Unveiling Taxi Drivers' Strategies via cGAIL

— Conditional Generative Adversarial Imitation Learning

Xin Zhang
Worcester Polytechnic Institute
xzhang17@wpi.edu

Yanhua Li
Worcester Polytechnic Institute
yli15@wpi.edu

Xun Zhou
University of Iowa
xun-zhou@uiowa.edu

Jun Luo
Lenovo Group Limited
jluo1@lenovo.com

***Abstract*—Smart passenger-seeking strategies employed by taxi drivers contribute not only to drivers' incomes, but also higher quality of service passengers received. Therefore, understanding taxi drivers' behaviors and learning the good passenger-seeking strategies are crucial to boost taxi drivers' well-being and public transportation quality of service. However, we observe that drivers' preferences of choosing which area to find the next passenger are diverse and dynamic across locations and drivers. It is hard to learn the location-dependent preferences given the partial data (i.e., an individual driver's trajectory may not cover all locations). In this paper, we make the first attempt to develop conditional generative adversarial imitation learning (cGAIL) model, as a unifying collective inverse reinforcement learning framework that learns the driver's decision-making preferences and policies by transferring knowledge across taxi driver agents and across locations. Our evaluation results on three months of taxi GPS trajectory data in Shenzhen, China, demonstrate that the driver's preferences and policies learned from cGAIL are on average 34.7% more accurate than those learned from other state-of-the-art baseline approaches.**

***Index Terms*—Urban Computing, Inverse Reinforcement Learning, Generative Adversarial Imitation Learning**

I. Introduction

Taxi service plays an important role in the public transportation systems and is an indispensable part for modern life. It not only provides a convenient way of transportation, but also creates a large number of jobs that support many drivers' families. Therefore, improving taxi operation efficiency is both a public management matter that imposes influences on the urban transportation and a business problem for each taxi driver. In the traditional taxi operation model when a taxi is vacant, the taxi driver is making a sequence of decisions on which directions to go to find the next passengers. A taxi driver may consider various factors when making such decisions, for example, the traffic condition and estimated travel demand in the surrounding areas, given the current location and time. Moreover, different drivers are likely to have different preferences over these decision-making factors, which ultimately lead to divergent business efficiencies and income levels. Hence, it is valuable to unveil the good strategies from those expert taxi drivers, and by sharing such knowledge, to boost taxi driver's business efficiencies and public transportation quality.

Inverse reinforcement learning (IRL) [1]–[7] is a typical solution to characterize such unique decision-making preferences of individual drivers. IRL learns a preference vector

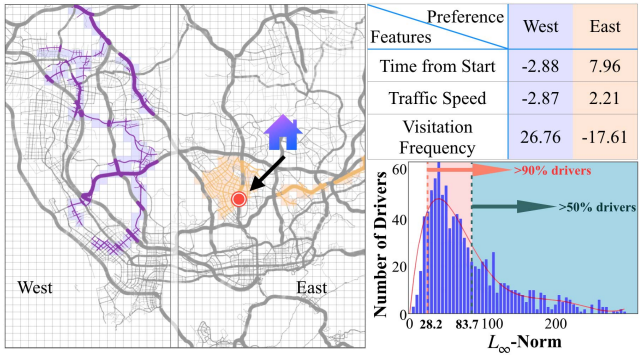

Fig. 1: Diverse driver preferences across regions.

to represent the significance of each factor to the driver. It is commonly assumed that the learned preference vector by IRL is inherent to the taxi driver and invariant across different geographical regions. Therefore, it can be used to estimate the decision-making policy of the driver in any region.

However, we found through analysis on real taxi GPS trajectory data that this is not true. The preference vectors of taxi drivers hinge significantly over different locations. Fig 1 shows the trajectory coverage of a selected taxi driver in Shenzhen, China. The driver's home location is marked on the map. We use MaxEnt IRL [4] approach to learn a preference vector based on the driver's GPS trajectories from the west and the east part of Shenzhen respectively. Three decision-making features were considered, including the time from work started (i.e., working duration), traffic speed (indicating traffic condition), visitation frequency (indicating the popularity) of the surrounding area of the current location. The table on the top-right suggests that the same driver exhibits drastically different preferences towards the same factors while driving in the two different sides of the city: When the driver is on the east part (downtown), she prefers a longer working time (i.e., close to the end of a day's work) to be close to home, regions with higher driving speed to avoid traffic, and less popular areas to escape downtown congestion. However the preferences are the opposite when she is working in the west part of the city where lies the rural areas.

