all possible initial state $s_0 \in \mathcal{S}$, resulting in the expected reward of the policy $\mu$ being $V^\mu = R + \gamma \mathcal{P}^\mu V^\pi$. Consequently, a condition equivalent to the optimality is

$$(\mathcal{P}^\pi - \mathcal{P}^\mu) V^\pi \geq 0, \text{ with } \mu(s) \in \mathcal{A} \setminus \{\pi(s)\},\ \forall s \in \mathcal{S}.$$

Notice that there are at most $n(m-1)$ constraints. Next, let $X^\mu := \mathcal{P}^\pi - \mathcal{P}^\mu$ and recall that $X^\mu \mathbf{1}_n = \mathbf{0}_n$, $|[X^\mu]_{ss'}| \leq 1$. Therefore, finding a reward function is equivalent to finding a nonnegative[2] vector $z$ so that $X^\mu z \geq 0$ for all possible $\mu$. However, nontrivial solutions to $z$ may not exist for particular systems and optimal policies given, as showing in the following example.

**Example III.1 (Non-trivial reward functions may not exist)** *Consider a 2-dimensional MDP where $\mathcal{S} := \{s_1, s_2\}$ and $\mathcal{A} := \{\text{stay}, \text{switch}\}$. The action* stay *and* switch *respectively retain and change the state with the probability 1. Given that the optimal policy is* switch *for both state $s_1$ and $s_2$. Then, the only possible rewards are trivial ones.*

Furthermore, we show the closed form solutions to reward functions for a system with two states and a given optimal policy in the following lemma.

**Lemma III.1 (Explicit Reward functions of 2-dimensional systems)** *Given an MDP with $|\mathcal{S}| = 2$ and given a policy $\pi$ that optimizes a set $\mathcal{R}$ of unknown reward functions. Then, all reward functions $R \in \mathcal{R}$ can be written in the following form: $R = \alpha \mathbf{1}_2 - (I_2 - \gamma \mathcal{P}^\pi)[0, \Delta]^\top$, $\alpha \in \mathbb{R}$, with either $\Delta \geq 0$ or $\Delta \leq 0$.*

*Proof:* To find $R$, we equivalently aim to find $z := [z_1, z_2]^\top \in \mathbb{R}^2_{\geq 0}$ so that $X^\mu z \geq 0$ for all possible $\mu$, with $X^\mu$ in the following parameterized form

$$X^\mu := \mathcal{P}^\pi - \mathcal{P}^\mu = \begin{bmatrix} \lambda_1^\mu, & -\lambda_1^\mu \\ \lambda_2^\mu, & -\lambda_2^\mu \end{bmatrix}, \text{ for some } \lambda_1^\mu, \lambda_2^\mu \in [-1, 1],$$

where $(\lambda_1^\mu, \lambda_2^\mu)$ depends on $\pi$ and is parameterized by $\mu$. There are three possibilities for solutions to $\{X^\mu z \geq 0\}_\mu$: 1) all parameters in $\mathcal{Z} := \{\lambda_1^\mu, \lambda_2^\mu\}_\mu$ are nonnegative: 2) there exist $\eta_1$, $\eta_2 \in \mathcal{Z}$ which have different signs; and 3) all parameters in $\mathcal{Z}$ are negative. We know that each pair of parameters $(\lambda_1^\mu, \lambda_2^\mu)$ satisfies:

$$\lambda_1^\mu (z_1 - z_2) \geq 0,\ \lambda_2^\mu (z_1 - z_2) \geq 0.$$

For the case 1), we have $z_1 - z_2 \geq 0$. Now, consider solution $z = [\alpha, \alpha - \Delta]^\top$ with $\alpha \in \mathbb{R}$ and $\Delta \geq 0$. We find explicit reward functions, written as $R = \alpha(1 - \gamma)\mathbf{1}_2 - (I_2 - \gamma \mathcal{P}^\pi)[0, \Delta]^\top$. As $\alpha$ can be arbitrary, we achieve the form as in the lemma. For the case 2), since $\eta_1$, $\eta_2$ must have different signs, then $z = \alpha \mathbf{1}_2$, $\alpha \in \mathbb{R}$, resulting in trivial reward functions $R = \alpha(1 - \gamma)\mathbf{1}_2$. This is again in form of the solution as in the lemma. Case 3) achieves the same form, with $\Delta \leq 0$. ■

[1]A function is *non-trivial* if its image set is not a singleton.

[2]Notice that whenever $Xz \geq 0$, $X\hat{z} \geq 0$ for any $\hat{z} = z + \alpha \mathbf{1}_n$, $\alpha \in \mathbb{R}$.

Intuitively, Lemma III-C says that, with a given $\pi$, the number of reward functions optimized by $\pi$ is infinite and, all these rewards $R$ fall in one of the following three situation: 1) $R(s_1) > R(s_2)$, 2) $R(s_1) < R(s_2)$ and 3), $R(s_1) = R(s_2)$. Therefore, Example III.1 and Lemma III-C indicate that a given model with a policy does not necessarily guarantee existence of a nontrivial reward function. Further, even when the policy is optimal, it does not necessarily come from a unique reward function and, we can only learn the reward function of an agent in an equivalence class. Then, without loss of generality, we learn reward functions whose range is in the unit interval and, we characterize the set of feasible reward functions for an agent $i \in \mathcal{N}$ by

$$\mathcal{R}_i^\beta := \left\{ R : \mathcal{S} \to [0,1] \;\middle|\; \begin{array}{l} R = (I - \gamma \mathcal{P}^{\pi_i}) V^{\pi_i}, \\ (\mathcal{P}^{\pi_i} - \mathcal{P}^{\mu}) V^{\pi_i} \geq \beta, \\ \forall \mu(s) \in \mathcal{A} \setminus \{\pi_i(s)\},\ s \in \mathcal{S}, \end{array} \right\}, \quad (2)$$

where the nonnegative parameter $\beta \in \mathbb{R}^n$ quantifies the degree of the optimality compared to alternative policies. The larger the $\beta^\top \mathbf{1}_n$, the smaller the number of rewards in $\mathcal{R}_i^\beta$ and, the larger the difference between the optimal value function $V^\pi$ and the rest of value functions $\{V^\mu\}$. In particular, we write $\mathcal{R}_i := \mathcal{R}_i^0$, which contains all reward functions that satisfy the optimality condition, including those trivial ones.

