cGAIL: Conditional Generative Adversarial Imitation Learning —An Application in Taxi Drivers' Strategy Learning

Xin Zhang, Yanhua Li, Xun Zhou and Jun Luo

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

1 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 typically used as a solution to characterize such unique decision-making preferences of individual drivers. IRL learns a preference vector 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.

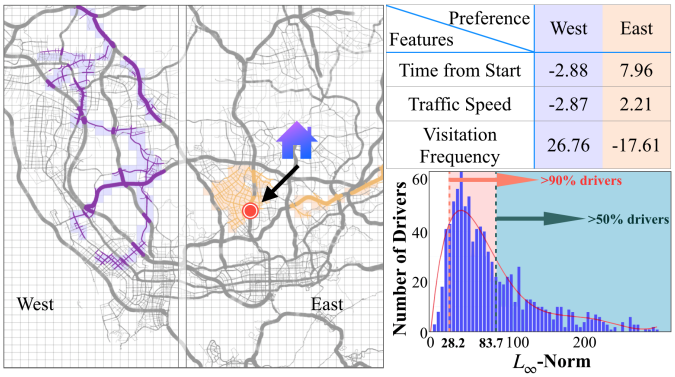


Fig. 1: Diverse driver preferences across regions.

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), and 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: (i) When the driver is on the east part, she prefers longer working time, i.e., close to the end of a day's work, since it is close to her home. However the preference is the opposite when she is working in the west

- *X. Zhang and Y. Li are with the Data Science Program, Worcester Polytechnic Institute, Worcester, MA, 01609. E-mail:{xzhang17, yli15}@wpi.edu*
- *X. Zhou is with the Tippie College of Business, University of Iowa, IA, 52242. E-mail:xun-zhou@uiowa.edu*
- *J. Luo is with the Lenovo Group Limited, Hong Kong, China. E-mail:jluo1@lenovo.com*

part of the city; (ii) The driver prefers regions with higher driving speed on the east part to avoid traffic, but prefers low driving speed areas in the west part to increase the chance of finding a passenger since it is primarily rural areas; (iii) The driver prefers less popular areas in the east (downtown) part due to congested traffic, but prefers the opposite in the west area (i.e., 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' preferences 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. 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 these observations, in this paper, we make the first attempt to develop conditional generative adversarial imitation learning (cGAIL[1]) model, that learns drivers' decision-making preferences and policies by transferring knowledge across taxi driver agents and across locations. **Our contributions** are summarized as follows:

- 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 4).
- We 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 5).
- 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 6).
- Compared with the preliminary version of this work in [8], we have i) introduced various inverse reinforcement learning approaches, highlighted their limitations, to motivate our design of cGAIL in Sec 5.2; ii) provided more comprehensive experimental results, to evaluate how various system parameters impact the learning performances (in Sec 6.5); with case studies to explain what practical factors impact the comparison results between our proposed cGAIL vs GAIL (in Sec 6.6); iii) provided more comprehensive discussion of related works in both Urban Computing area and Inverse Reinforcement Learning fields, and clearly distinguished our work from these works in Sec 7.

1. A preliminary version of the results in this paper appeared in [8].

2 Overview

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

2.1 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 taxis. Each of these taxis was equipped with a GPS set allowing them to generate GPS records in roughly every 30 seconds. This yields a total of 51,485,760 GPS records on a daily basis. 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. A sequence of GPS records forms a trajectory.

Road map data. The road map data of Shenzhen is obtained from OpenStreetMap [9], covering an area from 22.44°N to 22.87°N in latitude and from 113.75°E to 114.65°E in longitude. There are $455,944$ road segments in the datasets with six road levels, such as primary road, secondary road, tertiary road, trunk road, motorway, and unclassified road.

2.2 Problem Definition

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 as a sequence of GPS records.

Definition 1 (**Trajectory** tr) A trajectory tr is a sequence of spatio-temporal points, when the taxi is vacant and the driver is looking for passengers, e.g., starting from dropping off the last passenger to the next passenger aboard. Therefore, a trajectory can be 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. Hence, we have $Tr_d = \{tr_{d,1}, \cdots, tr_{d,m_d}\}$, with m_d as the number of trajectories from d.

Note that we focus on drivers' "seeking" trajectories, demonstrating the drivers' decision-making behaviors, when looking for passengers. We ignore the trajectories with passengers aboard, since drivers do not make strategic decisions except for heading towards the passenger's destination. As a result, the data captures a sequence 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 (formally defined below): *(i) reward function* (evaluating how effective a next

action is) and *(i) policy function* (likelihood of choosing a particular action).

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. In other words, each taxi driver is (implicitly) following a unique reward function when making decisions. 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, since drivers are intrinsically pursuing 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. For example, when a driver d's state s is "in the middle of 3rd Ave., which is along the east-west direction, at 3PM", the policy function $\pi(\cdot|s, d) = \{p(East) = 0.3, p(West) = 0.7\}$ means the driver d under such a situation has 30% chance to choose to go east, while 70% chance to go west.

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. Next, we outline our solution framework to tackle the two challenges and solve the proposed collective inverse preference learning problem.

