TrajGAIL: Trajectory Generative Adversarial Imitation Learning for Long-term Decision Analysis

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

Ziming Zhang
Worcester Polytechnic Institute
zzhang15@wpi.edu

Jun Luo
Lenovo Group Limited
jluo1@lenovo.com

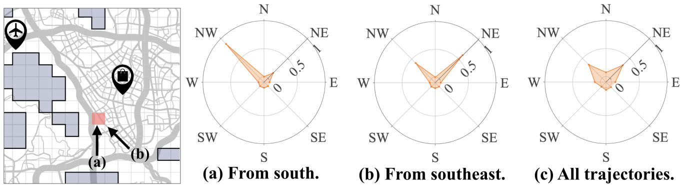
***Abstract*—Mobile sensing and information technology have enabled us to collect a large amount of mobility data from human decision-makers, for example, GPS trajectories from taxis, Uber cars, and passenger trip data of taking buses and trains. Understanding and learning human decision-making strategies from such data can potentially promote individual's well-being and improve the transportation service quality. Existing works on human strategy learning, such as inverse reinforcement learning, all model the decision-making process as a Markov decision process, thus assuming the Markov property. In this work, we show that such Markov property does not hold in real-world human decision-making processes. To tackle this challenge, we develop a Trajectory Generative Adversarial Imitation Learning (TrajGAIL) framework. It captures the long-term decision dependency by modeling the human decision processes as variable length Markov decision processes (VLMDPs), and designs a deep-neural-network-based framework to inversely learn the decision-making strategy from the human agent's historical dataset. We validate our framework using two real world human-generated spatial-temporal datasets including taxi driver passenger-seeking decision data and public transit trip data. Results demonstrate significant accuracy improvement in learning human decision-making strategies, when comparing to baselines with Markov property assumptions.**

***Index Terms*—Spatial-temporal data mining, human decision analysis, inverse reinforcement learning, imitation learning**

I. Introduction

Rapid development of mobile sensing and information technology enables us to collect massive amounts of mobility data from human decision-makers, which we call *human-generated spatial-temporal data (HSTD)*. Examples of emerging HSTD include GPS trajectories collected from taxis and personal vehicles, passenger trip data from automated fare collection devices on buses and trains, and working traces from the emerging gig-economy services, e.g., food delivery (DoorDash [1], Postmates [2]), and everyday tasks (TaskRabbit [3]). *Harnessing HSTD to extract the unique decision-making strategies of human agents* has transformative potential in many applications, including promoting individual well-being of gig-workers [4], [5], and improving service quality and revenue of transportation service providers [6]–[9].

Traditional methods of learning human decision-making strategies from HSTD, such as inverse reinforcement learning (IRL) and imitation learning (IL) [6], [8], [9], all model urban human decision-making processes as Markov decision processes (MDPs). Such MDP models have a strong Markov property assumption [10], namely, each decision made only depends on the current state of the human agent, not on any prior states or decisions. For example, with the Markov property assumption, prior works commonly assume that a taxi driver's decision of which direction to go to find the next passenger, should only depend on where she is, not on where she has visited.



(a) From south.
(b) From southeast.
(c) All trajectories.

Fig. 1: Taxi drivers' decision distributions (in polar plots) do not follow Markov Property.

In reality, when human agents make decisions in spatial-temporal spaces, they are likely to consider where they were, what they have experienced, and what decisions they have made, which was also suggested and supported by theories in behavioral and psychology research, such as the goal setting theory [11] and the ego depletion theory [12]. Moreover, Fig. 1 provides empirical evidence from real world taxi trajectory data that such Markov property does not hold. For a taxi driver, we extract all her passenger-seeking trajectories, that traversed the same "state", defined by a particular location (the orange box in the center of the map) at the same time 10AM of a day. Consider the driver has eight neighboring regions to go (as decisions). The decision strategy, i.e., the probability distribution of choosing different decisions, represented as the polar plots, vary significantly based on where the driver was from [1]. It is common that Markov property does not hold in many urban human decision-making scenarios (see more in Sec IV). Essentially, human decisions potentially have long-term dependency with their past states, and decisions. Hence,

[1] Note that a naive approach of redefining the state to include the past states and actions will not work in practice, due to the variable lengths of trajectories, and potentially huge thus computationally infeasible state space.

all existing works relying on MDP models fail to capture such dependency.

In this paper, we make the first attempt to tackle the above challenge by developing Trajectory Generative Adversarial Imitation Learning (TrajGAIL) framework. It successfully captures the long-term decision dependency by modeling the human decision processes as variable length Markov decision processes (VLMDPs), and designs a deep neural network based framework to inversely learn the decision-making strategy from the human agent's historical dataset. *Our contributions* are summarized as follows:

- We formulate the human agent sequential decision-making process as a variable length Markov decision process (VLMDP) that explicitly models the "history", i.e., the past states and decisions, as a factor influencing the current decision (See Sec III-A).
- We develop a novel trajectory generative adversarial imitation learning (TrajGAIL) model to inversely learn human agents' decision-making strategy (See Sec III-B).
- We validate our framework using two real world human-generated spatial-temporal datasets, including a taxi trajectory dataset representing the taxi driver's passenger-seeking decision processes, and a public transit trip dataset, capturing the transit mode and stop choices for daily commuting. Results show significant accuracy improvement in human decision-making strategies, when comparing to baselines with Markov property assumption (See Sec IV). *We made our code and unique dataset available to contribute to the research community via an anonymous link [13].*

II. Overview

In this section, we define the human strategy learning problem and highlight the research challenges. For brevity, we present a table of notations in Table I.