The above phenomenon are common in taxi trajectory data, where the histogram in Fig 1 shows that most (90%) drivers have significant preference difference (in L_∞-norm) across locations. Hence, in reality, the human (driver) agents' prefer-

2374-8486/19/$31.00 ©2019 IEEE
DOI 10.1109/ICDM.2019.00194

ences are dynamic and dependent on geographic locations. Assuming such preferences spatially invariant makes the results of IRL less accurate and might lead to infeasible policies being generated. Alternatively, a better solution is to learn **location-dependent** preferences of each driver. Unfortunately, this task is hard for traditional IRL approaches [1]–[6] because the data for each driver might only cover part of the city, making it hard to infer the driver's preferences in the rest of the areas.

In this paper we tackle the above challenge and propose a novel solution. We formulate the passenger-seeking problem as a Markov Decision Process (MDP) and extract various decision-making features that the drivers evaluate when making decisions, such as travel demand and traffic speed (Sec IV). Our observation is that all the taxi drivers (as a group) would have significantly higher data coverage over geographical regions compared to an individual driver, and many taxi drivers share common decision-making preferences. Built upon this observation, we make the first attempt to develop a novel conditional generative imitation learning (cGAIL) model to collectively and inversely learn the driver's decision-making preferences and policies by transferring knowledge across taxi driver agents and across locations (Sec V). We validate our framework using a unique dataset from Shenzhen, China, with three months of taxi GPS trajectory data. Results demonstrate that the policies learned from cGAIL are on average 34.7% more accurate than those learned from other state-of-the-art baseline approaches (Sec VI).

II. Overview

In this section, we introduce our dataset, define *collective inverse preference learning* problem, and outline the solution framework.

A. Data Description

We use two datasets for our study, including (1) taxi trajectory data and (2) road map data. For consistency, all these datasets are aligned with the same time period.

Taxi trajectory data. We use taxi trajectory dataset in July, August and September, 2016 in Shenzhen, China. This dataset contains GPS records from 17,877 unique GPS-set-equipped taxis. Each of these taxis generates GPS records in roughly every 30 seconds. Every GPS record holds five attributes, including a unique plate ID, longitude, latitude, time stamp and passenger indicator. The passenger indicator is a binary value with 1 indicating a passenger on board, and 0 otherwise.

Road map data. The road map data of Shenzhen is obtained from OpenStreetMap [8], covering an area from 22.44°N to 22.87°N in latitude and from 113.75°E to 114.65°E in longitude with $455,944$ road segments.

B. Problem Definition and Solution Framework

We denote each driver as d, and the set of all drivers as $\mathcal{D}$. Taxis equipped with GPS sets generate GPS records over time. Each GPS point p consists of a location in latitude lat and longitude lng, and a time stamp t, i.e., $p = \langle lat, lng, t \rangle$. Below, we define a trajectory of a taxi composed of GPS records.

Definition 1 (**Trajectory** tr). A trajectory tr is a sequence of GPS points when the taxi is vacant and the driver is looking for passengers, denoted as $tr = \{p_1, \cdots, p_{n_0}\}$ (n_0 is the length of trajectory tr). Each taxi driver d has a collection of GPS trajectories over time. We denote the set of trajectories generated by a driver $d \in \mathcal{D}$ as Tr_d.

Note that we focus on each drivers' "seeking" trajectories which capture sequences of decisions made by the taxi driver on which direction a to go from the current state s (i.e., where the taxi is and what time it is in a day) to look for passengers. Hence, the taxi driver's passenger-seeking strategy can be characterized by two inherent functions with driver (defined below): *(i) reward function* and *(ii) policy function*.

Definition 2 (**Reward function** R). Given the current state (e.g., location and time of day) s, the driver of a vacant taxi chooses an action a (e.g., go east or west) based on her own evaluation of the expected reward (e.g., revenue in the next hour) of such a move. Denote such a function as $R(s, a|d)$ for $d \in \mathcal{D}$.

Such a reward function (in general a non-linear function) governs which direction a the driver will follow for the intrinsic pursuit of a higher reward over time. Each driver's reward function might be unique due to different knowledge and driving habits. The underlying patterns of direction choice is characterized as a driver policy function as defined below.

Definition 3 (**Policy function** π). A policy function $\pi(a|s, d)$ of a taxi driver $d \in \mathcal{D}$ characterizes the probability distribution for d to choose action a given the current state s.

Here again, an action is a driver's driving behavior such as driving towards a particular direction, and we denote the set of all possible actions as $\mathcal{A}$. Given a driver d and a state s, $\pi(\cdot|s, d)$ gives the likelihood over all actions $a \in \mathcal{A}$ that the target driver is likely to take.

Now we are ready to formally define our problem as below.

Collective inverse preference learning problem. Given *trajectories* Tr_d collected from a group of taxi drivers $\mathcal{D} = \{d\}$, we aim to learn a unifying model to inversely and jointly learn the policy $\pi(a|s, d)$ and reward function $R(s, a|d)$ for all drivers $d \in \mathcal{D}$.