In the following, we consider selecting a representative reward function $R_0$ that solves Problems 1 and 2. To do this, let us denote by $\ell : \mathbb{R}^{n(N+2)} \to \mathbb{R}$, $(R_0, R_1, \ldots, R_N, \beta) \mapsto \mathbb{R}$, the loss function which encodes the classification of the anomaly as well as the determination of $R_0$ so that it is close to all normal agents' reward functions. Then, we seek for solutions of an optimization problem, defined as follows:

$$\begin{aligned} \min_{R_0,\, R_i,\, i \in \mathcal{N}} \quad & \ell(R_0, R_1, \ldots, R_N, \beta), \\ \text{s.t.} \quad & R_i \in \mathcal{R}_i^\beta,\ \forall i \in \mathcal{N}, \end{aligned} \qquad \text{(P)}$$

where $\beta$ can be an *a-priori* and fixed margin, or a decision variable to be jointly optimized. In the following section, we propose several loss functions $\ell$ so that solutions to (P) are efficient and, in the mean time, our goal can be achieved.

## *D. Reward-based Behavior Classification and Detection*

To enable the behavior classification and anomaly detection, a possible loss function is the number of agents that cannot be represented by $R_0$. In other words, we consider the reward-distance function $L : \mathbb{R}^{n(N+1)} \to \mathbb{R}^N$, where

$$L(R_0, R_1, \ldots, R_N) := (d(R_0, R_1), d(R_0, R_2), \ldots, d(R_0, R_N)),$$

with $d$ a metric on the space of reward functions, and we select the loss function $\ell$ to be the cardinality function (also known as 0-norm) of $L$, $\ell := \|L\|_0$. The cardinality function promotes sparsity and aims to make as many $R_i$'s equal to $R_0$ as possible. Such a selection clusters similar rewards and meanwhile allows for the learning of the actual number of anomalous agents, i.e., an unknown value $|\mathcal{M}|$. Further, we would like to mention that the optimality margin $\beta$ plays a role in determining the representative reward function. To exclude trivial reward functions, we use $\beta > 0$ as a hyperparameter. Precisely, we consider $d(R_0, R_i) := \|R_0 - R_i\|_1$, $i \in \mathcal{N}$, and $\ell := \lambda/N \|L\|_0 - 1/n \beta^\top \mathbf{1}_n + 1/(nN) \sum_{i \in \mathcal{N}} \|R_i\|_1$ with a given hyper-parameter $\lambda > 0$. Notice that, in $\ell$, the first term encodes our goal which clusters agents that are similar; the second regulates optimality of the learned rewards; and the third promotes sparse rewards. Particularly, we choose large $\lambda$ if we solely emphasize on our goal of clustering. Consequently, we write Problem (P) in the following form:

$$\begin{aligned} \min_{\substack{\beta, R_0, \\ R_i, V^{\pi_i},\, i \in \mathcal{N}}} \quad & \frac{\lambda}{N} \|L\|_0 - \frac{1}{n} \beta^\top \mathbf{1}_n + \frac{1}{nN} \sum_{i \in \mathcal{N}} \|R_i\|_1, \\ \text{s.t.} \quad & R_i = (I - \gamma \mathcal{P}^{\pi_i}) V^{\pi_i},\ \forall i \in \mathcal{N}, \\ & (\mathcal{P}^{\pi_i} - \mathcal{P}^{\mu}) V^{\pi_i} \geq \beta, \\ & \forall \mu(s) \in \mathcal{A} \setminus \{\pi_i(s)\},\ s \in \mathcal{S},\ i \in \mathcal{N}, \\ & \beta \geq \mathbf{0}_n,\ 0 \leq R_i \leq 1,\ \forall i \in \mathcal{N}. \end{aligned} \qquad \text{(P)}$$

The selected cardinality function results in a discontinuous objective function, which renders Problem (P) intractable. In the following, we investigate an equivalent problem formulation so that solutions to (P) is the same as that to a mixed-integer linear program.

**Lemma III.2 (Mixed-integer formulation is exact)** *Let us introduce auxiliary variables* $\mathbf{x} \in \{0,1\}^N$, $\mathbf{y} \in \mathbb{R}^N$ *and* $\mathbf{z}_i \in \mathbb{R}^n$, $i \in \mathcal{N}$. *Then, the solutions to* (P) *are the same as those to the following problem:*

$$\begin{aligned} \min_{\substack{\mathbf{x}, \mathbf{y}, \beta, R_0, \\ R_i, V^{\pi_i}, \mathbf{z}_i,\, i \in \mathcal{N}}} \quad & \frac{\lambda}{N} \mathbf{x}^\top \mathbf{1}_N - \frac{1}{n} \beta^\top \mathbf{1}_n + \frac{1}{nN} \sum_{i \in \mathcal{N}} \|R_i\|_1, \\ \text{s.t.} \quad & \textit{all constraints in } \text{(P)},\ \beta \geq \mathbf{0}_n, \\ & R_0 - R_i \leq \mathbf{z}_i,\ \forall i \in \mathcal{N}, \\ & R_i - R_0 \leq \mathbf{z}_i,\ \forall i \in \mathcal{N}, \\ & [\mathbf{z}_1{}^\top \mathbf{1}_n, \mathbf{z}_2{}^\top \mathbf{1}_n, \ldots, \mathbf{z}_N{}^\top \mathbf{1}_n]^\top \leq \mathbf{y}, \\ & \mathbf{0}_N \leq \mathbf{y} \leq n\mathbf{x}, \quad \mathbf{x} \in \{0,1\}^N. \end{aligned} \qquad \text{(MILP)}$$