TABLE I: Notations.

Notations	Descriptions
$\mathcal{S} = \{s\}$	State space.
$\mathcal{A} = \{a\}$	Action space.
$\mathcal{T} = \{\tau\}$	Trajectory set.
$\mathcal{H} = \{h_t\}$	History set.
$\pi(a_t \vert s_t, h_{t-1})$	Policy function.
$r(s_t, a_t \vert h_{t-1})$	Reward function.
$\pi_E(a_t \vert s_t, h_{t-1})$	Empirical policy from trajectory data.
$P(s_t \vert h_{t-1})$	Transition function.
γ	The discount factor.
η	Initial state distribution.
$\mathbf{W}^q$, $\mathbf{W}^k$, $\mathbf{W}^v$	Query, key and value matrices.
$\mathbf{X}$	Self-attention input.
$d_{\mathbf{X}}$	$\mathbf{X}$'s dimension.
$\mathbf{Q}$, $\mathbf{K}$, $\mathbf{V}$	Query, key and value of $\mathbf{X}$.
n_{head}	Multi-head self-attention head number.
N	Self-attention layer number.

A. Human-Generated Spatial-Temporal Data as Human Decision Trajectories

Human-generated spatial-temporal data (HSTD) capture sequential decisions made by human agents from their mobility. For example, taxi GPS trajectories represent the decisions from taxi drivers when completing a passenger-seeking task; trip data from automated fare collection devices on buses and trains infer the choices from passengers of which transit mode and transfer stops to choose for daily commute. As a result, HSTD can be viewed as a set of *trajectories*, where human agents traverse a series of *spatial-temporal states* by following a sequence of decisions (i.e., *actions*). We formally define these terms below.

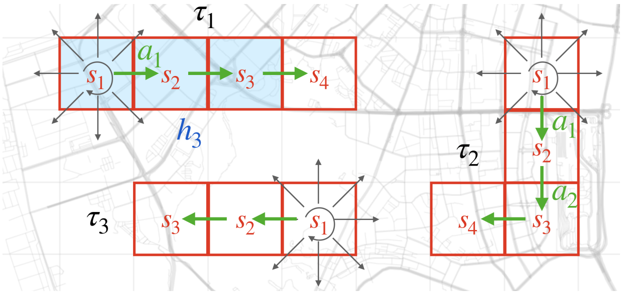


Fig. 2: Illustrations of state, action, trajectory and history.

Definition 1: A (spatial-temporal) state[2] s represents a location in latitude lat and longitude lng, and a time stamp t, namely, $s = \langle lat, lng, t \rangle$. Below, we will simply use state for spatial-temporal state for brevity.

Note that each (spatial-temporal) state is associated with a set of features, e.g., traffic speed and volume of the nearby area, characterizing what a human agent considers when making decisions.

Definition 2: An action a represents a decision a human agent makes at a state s, when completing a task in a geographic region. By following an action a, the human agent transits from s to s'. For example, when a taxi is vacant, the taxi driver can choose different directions as actions to go to find the next passenger.

Definition 3: A trajectory τ is a sequence of states and actions that a human agent traverses and takes, when completing a task in a geographic region, i.e., $\tau = (s_1, a_1, \cdots, s_T, a_T)$, where T is the length of τ. Furthermore, $\mathcal{T} = \{\tau_1, \cdots, \tau_m\}$ denotes a trajectory set, with m trajectories generated by human agents.

Definition 4: History h_{t-1}**.** Given a state s_t, i.e., the t-th state of a trajectory, $h_{t-1} = (s_1, a_1, \cdots, s_{t-1}, a_{t-1})$ represents the history of the trajectory at step t, including all states and actions prior to the t-th step.

An illustration example. Taking the passenger-seeking process as an example, a taxi driver traverses geographic regions over time to find the next passengers. When the taxi driver finds a passenger, a passenger-seeking trajectory is completed. Fig. 2 shows three trajectories τ_1, τ_2 and τ_3, each as a sequence of states s_i (as red blocks, representing the spatial-temporal region) and actions a_i (as green arrows, representing the moving directions the driver can choose). A history h_3 of trajectory τ_1 at step 4 is colored in blue as a sequence of states and actions prior to step 4.

B. Human Decision-Making Strategy

Each human agent has her own decision-making strategy to choose actions at different spatial-temporal states when completing a task. The human *decision-making strategy* can

[2] In defining a Markov decision process, some works use states [14], [15], and others use observations [16]. We use state to be consistent with [14], [15].

be characterized by two inherent functions (as defined below) of the human agents, namely i) *policy function* (probability distribution of actions) and ii) *reward function* (evaluating how effective a next action is).