Challenges. This problem is challenging in two aspects: i) a driver's reward and policy functions are location dependent (as observed in Fig 1). Therefore it is challenging to recover the two functions for areas without the target driver's demonstration data; ii) drivers possess diverse reward and policy functions, thus how to develop a unifying model to capture individual driver's reward and policy functions precisely is challenging.

Solution Framework. Our proposed solution to tackle the two challenges and solve the proposed collective inverse preference learning problem consists of three main components: *Stage 1 - data preparation*, *Stage 2 - data-driven modeling*, and *Stage 3 - conditional inverse preference learning* which are detailed in Sec III, IV and V respectively.

III. Stage 1 - data preparation

A. Map & Time Standardization and Trajectory Aggregation

Map griding. For the ease of analyzing taxi drivers' decision-making behaviors, we partition the city into small equal side-length grid cells [9], [10] with pre-defined side-length $b = 0.01°$. It leads to $1,934$ grid cells connected by road network. We denote each grid cell as g_i, with $1 \leq i \leq 1,934$, and the complete grid cell set as $\mathcal{G} = \{g_i\}$.

Time quantization. We further divide the time in a day into five-minutes intervals, i.e., 288 time slots a day, denoted as $\mathcal{I} = \{\tilde{t}_j\}$, with $1 \leq j \leq 288$.

Trajectory Aggregation. A combination of a grid cell g_i, time interval $\tilde{t}_j$, and the day of a week day, uniquely defines a spatio-temporal state, or state in short. Each GPS record $p = \langle lat, lng, t\rangle$ can thus be represented as an aggregated state $s = \langle g, \tilde{t}, day\rangle$, where the location $(lat, lng) \in g$, the time stamp $t \in \tilde{t}$, and day indicates the day of the week. Similarly, we can aggregate taxi trajectories into state level sequences. Each of taxi driver d's trajectories $tr \in Tr_d$ defined in section II-B can then be mapped as sequences of spatio-temporal states s, and the set of d's trajectories can be denoted by $\mathcal{T}_d$:

$$\tau = \{s_1, \cdots, s_{n'}\},\ \mathcal{T}_d = \{\tau_1, \cdots, \tau_{m_d}\}, \tag{1}$$

where n' is the length of a trajectory in states, and m_d is the number of trajectories of driver d.

B. Decision-Making Feature Extraction

Taxi drivers consider various factors (features) of the current "state" (i.e., where the taxi is and what time it is in a day), when making decisions of which direction to go to look for passengers. In this section, we extract and summarize all such features (denoted as a feature vector $\mathbf{f}$) into two categories below, namely, *state features* $\mathbf{f_s}$ and *condition features* $\mathbf{f_c}$. Clearly, $\mathbf{f} = [\mathbf{f_s}, \mathbf{f_c}]$. All of the state and condition features were extracted from historical taxi GPS trajectory data from 07/2016 to 09/2016 in Shenzhen, China.

State features $\mathbf{f_s}$. When a taxi driver d is at a certain state $s = \langle g, \tilde{t}, day\rangle$, the driver considers a list of features associated with the state s to make a decision, including three categories ($\mathbf{f_s} = [\mathbf{f_T}, \mathbf{f_M}, \mathbf{f_D}]$) as traffic features $\mathbf{f_T}$, temporal features $\mathbf{f_M}$ and PoI distance features $\mathbf{f_D}$ detailed below.

Traffic features ($\mathbf{f_T}$): This category include four features representing the traffic status of the state s from the historical data, including travel demand $f_{T,1}$, traffic volume $f_{T,2}$, traffic speed $f_{T,3}$, and waiting time $f_{T,4}$.

Temporal features ($\mathbf{f_M}$): This category includes the time of the day $f_{M,1}$ and the day of the week $f_{M,2}$ for the target state s.

Distance to places of interests (PoIs) ($\mathbf{f_D}$): There are 23 features $[f_{D,1}, \cdots, f_{D,23}]$ in this category, which characterize the distances in kilometers from the location of state s to 23 places of interests in Shenzhen, including 5 train stations, 1 airport, 5 popular shopping malls, 8 ports and checking points, and 4 major hospitals.

Condition features $\mathbf{f_c}$. Condition features $\mathbf{f_c}$ consist of four driver-related features serving as driver identity and a location identifier. Each driver is identified by his/her home location, working schedule and experience. A location identifier is a target grid cell g.

Home location ($f_{c,1}$): This feature characterizes the distance in kilometers from the current state location to the driver's home location, indicating the driver's preferences to work closer vs far away from home.

Working schedule ($f_{c,2}$ and $f_{c,3}$): This feature consists of time differences of current state s from the driver's average starting time and to the ending time, indicating the driver's working schedule.

Familiarity ($f_{c,4}$): This feature captures the average visitations of the driver to the current state s from the historical data, indicating how familiar the driver is to this particular region.