*Proof:* We achieve such a conclusion by equivalently reformulating the objective function of (P) using the definition of function $L$, metric $d$, and the binary, cardinality-indicator variable $\mathbf{x}$. ■

To find solutions of (P), Lemma III.2 indicates that we only need to solve Problem (MILP), which is a large-scale mixed-integer linear program. Notice that solving Problem (MILP) generally is NP-hard. Alternatively, we can consider a convex relaxation of (MILP) with $\mathbf{x}$ being a continuous variable in $[0,1]^N$. Then, the resulting problem becomes a linear program, which employs, in the original objective function $\ell$, $\|L\|_1$ in the place of $\|L\|_0$, and such a formulation can be efficiently solved in practice. Later, we use the convex relaxation for more efficient solutions.

Solutions to Problem (MILP) or its relaxation result in a representative reward function $R_0$ as well as rewards $\{R_i\}_{i \in \mathcal{N}}$ learned for each agents, which enable a threshold-based classifier for anomaly detection. Therefore, given a threshold $\epsilon > 0$, the set $\mathcal{M}$ of anomalous agents can be determined by identifying agent $i$ with $R_i$ satisfying $d(R_0, R_i) > \epsilon$, i.e., $\mathcal{M}^{\mathrm{c}} := \{i \in \mathcal{N} \mid d(R_0, R_i) \leq \epsilon\}$. Notice that the effectiveness of the detector depends on solutions to (MILP) as well as a proper threshold $\epsilon$ given.

### E. Scalable Information Fusion on Multiple-Agents Data

In this section, we consider how *a-priori* information of agents can be infused into Problem (MILP) or its relaxation.

The challenge of Problem 1 and 2 is the evaluation of constraints, especially the deduction of policy-related matrices $\mathcal{P}^{\pi_i}$, $i \in \mathcal{N}$. Notice that, in Problem 1, the agent model $\{\mathcal{P}_{sa}\}$ and policies $\{\pi_i\}_{i\in\mathcal{N}}$ are available, resulting in explicit formulas of matrices $\mathcal{P}^{\pi_i}$, $i \in \mathcal{N}$, with each entry $(s, s')$ being $[\mathcal{P}^{\pi_i}]_{ss'} := \mathcal{P}_{s\pi_i(s)}(s')$. Therefore, Problem 1 can be readily solved by a monolithic solution to Problem (MILP). In contrast, matrices $\{\mathcal{P}^{\pi_i}\}_{i\in\mathcal{N}}$ in Problem 2 must be estimated using agent-trajectory data $\{\tau_i, \eta_i\}_i$, or equivalently $\{(s_t^{(i)}, a_t^{(i)})\}_{i,t}$. Here, for any agent $i \in \mathcal{N}$, state $s \in \mathcal{S}$ and action $a \in \mathcal{A}$, we denote by $\mathcal{Q}_s^{(i)}(a) := \sum_t \mathbb{1}_{\{s\}}(s_t^{(i)})\mathbb{1}_{\{a\}}(a_t^{(i)})$ the total number of data points so that $(s_t^{(i)}, a_t^{(i)}) = (s, a)$ over all possible $t$. Then, the values $\{\mathcal{Q}_s^{(i)}(a)\}_a$ encode the *empirical* distribution of optimal actions of an agent $i$ at the state $s$. For simplicity, we select the policy to be

$$\pi_i(s) \in \operatorname*{argmax}_{a\in\mathcal{A}} \mathcal{Q}_s^{(i)}(a), \;\; i \in \mathcal{N}, \; s \in \mathcal{S}. \tag{3}$$

When all the states of agents have been visited in the trajectory data available, the above formula recovers matrices $\{\mathcal{P}^{\pi_i}\}_{i\in\mathcal{N}}$ and we solve Problem 2 via the solution to Problem (MILP) directly. Otherwise, there exists at least one agent $i$ which visited the state space partially. In such a scenario, let us denote by $\mathcal{S}_{\text{visited}}^{(i)}$ the set of visited states in the trajectory data $\{(s_t^{(i)}, a_t^{(i)})\}_t$ and we employ the policy-selection rule (3) for all agents over visited states. Because of the missing information on the unvisited states, reward functions must be learned partially. In the following, we propose the information fusion technique which learns a common reward function that combines trajectories from multiple agents. For an agent $i \in \mathcal{N}$, we define a state-selection matrix $H_i \in \mathbb{R}^{h_i\times n}$ where $h_i := |\mathcal{S}_{\text{visited}}^{(i)}|$, and rows of $H_i$ are the rows of the identity matrix $I_n$ whose indices are those of the elements in $\mathcal{S}_{\text{visited}}^{(i)}$. For example, given the set $\mathcal{S} = \{1, 2, 3, 4\}$ and $\mathcal{S}_{\text{visited}}^{(i)} = \{2, 4\}$, we have $H_i = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$. Then, each agent $i$ must have a reward in a relaxed set of $\mathcal{R}_i^{\beta}$ as follows:

$$\mathcal{R}_{i,\beta}^{\text{relax}} := \left\{ R : \mathcal{S} \to \mathbb{R} \;\middle|\; \begin{array}{l} H_iR = H_i(I - \gamma\mathcal{P}^{\pi_i})V^{\pi_i}, \\ H_i(\mathcal{P}^{\pi_i} - \mathcal{P}^{\mu})V^{\pi_i} \geq H_i\beta, \\ \forall\mu(s) \in \mathcal{A}\setminus\{\pi_i(s)\}, \; s \in \mathcal{S}_{\text{visited}}^{(i)}, \\ R \geq 0, 0 \leq V^{\pi_i} \leq 1, \end{array} \right\}.$$