Definition 5: Policy function $\pi(a_t|s_t, h_{t-1})$ of an agent characterizes the probability distribution to choose an action a_t at step t, given the state s_t and the trajectory history h_{t-1}. The policy function governs how a human agent makes decisions at different circumstances.

Definition 6: Reward function $r(s_t, a_t|h_{t-1})$ captures the "reward" the human agent receives when choosing an action a_t given the state s_t, and the history h_{t-1}.

For simplicity in notation, in policy and reward functions, the input state s represents both the spatial-temporal characteristics $\langle lat, lng, t\rangle$ and all the state features. Simply stated, a policy function controls how the agent chooses an action, and a reward function governs how the agent evaluates states and actions. Human agents are adapting policies for higher total reward when completing a task. As a result, each human agent possesses both policy and reward functions when making decisions, which together represent the *human decision-making strategy*.

C. Human Strategy Learning Problem

Problem Definition. Given a set of decision-making trajectories $\mathcal{T}$ generated by a human agent, we aim to inversely learn the decision-making strategy of the agent, namely, both policy function $\pi(a|s,h)$ and reward function $r(s,a|h)$.

Challenges. The proposed human strategy learning problem is challenging in two aspects: (**C1**) How to characterize the temporal dynamics of a human agent's decision-making strategy (as observed in Fig. 1)? (**C2**) Given policy and reward functions hinge on each other and are generally non-linear functions in nature, how to efficiently and accurately learn both jointly?

III. Methodologies

In this section, we solve the human strategy learning problem by modeling human agent decision-making process as a variable length Markov decision process (VLMDP) to capture the long-term dependency across states in a trajectory (i.e., tackling the challenge **C1**, See Sec III-A), and developing a trajectory generative adversarial imitation learning (TrajGAIL) framework in theory and implementation to jointly learn policy and reward functions using a generative adversarial net (GAN) [17] structure with self-attention mechanism (addressing challenge **C2**, See Sec III-B).

A. Modeling Human Sequential Decision-making Process as VLMDP

Limitations of the state-of-the-art works. There has been a rich literature studying human sequential decision-making processes, which are modeled as Markov decision processes (MDPs) [18]. A basic *Markov property assumption* [10] made with MDP is that each of action a_t made by a human agent only depends on the current state s_t, not on history h_{t-1}, namely, the policy and reward functions of the human agent follow $\pi(a_t|s_t, h_{t-1}) = \pi(a_t|s_t)$ and $r(s_t, a_t) = r(s_t, a_t|h_{t-1})$, respectively. If the Markov property holds, the policy and reward functions of the agents should be temporally invariant. However, as is shown in Fig. 1, the Markov property does not generally hold in real world human decision-making scenarios.

Human sequential decision-making processes as VLMDP. To capture the long-term (high-order) dependency of human decisions, we model the decision-making process as a *variable length Markov decision process* [19], represented as 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 $P(s_t|h_{t-1})$ as the probability of transitioning to state s_t by following history h_{t-1}; $r : \mathcal{S} \times \mathcal{A} \times \mathcal{H} \mapsto \mathbb{R}$ is the bounded reward function that outputs a reward value for a given state-action-history triple; $\gamma \in [0,1]$ is a discount factor, that discounts the future reward exponentially. The initial states are determined by the distribution $\eta : \mathcal{S} \mapsto [0,1]$. When an agent takes an action at time t, it considers the current state s_t and the history $h_{t-1} = (s_1, a_1, \cdots, s_{t-1}, a_{t-1})$, i.e., all states and actions prior to time t. We denote the set of histories as $\mathcal{H}$, i.e., $h_t \in \mathcal{H}$. Specifically, actions are chosen through a stationary and stochastic policy $\pi : \mathcal{S}\times\mathcal{H} \mapsto [0,1]$. A decision-making process forms a trajectory [3] $\tau = (s_1, a_1, \cdots, s_T, a_T)$, where T is the terminal time step, and the set of all trajectories is denoted as $\mathcal{T} = \{\tau\}$. We use expectation with respect to a policy π to denote an expectation with respect to the trajectories it generates. For instance, $\mathbb{E}_\pi[r(s,a|h)] = \mathbb{E}_{s_t,h_{t-1},a_t\sim\pi}[\sum_{t=1}^{T}\gamma^t r(s_t, a_t|h_{t-1})]$, denotes the following sample process as $s_1 \sim \eta$, $a_t \sim \pi(\cdot|s_t, h_{t-1})$, $s_t \sim P(s_t|h_{t-1})$ and each agent aims to maximize its expected reward $\mathbb{E}_\pi[r(s,a|h)]$.