2.3 Solution Framework

Figure 2 outlines our proposed solution framework. It takes road map and taxi GPS data as input and consists of three main components: (i) *Stage 1 - data preparation*, which partitions the urban area into equal-size grid cells, aggregates taxi drivers' GPS records in grid cell level, and extracts decision-making features from them; (ii) *Stage 2 - data-driven modeling*, which models taxi driver's trips as Markov decision processes (MDPs); (iii) *Stage 3 - conditional inverse preference learning*, where we develop a novel conditional generative adversarial imitation learning (cGAIL) algorithm to learn taxi driver's policy function and reward function.

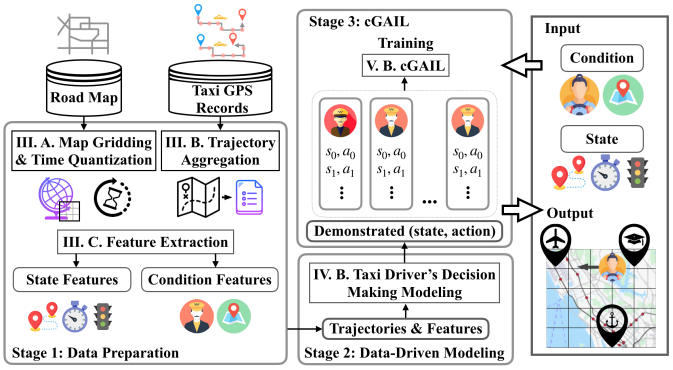


Fig. 2: Solution Framework.

3 Stage 1 - Data Preparation

3.1 Map Gridding and Time Quantization.

Map griding. For the ease of analyzing taxi drivers' decision-making behaviors, we partition the city into small equal side-length grid cells [10], [11] with pre-defined side-length $b = 0.01°$. It leads to $1,934$ grid cells connected by road network (See Fig 3). 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 $1 \leq j \leq 288$.

3.2 Trajectory Aggregation

A combination of a grid cell g_i, time interval $\tilde{t}_j$, and the day of the 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 2.2 can then be mapped as sequences of spatio-temporal states s, and the set of d's trajectories can be denoted by $\mathcal{T}_d$:

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

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

3.3 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* $\mathbf{f_s}$, which characterizes various statistics of the state from the historical data and *condition features* $\mathbf{f_c}$, which captures the features of the current state with respect to the driver's profile, e.g., home location, etc. 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

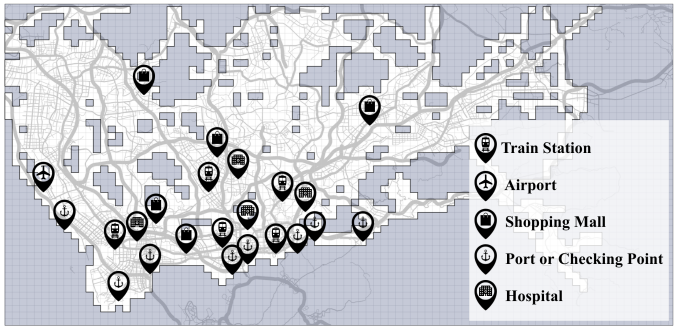


Fig. 3: Distribution of places of interests (PoIs).

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 $\mathbf{f_D}$. We detail all these state features 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 $f_{T,4}$. Travel demand $f_{T,1}$ captures the average number of requests for taxi pickups of the state s in the historical data. Traffic volume $f_{T,2}$ represents the average number of taxis in a state s from the historical data. It shows how congested a state is. A higher traffic volume is likely to be an evidence of heavy traffic, and a lower one is likely to show a light traffic. Traffic speed $f_{T,3}$ estimates the average speed of all trajectories passing a state s in the historical data. Low traffic speed indicates that s is likely to be under traffic congestion. Waiting time $f_{s,4}$ captures the average time a taxi stays in the target state s from the historical taxi trajectory data.

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.

Fig 3 shows the geodistribution of 23 places of interests for $f_{D,1}$ to $f_{D,23}$.

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 ($f_{c,1}$): Each driver's home location can be extracted from their GPS records. Different drivers have different preferences to work closer vs far away to the home location. This feature characterizes the distance in kilometers from the current state location to the driver's home location.

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

Familiarity ($f_{c,4}$): This feature captures the average visitations of the driver to the current state s from the historical data. It indicates 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.

4 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 review the Markov Decision Process (MDP), and elaborate on how to model taxi drivers' decision-making processes as MDPs.

4.1 Markov Decision Process (MDP)

Markov decision processes (MDPs) [12] provides a mathematical framework to model stochastic decision making processes. An MDP includes an agent as the decision maker and the environment that interacts with the agent. Each MDP contains five elements forming a 5-tuple $\langle \mathcal{S}, \mathcal{A}, P, R, \gamma \rangle$:

- $\mathcal{S}$ is a set of states, and $\mathcal{A}$ is a set of actions;
- P is the transition probability with $P(s'|s,a)$ as the probability of reaching state s' by taking action a at state s;
- $R : S \times A \mapsto \mathbb{R}$ is the reward function that outputs scores for a given state-action pair;
- $\gamma \in [0,1]$ is the discount factor informing importance difference between future and present rewards.