Notice how the number of constraints reduces due to the missing information and how the bounds on the range of $R$ are moved to that of $V$ by leveraging the bijection map. As a consequence, unvisited state entries of $R$ become arbitrary in $[0, 1]$ and, each set $\mathcal{R}_{i,\beta}^{\text{relax}}$ provides the characterization of visited entries using the available data from agent $i$. Consider such information of all agents in $\mathcal{N}$, we learn a common reward $R_0$ by solving a variant of Problem (P) as follows:

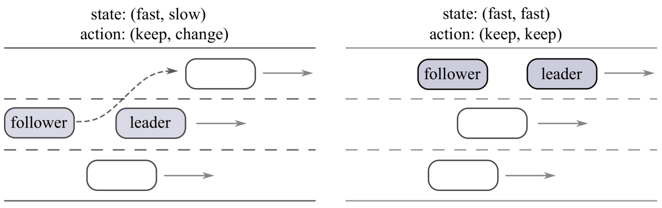


Fig. 2: A follower-centric driving behavior model.

$$\begin{aligned} \min_{\substack{\beta, R_0, \\ R_i, \; i\in\mathcal{N}}} \;& \frac{\lambda}{N}\|L\|_0 - \frac{1}{h_0}\mathbf{1}_{h_0}{}^{\top}H_0\beta + \frac{1}{nN}\sum_{i\in\mathcal{N}}\|R_i\|_1, \\ \text{s.t.} \;& R_i \in \mathcal{R}_{i,\beta}^{\text{relax}}, \; \forall i \in \mathcal{N}, \; \beta \geq \mathbf{0}_n, \end{aligned} \tag{P$_{\text{relax}}$}$$

where $H_0$ and $h_0$ respectively are the state-selection matrix and its rank corresponding to the set $\cup_{i\in\mathcal{N}}\mathcal{S}_{\text{visited}}^{(i)}$. Problem (P$_{\text{relax}}$) can be equivalently reformulated as a problem similar to (MILP), which provides us with a tractable formulation in case the trajectory data are incomplete or limited. In such a formulation, unvisited entries of the representative reward function $R_0$ will be rendered to zero. Furthermore, for each $i \in \mathcal{N}$, unvisited entries of $R_i$ will be equal to their corresponding entries in $R_0$. Consequently, solutions to Problem (P$_{\text{relax}}$) result in representative reward entries $H_0R_0$ as well as reward entries $\{H_iR_i\}_{i\in\mathcal{N}}$ learned for each agent, enabling a similar threshold-based classifier for anomaly detection. In words, given a threshold $\epsilon > 0$, the set $\mathcal{M}$ of anomalous agents can be determined by identifying agent $i$ with $R_i$ satisfying $d(H_iR_0, H_iR_i) > \epsilon$, where the metric $d$ is the $\infty$-norm on the reduced space. Then, we find $\mathcal{M}^{\text{c}} := \{i \in \mathcal{N} \mid \|H_iR_0 - H_iR_i\|_\infty \leq \epsilon\}$.

## IV. Application in Driving Behavior Analysis

In this section, we study vehicle interactions and employ the proposed framework for detecting anomalous driving behaviors. To characterize the vehicle interaction, we represent a *follower-leader* vehicle pair as an agent and consider the model of the agent as in Fig. 2. To analyze the proposed framework, we consider two case studies with and without known policies, corresponding to Problems 1 and 2, respectively. In particular, the first case study leverages a given MDP model and known agent policies, examines the equivalence class of rewards which can be learned, and analyzes the effectiveness of the proposed framework. In the second case study, we employ trajectory data collected from a simulator, use it to learn data-driven MDP models, and evaluate our framework on new data from the simulator. In such a scenario, actual agent rewards are unknown and most agent trajectories are incomplete, i.e., each trajectory visits only some subset of the states. Consequently, we focus on the empirical analysis of the solution.

### A. Case Study A

To study highway driving behaviors and test our framework in detecting anomalous agents, we define a 2-dimensional system state $s := (v_{\text{e}}, v_{\text{p}})$, where the first component is the speed of the *follower*, ego vehicle, and the second is that of the *leader*, preceding vehicle. For simplicity,

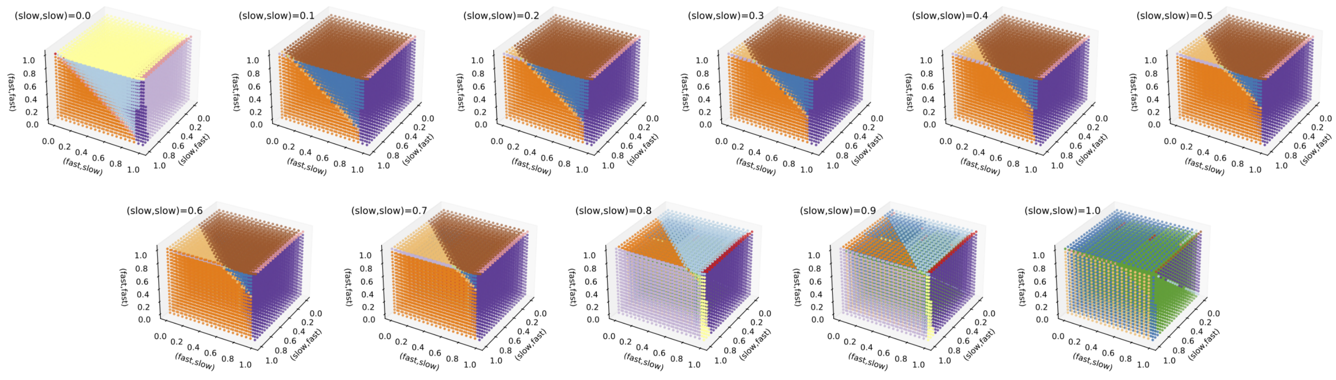


Fig. 3: Equivalence class of reward functions where each color represents a class that results in the identical optimal policy.