Location identifier (ℓ): Each location is a specific grid $g \in \mathcal{G}$ in the partitioned road map of the city.

IV. Stage 2 - data-driven modeling

Taxi drivers make a sequence of decisions on which direction to go to find the next passenger. In this section, we elaborate on how to model taxi drivers' decision-making processes as MDPs.

We consider each taxi driver as an "agent". When looking for passengers, the driver keeps evaluating various features in surrounding areas of the current state s, based on which the driver decides which direction to go to find the passengers. This whole process consisting of a sequence of decisions from the driver forms a trajectory. Each taxi driver aims to maximize the total received "reward" along the trajectory. As a result, the driver's passenger-seeking process can be naturally modelled as an MDP. Below, we explain how each component in an MDP is mapped and extracted from taxi trajectory data.

Agent: Each taxi driver d is considered as a unique agent. Different drivers have different reward functions.

State set $\mathcal{S}$: Each state $s \in \mathcal{S}$ is a spatio-temporal region, denoted as $\langle g, \tilde{t}, day\rangle$ as illustrated in Sec III. Map griding partitions the road map into $1,934$ grid cells, and each day is divided into 288 5-minutes intervals with seven days a week. As a result, the state space size is $1,934 \times 288 \times 7 = 3,898,944$.

Action set $\mathcal{A}$: An action $a \in \mathcal{A}$ denotes a direction to go when looking for passengers. We consider nine actions that an agent can take, including moving to one of the eight neighboring grid cells as an action, and staying at the current action.

Transition probability function $P : \mathcal{S} \times \mathcal{A} \mapsto [0, 1]$: Clearly, transitions in this MDP are deterministic, namely, an action will surely lead the agent to the corresponding next grid cell.

Reward $R : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$: A reward function $R(s, a)$ measures the reward a driver obtains by taking a direction (action) a from state s. Since a driver agent aims to maximize the total expected reward, the reward function governs how the driver chooses the next directions to go to. $R(s, a)$ is in general a non-linear function of the features associated with the surrounding regions of state s. In our study, $R(s, a)$ is unknown and is to be learned from the driver's historical trajectory data.

Policy function $\pi : \mathcal{S} \times \mathcal{A} \mapsto [0,1]$**:** A policy function $\pi(a|s,d)$ defines the probability of choosing a direction action $a \in \mathcal{A}$ at the current state s. Taking the features of a state s and the driver id d as input, a policy function randomly outputs a direction $a \in \mathcal{A}$ from the driver's policy distribution. In our study, the policy function (as a non-linear function in general) is to be learned from the driver's trajectories.

V. Stage 3 - conditional generative adversarial imitation learning

With the MDP modeling for taxi driver decision-making process, we are in a position to investigate how we may learn the policy and reward functions of each individual driver (agent) from their demonstrated trajectory data, with which we can further quantify and predict their passenger-seeking behaviors accurately. To achieve this goal, we need to answer two questions: *Q1 (Reward/Policy Function Learning):* For each individual driver agent, how to inversely learn the reward/policy function from the demonstrated trajectory data? *Q2 (Function Transferability across Locations and Agents):* How to learn the reward/policy functions for agents that are transferable across locations and agents?

To answer *Q1*, we introduce the state-of-the-art generative adversarial imitation learning, GAIL in Sec V-A. For *Q2*, we develop a novel conditional generative adversarial imitation learning, cGAIL, in Sec V-B. The proposed cGAIL model is a unifying inverse learning model that allows knowledge transfer across taxi driver agents and across locations.

A. Learning Reward/Policy functions with GAIL

User choice modeling has been extensively studied to learn human agents' decision-making reward and policy functions [4], [6], [11]–[13], where imitation learning methods like GAIL [6] learns general non-linear reward function. Therefore, we briefly introduce GAIL, and highlight its limitations on the transferability across locations and agents. Built upon these approaches, we will propose our cGAIL model in Sec V-B.

GAIL extends IRL by a non-linear reward function $R(s,a)$, and a non-linear policy function $\pi(a|s)$ both using deep neural networks. It introduces a regularizer function $\psi(R)$ to avoid overfitting, which leads to eq.(2)[1],

$$\max_R \psi(R) + \big(\min_\pi -H(\pi) - \mathbb{E}_\pi[R(s,a)]\big) + \mathbb{E}_{\pi_E}[R(s,a)]. \tag{2}$$

It was proven in [6] that when the function $\psi(R)$ is properly chosen, the dual problem of eq.(2) is equivalent to minimizing the Jensen-Shannon (JS) divergence between the trajectory distribution induced by obtained π and empirical π_E (from $\mathcal{T}$), namely, eq.(2) becomes[2]

$$\begin{aligned} &\min_\pi \quad -\lambda H(\pi) + D_{JS}(\pi, \pi_E), \quad \text{with} \\ &D_{JS}(\pi,\pi_E) = \max_R \mathbb{E}_{\pi_E}[\ln(R(s,a))] + \mathbb{E}_\pi[\ln(1-R(s,a))], \end{aligned} \tag{3}$$

[1]Note that in eq.(1) in [6], authors use cost function $c(s,a) : \mathcal{S} \times \mathcal{A} \mapsto (0,1)$ (indicating the cost of taking (s,a)). We in this work use reward $R(s,a)$, equivalent to $R(s,a) = 1 - c(s,a)$ for clarity.