The agent of an MDP starts from an initial state $s_0 \in \mathcal{S}$ and makes an action $a \in \mathcal{A}$ following his/her policy π. A policy specifies a probability distribution on actions to be executed in each state, defined as $\pi : \mathcal{S} \times \mathcal{A} \mapsto [0,1]$. The action a taken at state s leads to the transition to the next state s' based on the transition probability $P(s'|s,a)$ and receives a reward under the reward function $R(s,a)$. Continuing along this process generates a sequence of state-action pairs which we refer to as a *trajectory*[2]. We use $\tau = [(s_0,a_0),(s_1,a_1),...,(s_L,a_L)]$ to denote a trajectory generated by an agent over an MDP, where L is trajectory length. Without loss of generality, we set $\gamma = 1$ in this work. The problem of MDP aims at finding a policy π, such that the expected total reward is maximized, i.e., in the following eq.(2),

$$\pi^* = \arg\max_{\pi \in \Pi} \mathbb{E}^{\pi}\left(\sum_{t=0}^{T} \gamma^t R(s_t, a_t) | s_0 \sim \mu_0\right), \tag{2}$$

where s_t and a_t are random variables for the state and action at the time step t, and $T \in \mathbb{R} \cup \{\infty\}$ is the time horizon. The initial state s_0 follows the initial distribution $\mu_0 : \mathcal{S} \mapsto [0,1]$. Here, Π is the memory-less policy space.

4.2 Modeling Taxi Driver's Decision Making with MDP

We consider each taxi driver as an "agent". When looking for passengers, the driver keeps evaluating various features in surrounding areas of the current spatio-temporal region s, based on which the driver decides which direction to go to find the passengers. This whole process consisting

2. In this paper we use "trajectory" to refer to both the physical GPS trace of a taxi and the state-action pairs of a driver in the MDP model because each physical "trajectory" can be mapped to a certain sequence of state-action pairs so the two concepts are equivalent in our problem.

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 3. 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.

5 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 below:

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 inverse reinforcement learning, IRL (with linear function assumption) and generative adversarial imitation learning, GAIL (for non-linear functions) in Sec 5.1. For *Q2*, we develop a novel conditional generative adversarial imitation learning, cGAIL, in Sec 5.2. The proposed cGAIL model is a unifying inverse learning model that allows knowledge transfer across taxi driver agents and across locations.

5.1 Learning Reward/Policy functions with IRL and GAIL

User choice modeling has been extensively studied in the literature aiming to learn human agents' decision-making reward and policy functions from data they generated [4], [6], [13]–[16], where inverse reinforcement learning (IRL) [1] models assume linear reward function, and GAIL [6] learns general non-linear reward function. We briefly introduce both below, and highlight their limitations on the transferability across locations and agents. Built upon these approaches, we will propose our cGAIL model in Sec 5.2.

IRL with linear reward function assumption. Given the observed trajectory data set $\mathcal{T}$ from a driver agent and the features, $\mathbf{f} = [\mathbf{f_s}, \mathbf{f_c}]$, various IRL approaches have been proposed to inversely learn the agent's reward function, such as Apprenticeship learning [2], Maximum Entropy IRL [4], Maximum Causal Entropy IRL [3], and Relative Entropy IRL [5]. They all share the same principle and goal of recovering a linear reward function of the agent under which the observed trajectories have the highest likelihood to be generated. Below, we use Maximum Causal Entropy (MaxCausalEnt) IRL [3] as an example to illustrate the problem formulation (where other IRL approaches [4], [6], [13]–[16] share a common framework, and we omit the details for brevity[3]),

$$\max_{\pi_{\boldsymbol{\theta}}} : H(\pi_{\boldsymbol{\theta}}), \tag{3}$$

$$\text{s.t.: } \mathbb{E}_{\pi_{\boldsymbol{\theta}}}[R_{\boldsymbol{\theta}}(s,a)] = \tilde{\mathbb{E}}_{\pi_E}[R_{\boldsymbol{\theta}}(s,a)], \quad \forall s, a \in \mathcal{S}, \mathcal{A}, \tag{4}$$

$$\sum_a \pi_{\boldsymbol{\theta}}(s,a) = 1, \quad \forall s \in \mathcal{S}, \tag{5}$$

where $H(\pi_{\boldsymbol{\theta}}) = -\sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} D_{\pi_{\boldsymbol{\theta}}}(s,a) \ln \pi_{\boldsymbol{\theta}}(a|s)$ is the total causal entropy[4] induced by a policy $\pi_{\boldsymbol{\theta}}$, which measures the uncertainty present in the (causally conditioned) trajectory data distribution; $\mathbb{E}_{\pi_{\boldsymbol{\theta}}}[R_{\boldsymbol{\theta}}(s,a)]$ and $\tilde{\mathbb{E}}_{\pi_E}[R_{\boldsymbol{\theta}}(s,a)]$ are the expected reward over a state-action pair (s, a) induced by a policy $\pi_{\boldsymbol{\theta}}$ and by the empirical data $\mathcal{T}$, respectively. The reward function $R_{\boldsymbol{\theta}}(s,a) = \boldsymbol{\theta}^\top \cdot \mathbf{f}(s,a)$ is a linear combination of a reward (weight) vector and the feature vector $\mathbf{f}(s,a)$ at state-action pair (s, a). It is proven that the reward function (i.e., vector) $\boldsymbol{\theta}$ and the policy $\pi_{\boldsymbol{\theta}}$ in the above IRL problem can be optimally solved with maximum likelihood estimation (MLE). However, the linear assumption of reward function is in general too strong for real world applications, where GAIL [6] (as highlighted below) was proposed to extend the linear reward function in IRL approaches to non-linear fashion using deep neural networks.