we consider discrete system states where each vehicle's speed is either fast or slow, and define $\mathcal{S} := \{\text{fast}, \text{slow}\}^2$, with $n := |\mathcal{S}| = 4$. In particular, a state $s = (\text{fast}, \text{slow})$ indicates the ego-vehicle speed $v_\text{e}$ being fast and the preceding-vehicle speed $v_\text{p}$ being slow. Furthermore, we assume that only the ego vehicle can be controlled and we select the action space to be a 2-dimensional vector $a := (a_\text{s}, a_\text{l})$, where the first component is the action related to speed and the second is that to lane position. Again, we consider a discrete action space and define $\mathcal{A} := \{\text{keep}, \text{change}\}^2$, with $m := |\mathcal{A}| = 4$. For instance, an action $a = (\text{change}, \text{keep})$ means changing the speed and keeping the lane position for the ego vehicle. Based on the selected state and action spaces, we propose an MDP model with the following transition probabilities given all possible actions $(a_1^\text{o}, a_2^\text{o}, a_3^\text{o}, a_4^\text{o})$:

$$\mathcal{P}^{a_1^\text{o}} = \begin{bmatrix} .99 & .01 & 0 & 0 \\ .01 & .98 & 0 & .01 \\ .01 & 0 & .98 & .01 \\ 0 & 0 & .01 & .99 \end{bmatrix}, \; \mathcal{P}^{a_2^\text{o}} = \begin{bmatrix} .5 & .5 & 0 & 0 \\ .01 & .99 & 0 & 0 \\ 0 & 0 & .5 & .5 \\ 0 & 0 & .5 & .5 \end{bmatrix},$$

$$\mathcal{P}^{a_3^\text{o}} = \begin{bmatrix} .01 & 0 & .98 & .01 \\ 0 & .01 & .01 & .98 \\ .98 & .01 & .01 & 0 \\ .01 & .98 & 0 & .01 \end{bmatrix}, \; \mathcal{P}^{a_4^\text{o}} = \begin{bmatrix} 0 & 0 & .5 & .5 \\ 0 & 0 & .5 & .5 \\ .5 & .5 & 0 & 0 \\ .01 & .99 & 0 & 0 \end{bmatrix},$$

where both row and column indices of matrices are states in the order $s_1^\text{o} = (\text{slow}, \text{slow})$, $s_2^\text{o} = (\text{slow}, \text{fast})$, $s_3^\text{o} = (\text{fast}, \text{slow})$, $s_4^\text{o} = (\text{fast}, \text{fast})$, and actions taken for those transition probabilities are $a_1^\text{o} = (\text{keep}, \text{keep})$, $a_2^\text{o} = (\text{keep}, \text{change})$, $a_3^\text{o} = (\text{change}, \text{keep})$, $a_4^\text{o} = (\text{change}, \text{change})$. Notice how these matrices are related to the agent model $\{\mathcal{P}_{sa}\}$ as described in Section III-C.

**Equivalence class of reward functions:** Given an MDP, there are many reward functions that would lead to the same optimal policy. To explore this, we uniformly sample $21^n$ non-trivial reward functions in the space $[0, 1]^n$ and derive optimal policies of these reward functions via the classical policy iteration approach [25]. Overall, there can be $n^m = 256$ possible policies and, among all these policies, only 24 policies are optimal for non-trivial reward functions sampled. Apparently, there exist policies which can not be optimal for any non-trivial reward functions and, a policy can be optimal for multiple reward functions. Such a result coincides with our observation in Example III.1 and, therefore, we must learn a representative reward from an equivalence class. To verify this, Fig. 3 visualizes the space of reward functions

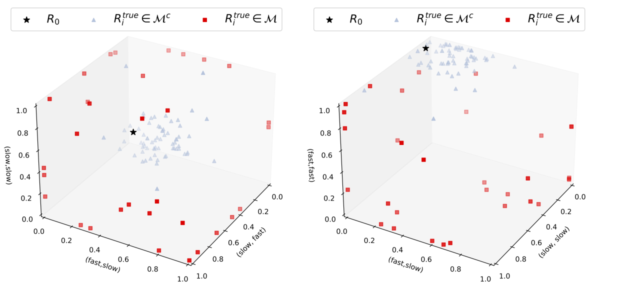


Fig. 4: Solution $R_0$, $\mathcal{M}$ returned by the proposed framework, with parameters $\lambda = 10$ and $\epsilon = 0.1$. The left figure is the learned rewards projected onto $(s_2^\text{o}, s_3^\text{o}, s_1^\text{o})$-space and the right is onto $(s_1^\text{o}, s_3^\text{o}, s_4^\text{o})$-space.

| | NC 1 | NC 2 | $\epsilon = 0.01$ | $\epsilon = 0.1$ | $\epsilon = 0.5$ |
|---|---|---|---|---|---|
| accuracy | 0.75 | 0.25 | 0.84 | 0.95 | 0.93 |
| $F_1$ | 0 | 0.39 | 0.82 | 0.9 | 0.87 |

TABLE I: Framework performance in Case A, compared to baseline naive classifiers NC1 and NC2.

$R \in \mathbb{R}^n$ where each point indicates a sampled reward function and the color indicates a particular optimal policy derived using that function. Notice that we sliced the four dimensional reward space over the dimension $(\text{slow}, \text{slow})$ for visualization purposes.

Next, we numerically evaluate the proposed framework on test data and demonstrate the solution to Problem 1 with the given agent policies and a threshold $\epsilon$.