[2]Please refer to [6] for detailed proof.

with $\lambda \geq 0$ as the Lagrangian multiplier introduced in deriving the IRL dual problem [6]. Clearly, $D_{JS}(\pi, \pi_E)$ is the JS-divergence. As a result, The problem in eq.(3) can be tackled using generative adversarial networks (GAN) model [14], where the policy function $\pi(a|s)$ and reward function $R(s,a)$ are the generator network and discriminator network, respectively. Hence, GAIL model applies to each individual driver agent to extract the policy and reward function. Given that the driver's reward function is location dependent in Fig 1, GAIL cannot model the reward function on locations where the driver have never visited from the demonstrated trajectory data. Moreover, for each individual driver agent, a separate GAN model needs to be trained, thus no knowledge is shared across driver agents. To tackle these problems (namely, answering *Q2*), we proposed a novel conditional generative adversarial imitation learning (cGAIL) model below.

B. Conditional Generative Adversarial Imitation Learning

There are two ideas behind cGAIL design: First, each individual driver agent covers partly the state (spatio-temporal regions) and action (directions to go) space in the underlying MDP, but the trajectories from all driver agents collectively provide a better coverage of states and actions; Second, driver agents share commonalities of their reward functions, e.g., some drivers may possess similar reward functions due to their common profiles (in ages, home locations, etc), thus their trajectories can be reused to infer reward functions of each other. To summarize, i) knowledge learned from trajectories of different driver agents is transferable across driver agents (referred to as *agent transferability*); ii) knowledge learned from trajectories in different geographical regions is transferable across locations (referred to as *location transferability*). In this section, we will develop conditional generative adversarial imitation learning (cGAIL), a unifying collective inverse reward learning framework to characterize drivers' rewards and policies by transferring knowledge across trajectories from various locations and driver agents.

To distinguish the locations and driver agents, we define the condition variable (vector) as a list of condition features (as defined in Sec III-B), i.e., $\mathbf{c} = \mathbf{f_c} = [f_{c,1}, f_{c,2}, f_{c,3}, f_{c,4}, \ell]$. The inverse reinforcement learning problem in eq.(2) was defined for a single agent and without location dependency, which can be extended to the following format to characterize location and agent transferabilities by considering it as a minmax game under condition $\mathbf{c}$,

$$\begin{aligned} &\max_R \min_\pi \quad -\lambda H(\pi(\cdot|\mathbf{c})) + \mathbb{E}_{\pi_E}[\ln(R(s,a|\mathbf{c}))] \\ &+ \mathbb{E}_\pi[\ln(1-R(s,a|\mathbf{c}))] + \mathbb{E}_{\pi_E}[\ln(1-R(s,a|\mathbf{c}'))], \end{aligned} \tag{4}$$

where the policy net (as the generator) π generates an action a for an input state s given a condition $\mathbf{c}$, such that (s,a) looks "real", i.e., as if generated by the given driver agent and location (defined in $\mathbf{c}$). Moreover, the reward net (as the discriminator) R increases the rewards for (s,a)'s from policy π_E with the condition $\mathbf{c}$, lowers down the rewards for (s,a)'s generated from π with the condition $\mathbf{c}$, and also decreases the rewards for (s,a)'s from expert policy π_E, but by an

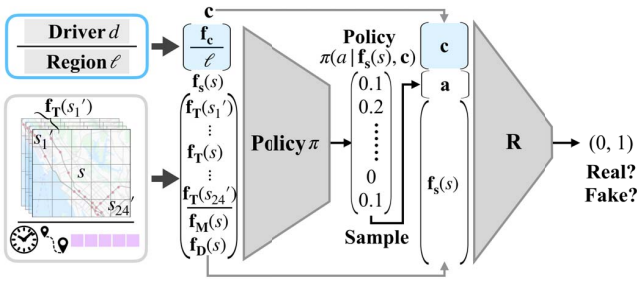

Fig. 2: cGAIL model structure.

inconsistent condition (in driver and/or location) $\mathbf{c}'$. Below, we detail the policy net π and reward net R, and the training algorithm for the proposed cGAIL model.