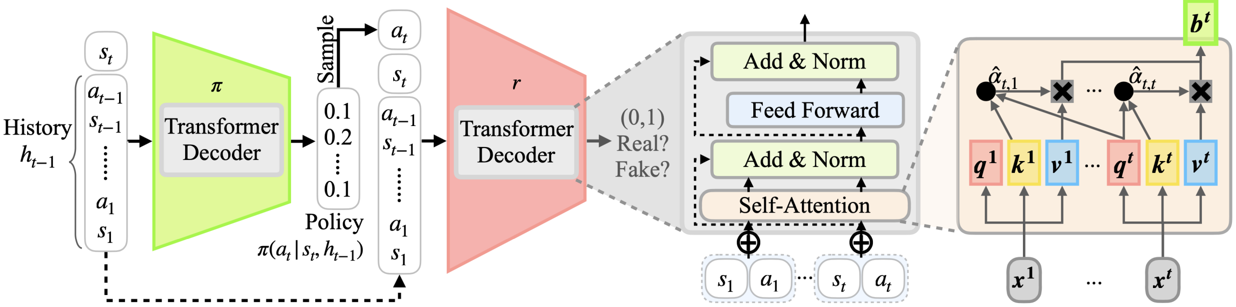
Based on the VLMDP model, we will formulate the human strategy learning problem and solve it by developing trajectory adversarial imitation learning framework.

B. TrajGAIL: Trajectory Generative Adversarial Imitation Learning

1) Theory: Now, we formally formulate the human strategy learning problem based on VLMDP model, and develop theoretical solution to the problem.

Limitation of the state-of-the-art works. In the literature, human strategy learning problem, namely learning policy and reward functions from human demonstration data, has been extensively studied as apprenticeship learning, maximum entropy inverse reinforcement learning (MaxEnt IRL) [20]–[23] and generative adversarial imitation learning (GAIL) [24], [25], etc. However, all these works were based on modeling human decisions as MDPs, thus fail to capture the high order decision dependency (as observed from our data in Fig. 1 and many other real world scenarios). To tackle this challenge, we develop trajectory generative adversarial imitation learn-

[3] In this paper we use "trajectory" to refer to both the physical trace of a human agent from HSTD and the state-action pair sequences of an agent in the VLMDP model because each physical "trajectory" can be mapped to a sequence of states and actions so the two concepts are equivalent in our problem.



(a) TrajGAIL architecture. **(b) Transformer decoder.** **(c) Self-attention.**

Fig. 3: A detailed illustration of the TrajGAIL structure.

ing below, by theoretically extending GAIL model to adapt VLMDP.

Trajectory Generative Adversarial Imitation Learning. The human strategy learning problem can be formulated as **P1** below. Note that **P1** follows the same maximum causal entropy principle as MaxCausalEnt IRL [23] and GAIL [24], but explicitly incorporates the long-term decision dependency, by adapting VLMDP model, rather than MDP.

P1: Strategy learning problem with decision dependency:

$$\max_r \min_\pi : \ -H(\pi), \tag{1}$$

$$\text{s.t.} : \mathbb{E}_\pi[r(s,a|h)] = \mathbb{E}_{\pi_E}[r(s,a|h)], \tag{2}$$

$$\sum_{a\in\mathcal{A}} \pi(a|s,h) = 1, \qquad \forall s \in \mathcal{S}. \tag{3}$$

The objective in eq. (1) is γ-discounted causal entropy, i.e., $H(\pi) = \sum_{t=1}^{T}\sum_{h_t}\gamma^t \pi(a_t|s_t,h_{t-1})\log\pi(a_t|s_t,h_{t-1})$, which measures the uncertainty present in a causally conditioned policy distribution $\pi(a_t|s_t,h_{t-1})$. The constraint in eq. (2) guarantees that the expected reward of a trajectory under the learned policy π, matches that of the empirical policy π_E (namely, the policy observed from the collected HSTD data). The constraint in eq. (3) ensures the policy π to be a proper probability distribution at each state. There are infinite many feasible π's and r's satisfying constraints eq. (2) and eq. (3) [20]. To break the tie, the objective function aims to *i)* find the policy function π with the maximum uncertainty, namely, without committing to any particular trajectory than what the two constraints require, and *ii)* the reward function r that enforces the constraint eq. (2) to hold.

The Problem **P1** can be relaxed to the unconstrained optimization problem below by introducing a Lagrangian multiplayer λ and a convex penalty function $\psi(r)$ using augmented Lagrangian method [26].

$$\max_r \min_{\pi\in\Pi} -\psi(r) - \lambda H(\pi) + \mathbb{E}_{\pi_E}[r(s,a|h)] - \mathbb{E}_\pi[r(s,a|h)], \tag{4}$$

where Π is the policy probability simplex space [27], guaranteeing the constraint eq. (3). The convex penalty function $\psi(r)$ characterizes the gap between the expected rewards under π_E vs π, namely, $\mathbb{E}_{\pi_E}[r(s,a|h)] - \mathbb{E}_\pi[r(s,a|h)]$, and the reward function r is chosen to minimize this gap ($\psi(r)$), such that the original constraint eq. (2) holds. Next, We establish Theorem III.1 below to prove the equivalence between **P1** and a generative adversarial network (GAN) problem [17].