**Test data:** We consider the anomaly detection Problem 1 given policies $\{\pi_i\}_{i \in \mathcal{N}}$ with $|\mathcal{N}| = 100$. These policies are obtained using policy-iteration from the known set of reward functions, denoted by $\{R_i^\text{true}\}_{i \in \mathcal{N}}$. We assume there roughly are $|\mathcal{M}^\text{c}| = 90, 80, 70, 60$ normal agents with $\{R_i^\text{true}\}_{i \in \mathcal{M}^\text{c}}$ being random rewards with the mean value $(0.1, 0, 0.2, 1)$. And the corresponding anomalous agents have rewards randomly sampled from the uniform distribution in $[0, 1]^n$. Notice that random anomalous agents can also take rewards close to those of normal agents. Finally, we emphasize that only policies are available for the proposed framework but the rewards are not.

**Anomaly detection:** We solve the linear programming relaxation of the problem (MILP) with $\lambda = 10$. We denote the obtained rewards by $(R_0, R_1, \ldots, R_N)$. Then, we select the detection threshold to be $\epsilon := 0.01, 0.1, 0.5$ and determine the set of abnormal agents by $\mathcal{M} := \{i \in \mathcal{N} \mid d(R_0, R_i) >$

$\epsilon\}$. As an example, Fig. 4 demonstrates the distribution of the actual rewards $\{R_i^{\text{true}}\}_{i\in\mathcal{N}}$ together with the learned representative reward $R_0$ with roughly 70 normal agents and $\epsilon = 0.1$. Notice that agents with the red color are classified as the anomaly, and those with the blue color are normal ones. For each $|\mathcal{M}^{\text{c}}| = 90, 80, 70, 60$, we run the method 10 times using the randomized data sets. Below are the results.

**Discussion:** We use the accuracy and $F_1$ score to measure the performance of the anomaly detection. Precisely, the accuracy is the ratio of correctly predicted samples to all samples and $F_1 := 2/(\text{precision}^{-1} + \text{recall}^{-1})$, where the precision is the ratio of correctly predicted anomalies to the total predicted anomalies and the recall is the ratio of correctly predicted anomalies to all actual anomalies. Intuitively, $F_1$ score balances the rates of false positive and false negative, and a high $F_1$ validates the effectiveness of the detection. To validate this, we use two very naive *zero-rule* detectors, where detector NC 1 classifies all as normal agents and detector NC 2 classifies all as anomalous agents. With various thresholds $\epsilon$, we compare the performance of our detector with that of the naive detectors in Table. I. These values are averaged over those measures in each scenario with $|\mathcal{M}^{\text{c}}| = 90, 80, 70, 60$. Notice that the proposed approach performs generally better than the baseline.

## B. Case Study B

Here, we use a vehicle simulator to generate data for the model construction as well as that for anomaly detection. The simulator is developed based on the Intelligent Driver Model [26], [27] as well as MOBIL [27], where we consider a three-lane highway with a collection of cars driving at the mean speed around 30 m/s. The data are generated by tuning car-following behavior as well as lane-change behavior in IDM and MOBIL, respectively. Except for the standard setting in [27], we generate normal and anomalous agents by changing the safe distance parameter $s_0$ in IDM and politeness parameter $p$ in MOBIL. Intuitively, $s_0$ is set an order of magnitude smaller for normal agents than for anomalous agents, while the distance between centers of $p$ values for these two group of agents is 0.25.

In this case, we have access to control actions of the ego vehicle as well as trajectories of all vehicles in the simulator. Based on this, we use a set $\{\mathcal{D}_i\}_{i=1}^{100}$ of 100 trajectories for the system-model estimation and provide a separate data set for anomaly detection.

**Data-based model:** To construct the model, we write each time step $i$ of the trajectory data as a tuple $(x_i, u_i, x_{i+1})$, where the state $x_i := (x_i^{\text{e}}, x_i^{\text{p}}, d_i)$ respectively consists of ego-vehicle speed, the *relative* speed between the leader and follower, and the headway distance; the action $u_i$ consists of a binary variable in $\{\text{keep}, \text{change}\}$ indicating if the vehicle performs a lane-change and a continuous variable in $\mathbb{R}$ indicating the acceleration of the vehicle. As discussed in Section III-B, we construct the MDP by partitioning the state space of the actual system into 27 rectangular grids, mapped to 27 states in $\mathcal{S}$. In particular, thresholds $\{27, 31\}(m/s)$, $\{-4, 4\}(m/s)$ and $\{50, 100\}(m)$ are used to partition components of the states, respectively. Furthermore, we manually select two thresholds $\pm 0.5m/s^2$ that maps the vehicle acceleration into three discrete actions $\{\text{accel}, \text{keep}, \text{decel}\}$. Here accel (or decel) indicates that the vehicle is accelerating (or decelerating), and keep indicates that the vehicle maintains the current speed with a near-zero acceleration in the given thresholds. Thus, the action space $\mathcal{U}$ is mapped to 6 actions in $\mathcal{A} := \{\text{keep}, \text{change}\} \times \{\text{accel}, \text{keep}, \text{decel}\}$. Notice how the state and action spaces of the MDP are extended from those in Case A, resulting in a more realistic model.

**Anomaly detection:** To evaluate our approach, we build a data pool of $|\mathcal{N}| = 50$ agents where the rate of anomalies, denoted by $r_{\text{a}}$, is selected to be 10%, 20%, 30%, 40%. We test the framework on various $|\mathcal{M}|$ or $r_{\text{a}}$ as well as the detection thresholds $\epsilon = 0.05, 0.1, 0.15$. For each given $r_{\text{a}}$ and $\epsilon$, we run our method 10 times with randomized data selections. The resulting detection measures $z$ are in Fig. 5.