Policy network π (Generator): The policy net π takes condition features $\mathbf{c} = \mathbf{f_c}$ as input, indicating the target driver agent and the target location (grid cell) ℓ. Moreover, the input state features for policy net π include three parts: (1) The traffic features $\mathbf{f_T} = [f_{T,1}, \cdots, f_{T,4}]$ of the current state s (at location ℓ) and all 24 neighboring grid cells in ℓ's 5×5 neighborhood, $\mathcal{N}(s) = \{s_1', \cdots, s_{24}'\}$, denoted as $[\mathbf{f_T}(s), \mathbf{f_T}(s_1'), \cdots, \mathbf{f_T}(s_{24}')]$; (2) Temporal features of the current state s, $\mathbf{f_M}(s) = [f_{M,1}(s), f_{M,2}(s)]$, and (3) POI distance features of the current state s, $\mathbf{f_D}(s) = [f_{D,1}(s), \cdots, f_{D,23}(s)]$ as defined in Sec III-B.

As a result, the input state features for s form a feature vector $\mathbf{f_s}(s) = [\mathbf{f_T}(s), \mathbf{f_T}(s_1'), \cdots, \mathbf{f_T}(s_{24}'), \mathbf{f_M}(s), \mathbf{f_D}(s)]$ with length of 125. The output of policy net π is a distribution $\pi(\cdot|s)$ indicating the probabilities of choosing nine actions. These actions will be randomly chosen based on $\pi(\cdot|s)$. Fig 2 illustrates the input and output of the policy net. Since the input traffic features $\mathbf{f_T}$ cover the 5×5 neighborhood of the target state s, which can be viewed as a local traffic map, we employ convolutional neural network [15] as the network structure for policy net.

Reward network R (Discriminator): The reward network R takes the same condition features $\mathbf{c}$ and state features $\mathbf{f_s}(s)$ from policy net, and the policy net output action a as input. It outputs scalars within $[0, 1]$, indicating the reward value of a state-action pair (s, a). Similar to policy net, we employ convolutional neural network for the reward network R.

cGAIL training algorithm: Alg 1 illustrates the detailed process to train our proposed cGAIL model. During the training process, we apply batch gradient descent approach to update the policy network π and reward network R, with a predefined K (i.e., the total number of epochs). The taxi driver's trajectories $\mathcal{T}_d$'s (as defined in Sec III-A) can be broken down into n individual triples in state features, action, and condition features, thus forming a training set for cGAIL as $\mathcal{T} = \{(\mathbf{f_s}(s_1), a_1, \mathbf{c}_1), \cdots, (\mathbf{f_s}(s_n), a_n, \mathbf{c}_n)\}$. During each epoch $1 \le i \le K$, we sample a batch of m real data points, as $\mathcal{T}_i = \{(\mathbf{f_s}(s_1^i), a_1^i, \mathbf{c}_1^i), \cdots, (\mathbf{f_s}(s_m^i), a_m^i, \mathbf{c}_m^i)\} \subset \mathcal{T}$ from the training set (Line 2). Then, we input the state and condition features in $\mathcal{T}_i$ into policy network π to generate actions $\tilde{a}$, to construct a generated sample set denoted as $\tilde{\mathcal{T}}_i = \{(\mathbf{f_s}(s_1^i), \tilde{a}_1^i, \mathbf{c}_1^i), \cdots, (\mathbf{f_s}(s_m^i), \tilde{a}_m^i, \mathbf{c}_m^i)\}$ (Line 3). Moreover, we replace the condition features in $\mathcal{T}_i$ with randomly sampled condition features from $\mathcal{T}$ to construct triples with real state-action pairs coupled with mismatched conditions, i.e., $\hat{\mathcal{T}}_i = \{(\mathbf{f_s}(s_1^i), a_1^i, \hat{\mathbf{c}}_1^i), \cdots, (\mathbf{f_s}(s_m^i), a_m^i, \hat{\mathbf{c}}_m^i)\}$ (Line 4). Then, the reward network parameters $\boldsymbol{\theta}_R$ are updated (Line 5) by eq.(6) to maximize $\tilde{V}_R$ in eq.(5), with step size η_R.

$$\tilde{V}_R = \frac{1}{m}\sum_{j=1}^{m}\Big(\ln\left(R(\mathbf{f_s}(s_j^i), a_j^i|\mathbf{c}_j^i)\right) + \ln\left(1 - R(\mathbf{f_s}(s_j^i), \tilde{a}_j^i|\mathbf{c}_j^i)\right) + \ln\left(1 - R(\mathbf{f_s}(s_j^i), a_j^i|\hat{\mathbf{c}}_j^i)\right)\Big), \tag{5}$$

$$\boldsymbol{\theta}_R = \boldsymbol{\theta}_R + \eta_R \nabla_{\boldsymbol{\theta}_R} \tilde{V}_R. \tag{6}$$