GAIL for general non-linear reward function. Generative adversarial imitation learning (GAIL) [6] naturally extends

3. Note that IRL approaches [2]–[5] share the common formulation in eq.(3)-(5), with various forms of $H(\pi)$, $\mathbb{E}_\pi[R(s,a)]$, and $\tilde{\mathbb{E}}_{\pi_E}[R(s,a)]$.

4. $D_{\pi_{\boldsymbol{\theta}}}(s,a)$ is the visitation frequency at (s, a) under $\pi_{\boldsymbol{\theta}}$. Please see more details in [3].

the IRL formulation (presented in eq.(3)–(5)) by a non-linear reward function $R(s,a)$, and a non-linear policy function $\pi(a|s)$ both using deep neural networks. The generalization was made by the following connections.

First, MaxCausalEnt IRL (in eq.(3)–(5)) can be rewritten as eq.(6) below (See eq.(1) in [6]), by rewriting the constraints into the objective function[5]:

$$\max_R \big(\min_\pi -H(\pi) - \mathbb{E}_\pi[R(s,a)]\big) + \mathbb{E}_{\pi_E}[R(s,a)]. \quad (6)$$

In large-scale real world problems, with a large amount of demonstration data, GAIL introduces a regularizer function $\psi(R)$ to avoid overfitting, which leads to eq.(7),

$$\max_R \psi(R) + \big(\min_\pi -H(\pi) - \mathbb{E}_\pi[R(s,a)]\big) + \mathbb{E}_{\pi_E}[R(s,a)]. \quad (7)$$

It was proven in [6] that when the function $\psi(R)$ is properly chosen, the dual problem of eq.(7) 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.(6) becomes[6]

$$\begin{aligned} &\min_\pi \quad -\lambda H(\pi) + D_{JS}(\pi, \pi_E), \quad \text{with} \quad (8)\\ &D_{JS}(\pi,\pi_E) = \max_R \mathbb{E}_{\pi_E}[\ln(R(s,a))] + \mathbb{E}_\pi[\ln(1-R(s,a))], \end{aligned}$$

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.(8) can be tackled using generative adversarial networks (GAN) model [17], 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.

5.2 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 3.3), 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.(6) 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 $\mathbf{c}$.

$$\begin{aligned} \max_R \min_\pi \quad & -\lambda H(\pi(\cdot|\mathbf{c})) + \mathbb{E}_{\pi_E}[\ln(R(s,a|\mathbf{c}))] \\ & + \mathbb{E}_\pi[\ln(1-R(s,a|\mathbf{c}))] + \mathbb{E}_{\pi_E}[\ln(1-R(s,a|\mathbf{c}'))], \end{aligned} \quad (9)$$

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 a different 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 below:

- 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 $[\mathbf{f_T}(s), \mathbf{f_T}(s'_1), \cdots, \mathbf{f_T}(s'_{24})]$.
- Temporal features of the current state s, $\mathbf{f_M}(s) = [f_{M,1}(s), f_{M,2}(s)]$, indicating the current time of the day and the day of the week.
- POI distance features of the current state s, $\mathbf{f_D}(s) = [f_{D,1}(s), \cdots, f_{D,23}(s)]$, indicating the current distances to 23 places of interests.

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 the nine actions (directions), and a particular action direction a. These actions will be randomly chosen based on $\pi(\cdot|s)$. Fig 4 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 [18] 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

5. 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.

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

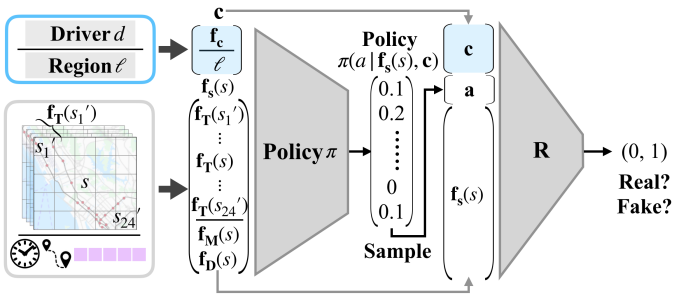


Fig. 4: cGAIL model structure.

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 3.1) 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 $1 \leq i \leq K$, 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.(11) to maximize $\tilde{V}_R$ in eq.(10), with step size η_R.

$$\tilde{V}_R = \frac{1}{m}\sum_{j=1}^{m}\Big(\ln\left(R(\mathbf{f}_s(s_j^i), a_j^i|\mathbf{c}_j^i)\right) + \ln\left(1 - R(\mathbf{f}_s(s_j^i), \tilde{a}_j^i|\mathbf{c}_j^i)\right) + \ln\left(1 - R(\mathbf{f}_s(s_j^i), a_j^i|\hat{\mathbf{c}}_j^i)\right)\Big), \tag{10}$$

$$\boldsymbol{\theta}_R = \boldsymbol{\theta}_R + \eta_R \nabla_{\boldsymbol{\theta}_R} \tilde{V}_R. \tag{11}$$

Next, we update policy network parameters $\boldsymbol{\theta}_\pi$ by eq.(12) to minimize $\tilde{V}_\pi$ below, with η_π as the step size (Line 6).

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

$$\boldsymbol{\theta}_\pi = \boldsymbol{\theta}_\pi + \eta_\pi \nabla_{\boldsymbol{\theta}_\pi} \tilde{V}_\pi. \tag{12}$$

6 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.