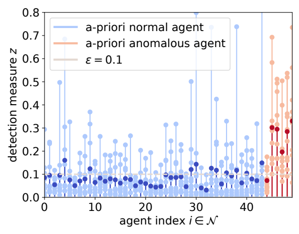
**Results:** We use the accuracy and $F_1$ score to measure the performance of the anomaly detection and these metrics are defined in the Case Study A. The results are again compared with two naive zero-rule detectors as well as two naive *sampling-based* detectors where the detector NC3 classifies each agent as normal or abnormal with the same probability and NC4 does that with the probability $r_{\text{a}}$ being anomalous. We also compare our results with k-Shape [28] and Kernel K-means (KKM) [29], which are unsupervised learning algorithms used in time-series clustering. The inputs to both of these methods are the raw driving trajectories and the desired number of clusters (2 in our case). Each method then outputs two clusters and assigns each driver to a respective cluster. We consider the cluster with the larger number of members to be the normal agents cluster and calculate the classification results accordingly. Results are shown in Table. II. Notice how the baseline contrasted with the proposed framework and how our approach trade off accuracy with $F_1$ score. Intuitively, the higher the $F_1$ score, the higher the ability to identify anomalies with that given threshold $\epsilon$ and anomaly rate $r_{\text{a}}$. To further verify the performance, Fig. 5 provides all agents' detection measures $z$ which are respectively associated with the parameter $r_{\text{a}} :=$ 10%, 20%, 30%, and 40%. In the figure, each line is one single test, where the $x$-axis is the index of agents and $y$-axis is the detection measure $z$ of that agent. Measures on the actual, human-classified normal agents are colored blue and that on anomalous are red. In particular, the averaged detection measure is highlighted in figures. It can be seen that, the lower the threshold $\epsilon$, the lower the precision but the higher the recall. Furthermore, anomalous agents tend to have uniformly-higher detection measures and as the ratio of anomalies increases, the detection ability decrease if the threshold is fixed. In practice, the threshold parameter $\epsilon$ can be tuned to optimize the performance of the framework.

## V. Conclusions

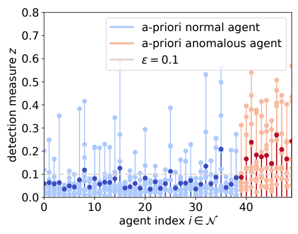
A novel, reward-based detection framework is proposed to identify agents with anomalous behaviors. The proposed framework identifies anomalies in the given data set and provides a representative reward which is robust w.r.t. outliers. An application in detecting abnormal driving behavior

| (accuracy, $F_1$) | NC 1 | NC 2 | NC 3 | NC 4 | k-shape | KKM | $\epsilon = 0.05$ | $\epsilon = 0.1$ | $\epsilon = 0.15$ |
|---|---|---|---|---|---|---|---|---|---|
| $r_a = 0.1$ | (0.90, 0) | (0.10, 0.18) | (0.50, 0.17) | (0.82, 0.10) | (0.88, 0.02) | (0.57, 0.18) | (0.56, 0.34) | (0.78, 0.49) | (0.85, 0.53) |
| $r_a = 0.2$ | (0.80, 0) | (0.20, 0.33) | (0.50, 0.29) | (0.68, 0.20) | (0.80, 0) | (0.54, 0.27) | (0.64, 0.52) | (0.83, 0.58) | (0.81, 0.44) |
| $r_a = 0.3$ | (0.70, 0) | (0.30, 0.46) | (0.50, 0.37) | (0.58, 0.30) | (0.72, 0.16) | (0.53, 0.36) | (0.62, 0.57) | (0.73, 0.46) | (0.74, 0.35) |
| $r_a = 0.4$ | (0.60, 0) | (0.40, 0.57) | (0.50, 0.45) | (0.52, 0.40) | (0.60, 0.03) | (0.56, 0.47) | (0.55, 0.61) | (0.59, 0.49) | (0.62, 0.32) |
| avg. rates | (0.75, 0) | (0.25, 0.39) | (0.5, 0.32) | (0.65, 0.25) | (0.75, 0.05) | (0.55, 0.32) | (0.59, 0.51) | (0.73, 0.51) | (0.76, 0.41) |

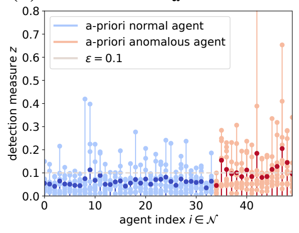
TABLE II: Framework performance in Case B, compared to baselines NC1, NC2, NC3, NC4, k-shape and KKM.



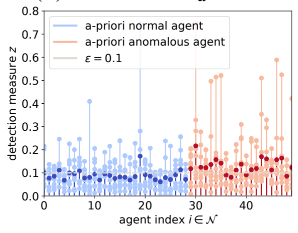
(a) case with $r_a = 10\%$



(b) case with $r_a = 20\%$



(c) case with $r_a = 30\%$



(d) case with $r_a = 40\%$

Fig. 5: Statistics of the detection measure $z$. Each dot is the measure of a single tested agent, except that the highlighted dot is the averaged measure over 10 tests for that agent index. The blue portion are measures of the labeled normal agents and the red are those of anomalous ones.

is studied in detail. Future work will focus on the extension of the framework to online detection scenarios and incorporating logical structure in addition to rewards to improve interpretability. We are also interested in using the learned rewards (both for normal and for anomalous agents) for downstream tasks like generating test scenarios for evaluating autonomous driving algorithms. Another direction is to develop similar anomaly detection techniques by adopting more recent IRL methods. This will allow for capturing rich feature-based rewards and expanding to the continuous domains.

## References

[1] A. Ukil, S. Bandyoapdhyay, C. Puri, and A. Pal, "IoT healthcare analytics: The importance of anomaly detection," in *IEEE Conf. on advanced information networking and applications*, 2016, pp. 994–997.