Theorem III.1. *With a proper choice of ψ,* **P1** *problem is equivalent to eq. (5) below:*

$$\min_{\pi\in\Pi}\max_r -\lambda H(\pi) + \mathbb{E}_{\pi_E}[\log(r(s,a|h))] + \mathbb{E}_\pi[\log(1-r(s,a|h))]. \tag{5}$$

Proof (Sketch). We first define the conditional occupancy measure ρ_π of a policy π as

$$\rho_\pi(s_t,a_t|h_{t-1}) = \gamma^t\pi(a_t|s_t,h_{t-1})P(s_t|h_{t-1}),$$

which represents the probability distribution of a state-action pair (s_t,a_t) at step t, when following h_{t-1} under the policy π. We proved that (See Appx. B) eq. (4) is the dual problem of

$$\min_{\pi\in\Pi} -H(\pi) + \psi^*(\rho_{\pi_E} - \rho_\pi), \tag{6}$$

where ψ^* is the conjugate of ψ. Moreover, we proved (in Appx. B) that with a proper choice of ψ, we get

$$\psi^*(\rho_{\pi_E} - \rho_\pi) = \max_r \mathbb{E}_{\pi_E}[\log(r(s,a|h))] + \mathbb{E}_\pi[\log(1-r(s,a|h))]. \tag{7}$$

Combining eq. (6) and eq. (7) completes the proof. □

Clearly, Theorem III.1 indicates that (under mild conditions) the strategy learning problem **P1** can be explained as a GAN problem, thus can be solved using GAN framework, with policy net π as the generator net and reward net r as the discriminator net. We will introduce a trajectory generative adversarial imitation learning (TrajGAIL) framework below for practically solving the strategy learning problem **P1**.

2) Implementation and Architecture: In this section, we describe the architecture of TrajGAIL for human strategy learning with long term decision dependency, as shown in Fig. 3. It consists of *i)* a GAN structure based on the solution formulation in Theorem III.1 as shown in Fig 3(a), and *ii)* a transformer decoder structure in both policy and reward networks to capture the long-term decision dependency, i.e., $\pi(a|s,h)$ and $r(s,a|h)$. We will detail these two key components below.

i) Learning decision-making strategy with GAN structure. As is shown in Fig. 3(a), TrajGAIL is composed of a reward net r as a discriminator and a policy net π as a generator. Given previous history h_{t-1} and current state s_t, the policy π outputs an action distribution following $\pi(a_t|s_t,h_{t-1})$. Then an action is sampled from this distribution to be combined with h_{t-1} and s_t as a new state-action-history tuple. This new tuple is feed into the reward net r as negative samples to be distinguished from expert demonstrated state-action-history tuples. We had similar observations as GAIL [24], that there is no significant performance difference when changing Lagrangian multiplier λ. Hence, we use no causal entropy regularization in experiments.

ii) Capturing long-term decision dependency with transformer decoder network. To capture the long-term decision dependency, we employ transformer decoder [28] as the network structure in both policy net π and reward net r. The reason of choosing transformer decoder is that other state-of-the-art deep neural network structures, such as recurrent neural networks (RNN) [29], gated recurrent units (GRUs) [30], long-short term memory (LSTM) [31] and convolutional LSTM (ConvLSTM) [32], are less efficient in long-term memorization because of memory decay [33].

As shown in Fig. 3(b), a transformer decoder is composed of N layers of multi-head self-attention, add & layer normalization, and feed forward networks. The architecture in Fig. 3(b) shows one such layer, i.e., $N = 1$. Next, we detail the structure of the self-attention module in the transformer decoder.

Self-attention module. The self-attention module takes state-action pair sequence $(s_1, a_1, \cdots, s_t, a_t)$ as input, and converts it into $\mathbf{X} = [\mathbf{x}^1, \cdots, \mathbf{x}^t]$, with each $\mathbf{x}^i = (s_i, a_i, i) \in \mathbb{R}^{d_\mathbf{X}}$, namely, attaching the sequence ID i to each state-action pair (s_i, a_i), with $d_\mathbf{X}$ as the dimension of each $\mathbf{x}^i$. The self-attention module can be implemented in either single- or multi-head mode to process $\mathbf{X}$ as detailed below.

- *A single-head self-attention module* shown in Fig. 3(c) has three matrices as a query matrix $\mathbf{W}^q \in \mathbb{R}^{d_\mathbf{X} \times d_\mathbf{X}}$, a key matrix $\mathbf{W}^k \in \mathbb{R}^{d_\mathbf{X} \times d_\mathbf{X}}$ and a value matrix $\mathbf{W}^v \in \mathbb{R}^{d_\mathbf{X} \times d_\mathbf{X}}$. Input units $[\mathbf{x}^1, \cdots, \mathbf{x}^t]$ are multiplied by the query, key, value matrices, respectively to produce matrices $\mathbf{Q} = [\mathbf{q}^1, \cdots, \mathbf{q}^t]$ with $\mathbf{q}^i = \mathbf{W}^q \mathbf{x}^i$, $\mathbf{K} = [\mathbf{k}^1, \cdots, \mathbf{k}^t]$ with $\mathbf{k}^i = \mathbf{W}^k \mathbf{x}^i$, and $\mathbf{V} = [\mathbf{v}^1, \cdots, \mathbf{v}^t]$ with $\mathbf{v}^i = \mathbf{W}^v \mathbf{x}^i$. The attention module attends the t-th query output $\mathbf{q}^t$ to every key output $\mathbf{k}^i$ in $\mathbf{K}$, which are further used to compute a weighted sum of all value outputs $\mathbf{v}^i \in \mathbf{V}$, leading to an output $\mathbf{b}^t$ as