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 $1 \leq i \leq K$ **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.11;
6: Update $\boldsymbol{\theta}_R$ with Eq.12;
7: **end for**

6.1 Evaluation metrics

In order to measure the accuracy of the learned policy net [7] from the empirical ground-truth policy from the collected data, we employ the Kullback-Leibler (KL) divergence and L_2-norm. KL divergence [19] measures how one distribution P is different from a ground truth distribution Q as in eq.13,

$$D_{KL}(P||Q) = -\sum_{x\in\mathcal{X}} P(x)\ln\frac{Q(x)}{P(x)}. \tag{13}$$

L_2-norm [20] is also called the Euclidean norm, which views two n-dimensional policies as two points in n-dimensional space and measures the ordinary distance from the ground truth policy $Q = (q_1, \cdots, q_n)$to the learned policy$P = (p_1, \cdots, p_n)$, i.e. in eq.14,

$$L_2(P, Q) = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2} \tag{14}$$

6.2 Expert Driver Selection

In all inverse reinforcement learning (IRL) approaches [21], 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 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. Based on the earning efficiency, we define and select expert drivers with earning efficiency ranked top 15% in 07/2016-09/2016 (with average monthly earning efficiency above 61.5 CNY/hour). 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,

7. 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.

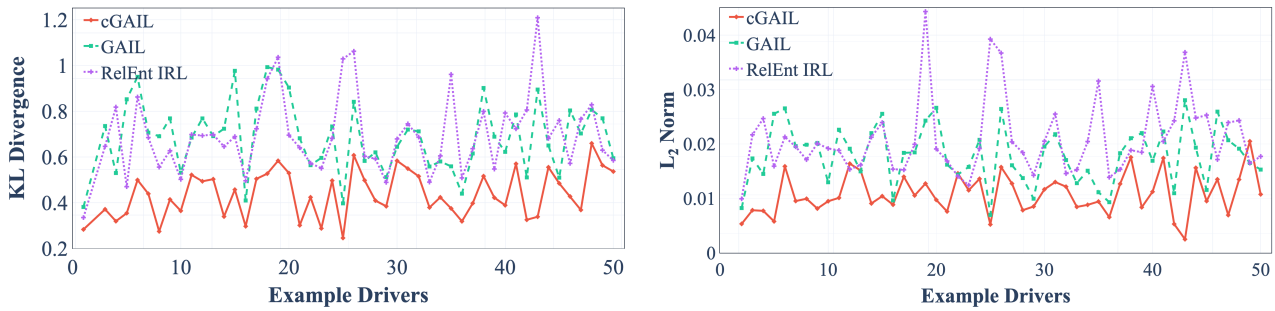


Fig. 5: Comparison with baselines

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.

6.3 Experiment settings.

We conduct two sets of data-driven experiments.

- 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;
- Effects of model parameters. we evaluate and examine how the policy learning accuracy varies with different parameters in cGAIL framework.
- Case studies. We present case studies in a few concrete driver examples to explain why cGAIL outperforms other baseline approaches.

6.4 Baseline methods comparison

When implementing cGAIL, we employ three (de-)convolutional layers with sigmoid activation functions for both policy net and reward net. Given the input state features with size of $5 \times 5 \times 5$, we use a kernel size of 2×2 for the convolutional layers with padding of 1 and shrink the channel number 1 at a layer. Similar network structures are used for baseline method GAIL for a fair comparison.

Figure 5 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 distance respectively.

6.5 Effects of model parameters

When applied to a particular target driver d and a location ℓ, cGAIL can precisely generate actions for various states the driver d may encounter at ℓ. Especially, even no trajectory data was collected for d at ℓ, cGAIL can still accurately estimate d's policy at ℓ by transferring knowledge from other "supporting" drivers used in the training process. However, given a target driver d, data from different supporting drivers may have diverse abilities to assist learning the policy of d at a location ℓ. Intuitively, the more "similar" the supporting drivers are to the target driver, the more knowledge we can utilize to "support" learning the policy of target driver d. Below, we evaluate how two similarity measures between the target and supporting drivers and the total number of supporting drivers affect the accuracy of learned policy for target drivers at various locations. We run our experiments over all expert drivers (as target drivers separately) in the selected group $\mathcal{E}$. Similar results were obtained for all drivers, where we demonstrate the results from a few example drivers. Fig. 6 illustrates one example target driver, with orange grid cells as 20 testing locations selected with the number of visits more than 2,000 times. Clearly, those highly visited locations are primarily transport stations, airport, as marked in the figure. In the training process, we hide the data from the 20 testing locations to train cGAIL, and compare the difference between the learned policies and the true policies observed in the data.