Next, we update policy network parameters $\boldsymbol{\theta}_\pi$ by eq.(7) to minimize $\tilde{V}_\pi$ below, with η_π as the step size (Line 6).

$$\tilde{V}_\pi = \sum_{j=1}^{m}\Big(\frac{1}{m}\ln\left(1 - R(\mathbf{f_s}(s_j^i), \tilde{a}_j^i|\mathbf{c}_j^i)\right) - \lambda H(\pi(\mathbf{f_s}(s_j^i)|\mathbf{c}_j^i))\Big),$$

$$\boldsymbol{\theta}_\pi = \boldsymbol{\theta}_\pi + \eta_\pi \nabla_{\boldsymbol{\theta}_\pi} \tilde{V}_\pi. \tag{7}$$

Algorithm 1 cGAIL Training Process

Input: Taxi drivers' decision-making data as state-action-condition pairs $\mathcal{T} = \{(\mathbf{f_s}(s), a, \mathbf{c})\}$. Initialize parameter vectors $\boldsymbol{\theta}_\pi$ and $\boldsymbol{\theta}_R$ for policy net and reward net, respectively;

Output: Resulting $\boldsymbol{\theta}_\pi$ and $\boldsymbol{\theta}_R$.

1: **for** Each Epoch $1 \le i \le K$ **do**
2: Sample $\mathcal{T}_i \subset \mathcal{T}$;
3: Generate $\tilde{\mathcal{T}}_i$ from policy net π;
4: Sample/construct $\hat{\mathcal{T}}_i$ from $\mathcal{T}$;
5: Update $\boldsymbol{\theta}_\pi$ with Eq.6;
6: Update $\boldsymbol{\theta}_R$ with Eq.7;
7: **end for**

VI. Evaluations

We use three months taxi trajectory data collected from 07/2016 to 09/2016 to evaluate our proposed cGAIL in inversely learning the driver agents' policy and reward functions. Our results demonstrate that the policies learned from cGAIL are on average 34.7% more accurate than those learned from other state-of-the-art baseline approaches.

A. Experiment settings

Evaluation metrics. In order to measure the accuracy of the learned policy net[3] from the empirical ground-truth policy from the collected data, we employ the Kullback-Leibler (KL) divergence [16] and L_2-norm [17].

Expert Driver Selection. In all inverse reinforcement learning (IRL) approaches [18], a common assumption is made that the demonstrations were collected from experts, namely, generated by the (near-)optimal policy. As a result, we select experienced drivers (with high earning efficiencies) from our datasets to

[3]Note that reward net and policy net are coupled in mimicking data distributions generated from driver agents. It is sufficient to evaluate policy net (rather than reward net) by comparing the obtained policy to the empirical policy from the data.

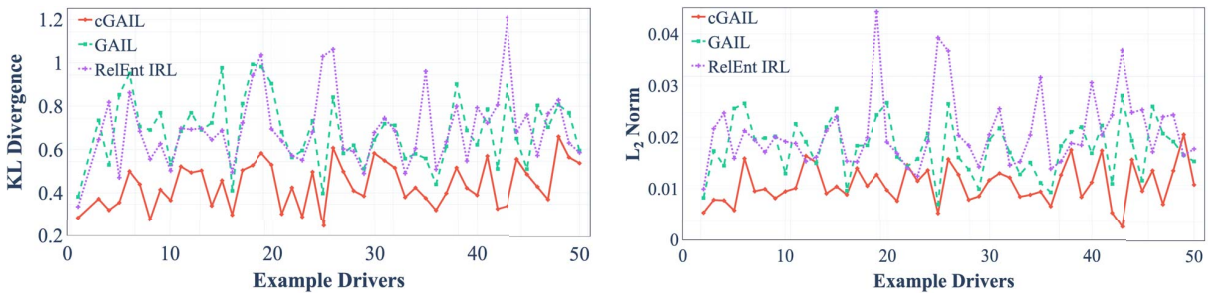

Fig. 3: Comparison with baselines

conduct our study. First, we quantify the expertise of taxi driver by their Earning efficiency r_e, defined as $r_e = E/t_w$, where E is the total income in the sampling time span, and t_w represents the driver's total working time in hour in the same sampling period of three months. We thus define and select expert drivers with earning efficiency ranked top 15% in 07/2016-09/2016. We denote this set of expert taxi drivers as $\mathcal{E}$, and each individual expert taxi driver as $e \in \mathcal{E}$. Eventually, we obtained a group of $3,044$ expert drivers for our study, out of a total of $17,877$ drivers from the data.

Testing location selection. For each expert taxi driver, we choose 20 grid cells as testing locations. The testing locations are with high visits by the driver, say, more than 2000 visits, so we have a relatively accurate estimate of the ground-truth policy in these grid cells. Then, we train the cGAIL model without these testing locations, infer the policies for these locations, and compare them with the ground-truth policies.