$$\mathbf{b}^t = \mathrm{attn}(\mathbf{q}^t, \mathbf{K}, \mathbf{V}) = \mathbf{V}^\intercal \mathrm{softmax}(\frac{\mathbf{K}\mathbf{q}^t}{\sqrt{d_\mathbf{X}}}), \tag{8}$$

where $\mathrm{softmax}(\cdot)$ is a column-wise softmax operator.

- *A multi-head self-attention module (with $n_{head} > 1$ heads)* runs through the single-head self-attention module n_{head} times in parallel. Each input vector $\mathbf{X}^i \in \mathbf{X}$ is chopped into n_{head} pieces, with $d_\mathbf{X}/n_{head}$ dimensions, and gets processed by n_{head} single-head self-attention modules in parallel. The n_{head} parallel outputs are simply concatenated, denoted as $\mathbf{b}^t = [\mathbf{b}^t_1, \cdots, \mathbf{b}^t_{n_{head}}]$ and linearly transformed by a matrix $\mathbf{W}^O \in \mathbb{R}^{d_\mathbf{X} \times d_\mathbf{X}}$ as below:

$$\mathrm{MultiHead}(\mathbf{q}^t, \mathbf{K}, \mathbf{V}, \mathbf{W}^O) = \mathrm{Concat}(\mathrm{attn}_1; \cdots; \mathrm{attn}_{\mathrm{n_{head}}})\mathbf{W}^O, \tag{9}$$

$$\mathrm{attn_j} = \mathrm{attn}(\mathbf{q}^t_j, \mathbf{K}_j, \mathbf{V}_j), \qquad 1 \leq j \leq n_{head}. \tag{10}$$

Here, matrices $\mathbf{Q}_j, \mathbf{K}_j, \mathbf{V}_j$ $(1 \leq j \leq n_{head})$ and $\mathbf{W}^O$ are parameters to be learned.

The transformer decoder's self-attention layer applies a multi-headed self-attention operation over the input $\mathbf{X}$ followed by position-wise feed-forward layers to produce an output distribution (i.e., policy in π or reward score in r) over target input units. Since self-attention and transformer decoder allow parallelization, the time complexity is significantly reduced.

Algorithm 1 TrajGAIL

```
Require: Initial parameters of policy and reward nets, $\theta$ and
    $\omega$; expert demonstrations $\mathcal{T} = \{(s_j, a_j)\}_{j=1}^T$; batch size
    $B$; variable length Markov decision process as a block
    box $(\mathcal{S}, \mathcal{A}, P, \eta, r, \gamma)$.
Ensure: A learned policy $\pi_\theta$ and a reward functions $r_\omega$.
1: for each epoch $i = 0, 1, 2, ...$ do
2:     Generate trajectories of batch size $B$ from $\pi^i$ through
       the process: $s_1 \sim \eta$, $a \sim \pi^i(\cdot|s_t, h_{t-1}), s_{t+1} \sim$
       $P(s_{t+1}|h_t)$; denote the generated trajectory set as $\tilde{\mathcal{T}}$.
3:     Sample state-action-history sequences from $\tilde{\mathcal{T}}$ and $\mathcal{T}$
       each with batch size $B$ denoted as $\tilde{\mathcal{D}}$ and $\mathcal{D}$.
4:     Update $\omega$ to increase the objective in eq. 5.
5:     Compute reward $r$ for state-action-history sequences
       $(s, a, h) \in \tilde{\mathcal{D}}$ using $r_{\omega_{i+1}}$.
6:     Update $\theta$ by policy gradient to decrease the objective
       in eq. 5.
7: end for
8: Return the learned policy $\pi_\theta$ and reward function $r_\omega$.
```

TrajGAIL algorithm. Now, we are in a position to present TrajGAIL algorithm (Alg 1) to inversely learn both policy and reward networks using the TrajGAIL architecture (Fig. 3).