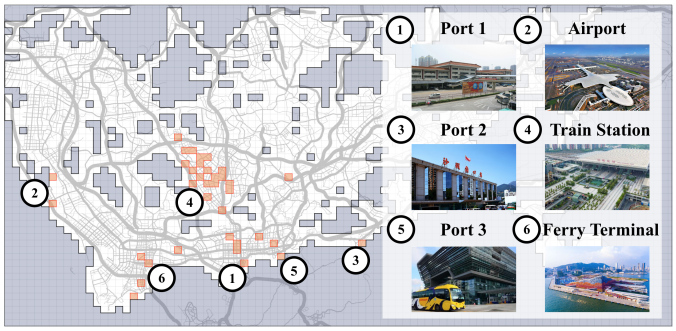
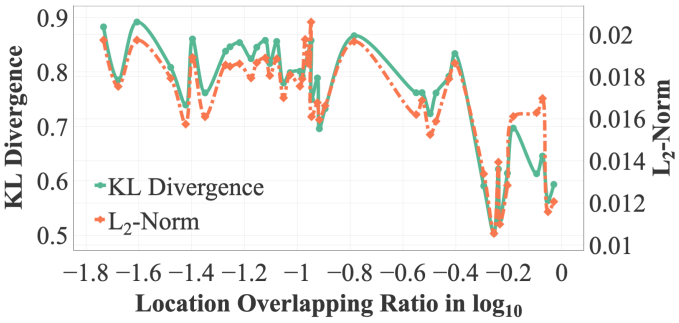
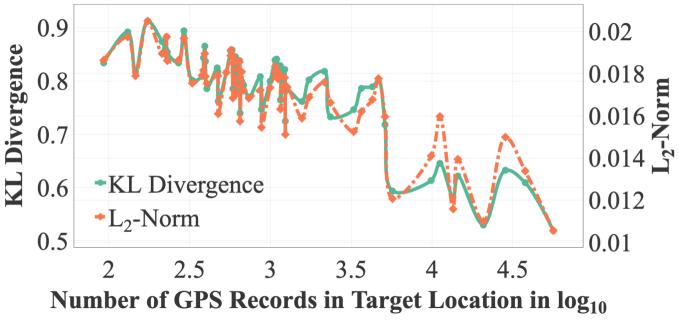


Fig. 6: 20 testing locations of an example driver agent.

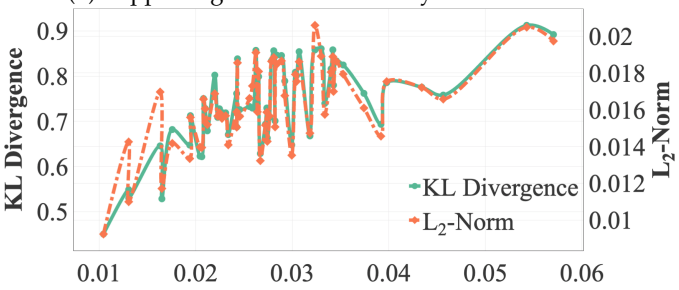
Working location overlapping ratio. This measure quantifies how similar in working regions between the supporting driver and the target driver, and is calculated by the ratio between the overlapping working regions and the union of all working regions of the supporting and target driver. Fig 7a shows that as the location overlapping ratio increases between the supporting and target drivers, the accuracy of learned policy (in KL-divergence on the left and L_2-norm on the right) improves significantly.



(a) Supporting and target driver's common locations.



(b) Supporting driver's familiarity to test locations.



(c) Policy similarity in L_2-norm.

Fig. 7: Effects of cGAIL parameters.

Supporting driver's familiarity to the testing locations. This measure quantifies the number of times the supporting driver has visited testing locations from the training data, which indicates the familiarity of the supporting driver to the testing location. Fig 7b shows that as the familiarity increases, the accuracy of learned policy (in KL-divergence on the left and L_2-norm) for the target driver improves accordingly.

Supporting driver's policy similarity to target driver. This parameter uses L_2-norm to evaluate how similar a supporting driver is to the target driver when making decisions in their common visited areas. Fig 7c shows that higher policy similarity (in L_2-norm) contributes to better cGAIL policy inference.

The number of supporting drivers. We examined the effect of the number of supporting drivers from 2 to 49 to the accuracy of learned policy of target driver. We randomly choose 50 target drivers from $\mathcal{E}$ to reduce the randomness of the results. For each target, we randomly choose 2 to 49 supporting drivers from $\mathcal{E}$ to train a cGAIL model and evaluate the average accuracy of learned policies in 20 testing locations. Fig. 8 shows the average KL-divergence and L_2-norm over the number of supporting drivers. Clearly, more supporting drivers lead to more accurate learned policies for target driver (with both KL-divergence and L_2-norm decreases). Moreover, when there are less than five supporting drivers, adding an additional supporting driver improves the learning accuracy significantly, while when there are more than five supporting drivers, the effect of adding additional drivers becomes smaller.

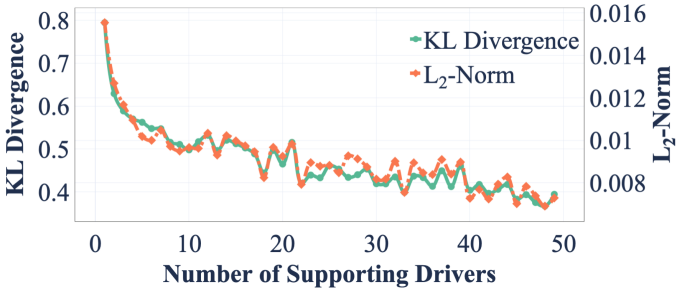


Fig. 8: Number of supporting drivers.