Baseline methods comparison. We learn expert taxi drivers' policies and compare the learning accuracy to various baseline methods, including MaxEnt IRL [4], MaxCausalEnt IRL [3], RelEnt IRL [5] and GAIL [6] against ground truth.

B. Experiment results

Figure 3 shows the KL-divergence and L_2-norm of the learned policies from the ground-truth policies for different methods. We randomly choose 50 driver agents (on the x-axis) to show the comparison results. The results with MaxEnt IRL and MaxCausalEnt IRL have poor accuracies, say, roughly 1.5–8 times of that with cGAIL, and we ignored their results for brevity. Their poor performances are simply due to the linear assumption of the reward function and their inaccurate estimation of transition probability matrix (given MaxEnt IRL and MaxCausalEnt IRL are both model-based approaches). When comparing to RelEnt IRL and GAIL, our proposed cGAIL still outperform them with an average of 34.7% and 31.0% reduction on KL-divergence and L_2-norm respectively.

VII. Conclusion

In this paper, we developed a novel conditional generative adversarial imitation learning (cGAIL) model that learns drivers' decision-making preferences and policies by transferring knowledge across taxi driver agents and across locations. Our evaluation results on three months of taxi GPS trajectory data in Shenzhen, China, demonstrated that the driver's preferences and policies learned from cGAIL are on average 34.7% more accurate than those learned from other state-of-the-art baseline approaches.

VIII. Acknowledgements

Yanhua Li and Xin Zhang were supported in part by NSF grants CNS-1657350 and CMMI-1831140, and a research grant from DiDi Chuxing Inc. Xun Zhou was partially supported by NSF grant IIS-1566386.

References

[1] A. Y. Ng, S. J. Russell, *et al.*, "Algorithms for inverse reinforcement learning.," in *ICML*, vol. 1, p. 2, 2000.
[2] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *ICML*, p. 1, ACM, 2004.
[3] B. D. Ziebart, J. A. Bagnell, and A. K. Dey, "Modeling interaction via the principle of maximum causal entropy," 2010.
[4] B. D. Ziebart, A. L. Maas, J. A. Bagnell, and A. K. Dey, "Maximum entropy inverse reinforcement learning.," in *Aaai*, vol. 8, pp. 1433–1438, Chicago, IL, USA, 2008.
[5] A. Boularias, J. Kober, and J. Peters, "Relative entropy inverse reinforcement learning," in *AISTATS*, pp. 182–189, 2011.
[6] J. Ho and S. Ermon, "Generative adversarial imitation learning," in *NeurIPS*, pp. 4565–4573, 2016.
[7] J. Fu, K. Luo, and S. Levine, "Learning robust rewards with adversarial inverse reinforcement learning," *arXiv preprint arXiv:1710.11248*, 2017.
[8] OpenStreetMap, "Road map data," 2016. data retrieved from Open Street Map, http://www.openstreetmap.org/.
[9] Y. Li, M. Steiner, J. Bao, L. Wang, and T. Zhu, "Region sampling and estimation of geosocial data with dynamic range calibration," in *ICDE*, pp. 1096–1107, IEEE, 2014.
[10] Y. Li, J. Luo, C.-Y. Chow, K.-L. Chan, Y. Ding, and F. Zhang, "Growing the charging station network for electric vehicles with trajectory data analytics," in *ICDE*, pp. 1376–1387, IEEE, 2015.
[11] A. A. Kumar, J. E. Kang, C. Kwon, and A. Nikolaev, "Inferring origin-destination pairs and utility-based travel preferences of shared mobility system users in a multi-modal environment," *Transportation Research Part B: Methodological*, vol. 91, pp. 270–291, 2016.
[12] G. Wu, Y. Ding, Y. Li, J. Luo, F. Zhang, and J. Fu, "Data-driven inverse learning of passenger preferences in urban public transits," in *IEEE CDC*, pp. 5068–5073, IEEE, 2017.
[13] G. Wu, Y. Li, J. Bao, Y. Zheng, J. Ye, and J. Luo, "Human-centric urban transit evaluation and planning," in *ICDM*, pp. 547–556, IEEE, 2018.
[14] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *NeurIPS*, pp. 2672–2680, 2014.
[15] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel, "Handwritten digit recognition with a back-propagation network," in *NeurIPS*, pp. 396–404, 1990.
[16] S. Kullback and R. A. Leibler, "On information and sufficiency," *The annals of mathematical statistics*, vol. 22, no. 1, pp. 79–86, 1951.
[17] H. Anton and C. Rorres, *Elementary linear algebra: applications version*. John Wiley & Sons, 2010.
[18] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and autonomous systems*, vol. 57, no. 5, pp. 469–483, 2009.