[2] M. Fahim and A. Sillitti, "Anomaly detection, analysis and prediction techniques in IoT environment: A systematic literature review," *IEEE Access*, vol. 7, pp. 81 664–81 681, 2019.

[3] M. Toledano, I. Cohen, Y. Ben-Simhon, and I. Tadeski, "Real-time anomaly detection system for time series at scale," in *Workshop on Anomaly Detection in Finance*, 2018, pp. 56–65.

[4] A. Bezemskij, G. Loukas, R. Anthony, and D. Gan, "Behaviour-based anomaly detection of cyber-physical attacks on a robotic vehicle," in *Int. Conf. on ubiquitous computing and communications and Int. symposium on cyberspace and security*, 2016, pp. 61–68.

[5] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys*, vol. 41, no. 3, pp. 1–58, 2009.

[6] F. V. Wyk, Y. Wang, A. Khojandi, and N. Masoud, "Real-time sensor anomaly detection and identification in automated vehicles," *T-ITS*, vol. 21, no. 3, pp. 1264–1276, 2019.

[7] M. Riveiro, M. Lebram, and M. Elmer, "Anomaly detection for road traffic: A visual analytics framework," *T-ITS*, vol. 18, no. 8, pp. 2260–2270, 2017.

[8] D. Zhao, H. Lam, H. Peng, S. Bao, D. J. LeBlanc, K. Nobukawa, and C. S. Pan, "Accelerated evaluation of automated vehicles safety in lane-change scenarios based on importance sampling techniques," *T-ITS*, vol. 18, no. 3, pp. 595–607, 2016.

[9] C. Yang, A. Renzaglia, A. Paigwar, C. Laugier, and D. Wang, "Driving behavior assessment and anomaly detection for intelligent vehicles," in *IEEE Int. Conf. on Cybernetics and Intelligent Systems and IEEE Conf. on Robotics, Automation and Mechatronics*, 2019, pp. 524–529.

[10] A. Y. Ng and S. J. Russell, "Algorithms for inverse reinforcement learning," in *ICML*, vol. 1, 2000, p. 2.

[11] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *ICML*, 2004, p. 1.

[12] J. Fu, K. Luo, and S. Levine, "Learning robust rewards with adversarial inverse reinforcement learning," *ICLR*, 2018.

[13] D. Vasquez, B. Okal, and K. Arras, "Inverse reinforcement learning algorithms and features for robot navigation in crowds: an experimental comparison," in *IROS*, 2014, pp. 1341–1346.

[14] S. Sharifzadeh, I. Chiotellis, R. Triebel, and D. Cremers, "Learning to drive using inverse reinforcement learning and deep q-networks," *NeurIPS*, 2016.

[15] C. You, J. Lu, D. Filev, and P. Tsiotras, "Advanced planning for autonomous vehicles using reinforcement learning and deep inverse reinforcement learning," *Robotics and Autonomous Systems*, vol. 114, pp. 1–18, 2019.

[16] L. Sun, W. Zhan, and M. Tomizuka, "Probabilistic prediction of interactive driving behavior via hierarchical inverse reinforcement learning," in *ITSC*, 2018, pp. 2111–2117.

[17] Q. Zou, H. Li, and R. Zhang, "Inverse reinforcement learning via neural network in driver behavior modeling," in *IEEE Intelligent Vehicles Symposium (IV)*, 2018, pp. 1245–1250.

[18] Z. Wu, L. Sun, W. Zhan, C. Yang, and M. Tomizuka, "Efficient sampling-based maximum entropy inverse reinforcement learning with application to autonomous driving," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5355–5362, 2020.

[19] S. Rosbach, V. James, S. Großjohann, S. Homoceanu, and S. Roth, "Driving with style: Inverse reinforcement learning in general-purpose planning for automated driving," in *IROS*, 2019, pp. 2658–2665.

[20] M. Wulfmeier, D. Rao, D. Z. Wang, P. Ondruska, and I. Posner, "Large-scale cost function learning for path planning using deep inverse reinforcement learning," *The Int. Journal of Robotics Research*, vol. 36, no. 10, pp. 1073–1087, 2017.

[21] M. H. Oh and G. Iyengar, "Sequential anomaly detection using inverse reinforcement learning," in *ACM SIGKDD Int. Conf. on Knowledge Discovery & data mining*, 2019, pp. 1480–1490.

[22] C. Yoo and C. Belta, "Rich time series classification using temporal logic," in *Robotics: Science and Systems XIII, MIT*, 2017.

[23] Z. Kong, A. Jones, and C. Belta, "Temporal logics for learning and detection of anomalous behavior," *IEEE Transactions on Automatic Control*, vol. 62, no. 3, pp. 1210–1222, 2017.

[24] R. Bellman, "Dynamic programming," *Science*, vol. 153, no. 3731, pp. 34–37, 1966.

[25] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[26] A. Kesting, M. Treiber, and D. Helbing, "Enhanced intelligent driver model to access the impact of driving strategies on traffic capacity," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 368, no. 1928, pp. 4585–4605, 2010.

[27] M. Treiber and A. Kesting, "Traffic flow dynamics," *Traffic Flow Dynamics: Data, Models and Simulation, Springer-Verlag Berlin Heidelberg*, pp. 983–1000, 2013.

[28] J. Paparrizos and L. Gravano, "k-shape: Efficient and accurate clustering of time series," in *ACM SIGMOD Int. Conf. on management of data*, 2015, pp. 1855–1870.

[29] I. S. Dhillon, Y. Guan, and B. Kulis, "Kernel k-means: spectral clustering and normalized cuts," in *ACM SIGKDD Int. Conf. on Knowledge discovery and data mining*, 2004, pp. 551–556.