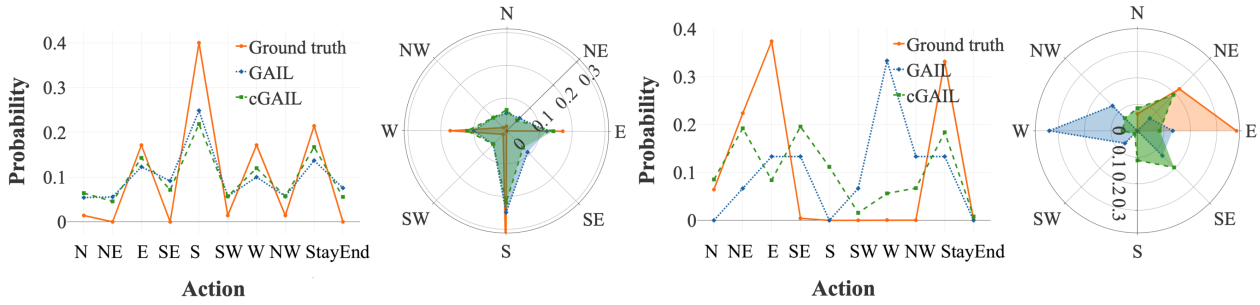
6.6 Case studies

Now, we further dissect how and under what circumstances, cGAIL improves the policy learning accuracy (over GAIL), by zooming in to look at the learned policies vs ground-truth policies in a few representative case studies. In most (about 97%) of the cases, our cGAIL can learn a more accurate policy than GAIL by transferring knowledge from supporting drivers. Under certain cases, we observe that cGAIL and GAIL both perform well, or both perform badly. We will investigate these cases in more details below.

When cGAIL and GAIL both perform well? Fig 9a shows (in both probability density distribution on the left and dimension-map on the right) an example case, where the policies learned from cGAIL (green dashed) and GAIL (blue dotted) both match the ground-truth policy (orange) well. This happens because the target driver has consistent policy function over locations, namely, the policy function is independent to locations. Thus, without supporting drivers' knowledge, GAIL can perform well to learn the ground-truth policy.

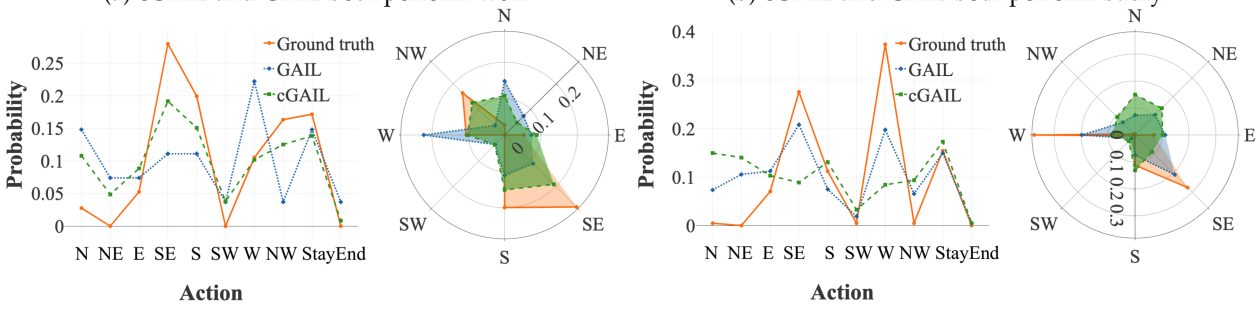
When cGAIL and GAIL both perform badly? Fig 9b shows an example case, where the policies learned from cGAIL (green dashed) and GAIL (blue dotted) both match the ground-truth policy (orange) poorly. Two reasons contribute to this phenomenon. Firstly, the target driver's policy is highly dependent to the locations. Therefore, GAIL is not capable of learning a good policy to locations not observed in the training data. Secondly, the supporting driver is not familiar to testing location, i.e., only visited it 106 times. This case can be improved by carefully choosing more supporting drivers for the target driver.

When cGAIL outperforms GAIL? Fig 9c shows the case where cGAIL gives better matching on ground truth policy. This is due to the positive supports from the supporting driver. In this case, the supporting driver is familiar to the testing location with high visitation frequency and acts similarly to the target driver with similar policy in common activity regions.



(a) cGAIL and GAIL both perform well.

(b) cGAIL and GAIL both perform badly.



(c) cGAIL outperforms GAIL.

(d) GAIL outperforms cGAIL.

When GAIL outperforms cGAIL? Fig 9d shows the case where GAIL outperforms cGAIL. This is the effect of misleading knowledge from a supporting drier. In this location, the chosen supporting driver has adequate experience driving in the test location. However, the supporting driver acts nothing like the target driver. This introduces misleading information to the inference of target driver's policy. This case shows the importance of careful supporting driver selection in application.

7 Related Work

In this section, we summarize the literature works in two related areas to our study: 1) urban computing, and 2) inverse reinforcement learning.

Urban Computing is a research area encompassing urban sensing, data management and data analytic. It forms a unified process to explore, analyze and solve existing critical problems that are closely related with daily life in urban area such as traffic congestion, energy consumption and pollution [22]. In taxi operation management, works are focusing on dispatching [23], [24], and passenger seeking [25], [26], [27]. They targets finding an optimal actionable solution to improve the performance/revenue of individual taxi drivers. [28] solves the passenger seeking problem by giving direction recommendations to drivers. However, all of these works focus on finding the optimal strategies instead of answering "how" good drivers make decisions. By contrast, our work focuses on learning the decision-making policy and reward from taxi drivers.

Inverse Reinforcement Learning (Imitation Learning) aims to inversely learn the reward function and policy of experts from their demonstrations [1]–[7], which models the agent's decision making process as Markov Decision Processes (MDPs). MaxEnt IRL [4], MaxCausalEnt IRL [3], and RelEnt IRL [5] were proposed to learn a reward function with maximized entropy, causal entropy, and relative entropy of the distribution on trajectories under the learned policy, respectively. They all assume a linear reward function of the feature vectors associated with state-action pairs. GAIL [6], [29] extends the above approaches (especially MaxCausalEnt IRL) to general non-linear reward function by using generative adversarial networks (GANs) framework. Adversarial inverse reinforcement learning (AIRL) [7] also employs GANs structure to learn non-linear policy function for individual agent, by considering the dynamics of the environment (i.e., the transition probability), but AIRL does not allow knowledge sharing across agents, and environment dynamics studied in AIRL does not exist in our problem. Differing from all these works, our study focuses on a unique scenario, where the human agent's reward/policy are location dependent, and each human agent only has limited data coverage over location. We develop a unifying conditional generative adversarial imitation learning framework, which collectively integrates and transfers knowledge across human agents and locations to improve the policy learning accuracy.

8 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.

9 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] X. Zhang, Y. Li, X. Zhou, and J. Luo, "Unveiling taxi drivers strategies via cgail-conditional generative adversarial imitation learning," in *2019 IEEE International Conference on Data Mining (ICDM)*, IEEE, 2019.

[9] OpenStreetMap, "Road map data," 2016. data retrieved from Open Street Map, http://www.openstreetmap.org/.

[10] 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.

[11] 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.

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

[13] 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.

[14] L. Zhang, "Agent-based behavioral model of spatial learning and route choice," tech. rep., 2006.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] S. Kullback and R. A. Leibler, "On information and sufficiency," *The annals of mathematical statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[20] H. Anton and C. Rorres, *Elementary linear algebra: applications version*. John Wiley & Sons, 2010.

[21] 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.

[22] Y. Zheng, L. Capra, O. Wolfson, and H. Yang, "Urban computing: concepts, methodologies, and applications," *TIST*, vol. 5, no. 3, p. 38, 2014.

[23] S. Ma, Y. Zheng, and O. Wolfson, "T-share: A large-scale dynamic taxi ridesharing service," in *ICDE*, pp. 410–421, IEEE, 2013.

[24] N. J. Yuan, Y. Zheng, L. Zhang, and X. Xie, "T-finder: A recommender system for finding passengers and vacant taxis," *TKDE*, vol. 25, no. 10, pp. 2390–2403, 2012.

[25] J. Yuan, Y. Zheng, L. Zhang, X. Xie, and G. Sun, "Where to find my next passenger," in *Ubicomp*, pp. 109–118, ACM, 2011.

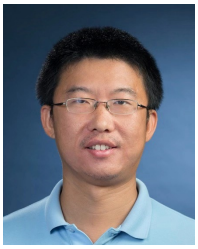
[26] Y. Ge, H. Xiong, A. Tuzhilin, K. Xiao, M. Gruteser, and M. Pazzani, "An energy-efficient mobile recommender system," in *KDD*, pp. 899–908, ACM, 2010.

[27] Y. Ge, C. Liu, H. Xiong, and J. Chen, "A taxi business intelligence system," in *KDD*, pp. 735–738, ACM, 2011.

[28] H. Rong, X. Zhou, C. Yang, Z. Shafiq, and A. Liu, "The rich and the poor: A markov decision process approach to optimizing taxi driver revenue efficiency," in *CIKM*, pp. 2329–2334, ACM, 2016.

[29] A. Kuefler, J. Morton, T. Wheeler, and M. Kochenderfer, "Imitating driver behavior with generative adversarial networks," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, pp. 204–211, IEEE, 2017.

Xin Zhang received B.S. degree in applied mathematics from University of Illinois at Urbana-Champaign, IL, USA in 2017 and M.S. degree in data sciences from Worcester Polytechnic Institute, MA, USA in 2019. She is a Ph.D. candidate in the department of data sciences in Worcester Polytechnic Institute. Her research interests are curriculum learning, inverse reinforcement learning and their applications on urban data.



Yanhua Li received two Ph.D. degrees in electrical engineering from Beijing University of Posts and Telecommunications, Beijing in China in 2009 and in computer science from University of Minnesota at Twin Cities in 2013, respectively. He has worked as a researcher in HUAWEI Noahs Ark LAB at Hong Kong from Aug 2013 to Dec 2014, and has interned in Bell Labs in New Jersey, Microsoft Research Asia, and HUAWEI research labs of America from 2011 to 2013. He is currently an Assistant Professor in the Department of Computer Science at Worcester Polytechnic Institute (WPI) in Worcester, MA. His research interests are big data analytics and urban computing in many contexts, including urban network data analytics and management, urban planning and optimization.

Xun Zhou is currently an Assistant Professor in the Department of Management Sciences at the University of Iowa. He received a PhD degree in Computer Science from the University of Minnesota, Twin Cities in 2014. His research interests include big data management and analytics, spatial and spatio-temporal data mining, and Geographic Information Systems (GIS). He has published over 30 papers in these areas and has received three best paper awards. He also served as a co-editor-in-chief of the Encyclopedia of GIS, 2nd Edition.

Jun Luo is a principal researcher at Lenovo Machine Intelligence Center, Lenovo Group Limited, in Hong Kong. He received his PhD degree in computer science from the University of Texas at Dallas, USA, in 2006. His research interests include big data, machine learning, spatial temporal data mining and computational geometry. He has published over 90 journal and conference papers in these areas.