In Alg 1, both π and r are represented using neural networks: TrajGAIL fits a parameterized policy network π_θ, with weights θ, and a reward network $r_\omega : \mathcal{S} \times \mathcal{A} \times \mathcal{H} \mapsto (0, 1)$, with weights ω. TrajGAIL utilizes the Adam [34] gradient step on ω to increase eq. 5 with respect to r, and the Trusted Region Policy Optimization (TRPO) [35] step on θ to decrease eq. (5) with respect to π. In each epoch i, we recursively input the policy π^i with previous history and current state to get an action sampled which leads to the next state following transition $P(s_{t+1}|h_t)$ working as the next iteration's input. Following this recursion, B trajectories are generated and denoted as $\tilde{\mathcal{T}}$, where the initial states are sampled from η (Line 2). With generated trajectories $\tilde{\mathcal{T}}$ and expert demonstrated trajectories $\mathcal{T}$, we partition each trajectory in the two trajectory sets as state-action-history sequences forming two new sets – generated state-action-history sequence set $\tilde{\mathcal{D}}$ and demonstrated state-action-history sequence set $\mathcal{D}$ respectively; then, we sample the state-action-history sequences from the two partitioned sets each with a batch size of B (Line 3). We update the reward net's parameters ω with an Adam [34] step following eq. (5) (Line 4). With the current reward r^i, we evaluate the reward for generated state-action history sequences $(s, a, h) \in \tilde{\mathcal{D}}$ (Line 5). Finally, we update θ with a TRPO step to decrease objective in eq. (5) (line 6).

IV. Experiment

In this section, we evaluate the performances of TrajGAIL framework using two real-world human-generated spatial-temporal datasets. One is taxi trajectory dataset representing the taxi driver's passenger-seeking decision processes. The other is public transit trip dataset, capturing the transit mode and stop choices for daily commuting.

A. Data Description and Preparation

Taxi trajectory data were collected from July to September in 2016 in Shenzhen, China, which contains GPS records from 17,877 taxis. On average, a GPS record is generated every 30 seconds. Every GPS record includes five attributes: 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. The passenger-seeking trajectories are consecutive GPS records with passenger indicator being 0.

•*State space.* We divide the time in a day into five-minutes intervals and partition the road map into equal side-length grid cells. Thus, a spatial grid cell and a five-minutes interval uniquely define a state, which characterizes where the taxi is and what time it is in a day.

•*State features.* Given a spatial-temporal state, taxi drivers potentially consider many features of the surrounding area when deciding where to find the next passenger. We extract four traffic features including traffic speed, traffic volume, travel demand, and waiting time of the state grid cell, and its neighboring 5×5 grid cells. Traffic volume characterizes the average number of taxis in a state s from the historical data showing how congested a state is. A higher traffic volume is likely to indicate a heavy traffic, and a lower one show a light traffic. Traffic speed 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 captures the average time a taxi stays in the target state s from the historical taxi trajectory data.

•*Action space.* When a taxi is at a spatial-temporal state, the driver has 10 actions to choose, including 8 neighboring grid cells, staying at the current grid cell, or terminating the trip.

Public transit trajectory data were collected in Shenzhen, China from June to December in 2016 from the automatic fare collection (AFC) systems equipped in buses and subways. Once a passenger swipes her smart card at an AFC device to get on board a bus or enter/leave a subway station, we are able to get their travel information including five attributes - passenger ID, transaction type, cost, transaction time, transit station/stop name and location. The transaction type field indicates if it is an event of getting on a bus, or leaving/entering a subway station. Most trips contain 1 to 4 transits and the average of transits is 1.1.

•*State space.* Similar to taxi passenger-seeking scenario, a spatial grid cell and a five-minutes interval define a state.

•*State features.* A passenger may consider various features of the surrounding areas when deciding which subway line or bus route to take to reach the destination. The features we considered for this study include monetary cost, time cost and level of convenience. Monetary calculates how much the traveler needs to pay of taking action a at state s, e.g. the fare of taking a subway or a bus. Time cost measures how much time the traveler needs to spend of taking action a at state s. Level of convenience reflects how convenient an action a is. It measures the number of transfers needed, the number of other transit choices available and the transit mode of the action a at state s.

TABLE II: Method architecture designs. (CNN – convolutional neural network [38]; LSTM – long short-term memory [31]; TD – transformer decoder [28], [39].)

	GAIL	GAIL$_l$	LSTM-GAIL	TrajGAIL
π	CNN	CNN	LSTM	TD
r	CNN	CNN	LSTM	TD

•*Action space.* The action space include all transit modes and routes a passenger agent can choose, e.g., a certain bus route or subway line with transfer or destination stations.

B. Experiment Settings

In this section, we detail the experiment settings including the baseline methods and evaluation metrics. We randomly split each dataset into three parts: *training set* (80%), *validation set* (10%) and *test set* (10%). We present performance results from the test set in this section [4].

Baselines. We compare our proposed TrajGAIL with baselines below, with Table II showing the detailed neural network architecture designs [5]:

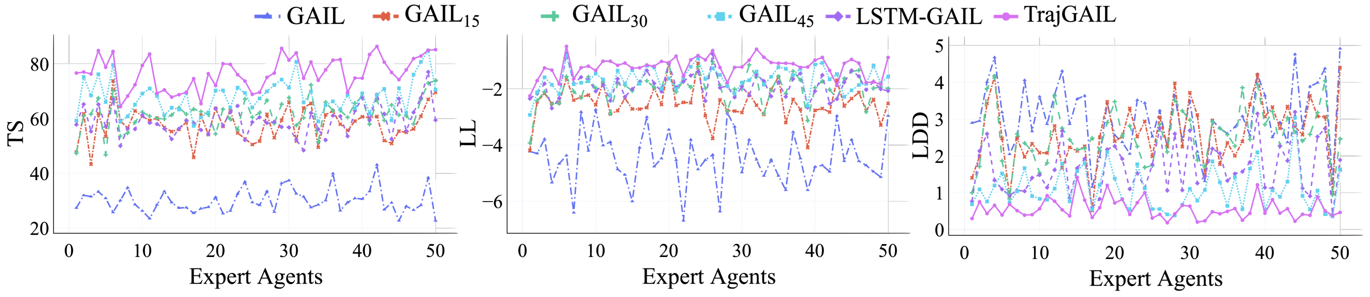
- **GAIL** [24]: The inputs in the policy net and reward net are state s and state-action pair (s, a) respectively;
- **GAIL**$_\ell$ [24]: It employs the same model structure as GAIL, but redefines the state $\tilde{s}$ by including a fixed ℓ-length history, i.e., $\tilde{s}_t = (s_j, a_j)_{j=t-\ell}^{\ell}$. By introducing a fixed length history into the state, it explicitly models the decision dependency within a fixed length history, however, at a cost of storage space and processing time for states. We choose different ℓ's in the evaluations.
- **LSTM-GAIL**: This baseline implements the policy net π and reward net r using long-short term memory (LSTM) [31]. LSTM is one form of recurrent neural network (RNN) that does not suffer from vanishing and exploding gradient problem [38]. It processes a variable length sequence by incrementally adding new input into a single memory unit and control the extent to which new content should be memorized, old content should be erased, and current content should be exposed using gates.

Evaluation metrics. We use four metrics below to evaluate the performance of TrajGAIL frameworks, including i) trajectory sharing percentage, ii) test trajectory log likelihood iii) length distribution difference and iv) time complexity. Below we detail all four evaluation metrics:

- **Trajectory sharing percentage (TS).** With the learned policy network π, we generate a trajectory set $\tilde{\mathcal{T}}$ and compare it with the human agent generated trajectory set $\mathcal{T}$. Trajectory sharing percentage evaluates the ratio of the shared route distance and the total route distance of trajectories in $\tilde{\mathcal{T}}$ and

[4] Data and code for TrajGAIL and baseline methods are available in [13].

[5] Note that we do not include linear reward function based strategy learning algorithms as baselines, such MaxEntIRL [22], MaxCausalEntIRL [23], RelEntIRL [36], etc, because linear reward function approximation generally fails to capture complex real-world reward functions devised by human agents [9], [37].

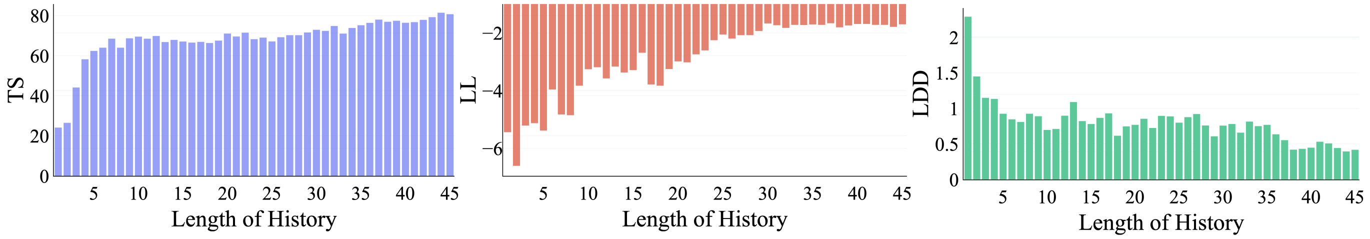


(a) Trajectory sharing percentage (TS).

(b) Test trajectory log likelihood (LL).

(c) Length distribution difference (LDD).

Fig. 4: Passenger-seeking policy inference performance.



(a) Trajectory sharing percentage (TS).

(b) Test trajectory log likelihood (LL).

(c) Length distribution difference (LDD).

Fig. 5: The performances of GAIL$_\ell$ over different length of history ℓ's.

those in $\mathcal{T}$. Clearly, a higher TS indicates a better ability to mimic the behaviors of human agents.

- **Test trajectory log likelihood (LL).** We divide the human generated trajectory set into training, validation and testing sets. LL evaluates the average log probability of trajectories in testing set under the learned policy π. A higher LL indicates a closer policy to that of the human agent.
- **Length distribution difference (LDD)** is a measure that calculates the Kullback–Leibler divergence distance [40] between the length distribution of trajectories generated by the learned policy π vs those of human generated trajectories.
- **Processing time (PT)** measures the time cost for each training epoch in seconds, indicating the time complexity of training process of a given learning structure.

C. Results with Passenger-seeking Dataset

Fig. 4(a)-(c) show the performance comparison results of TrajGAIL to baseline methods in learning the passenger-seeking strategy of taxi drivers. In particular, we choose the fixed history length ℓ to 15, 30, 45 in the baseline model GAIL$_\ell$. We randomly choose 50 taxi drivers (as x-axis) to show the results.

Clearly, TrajGAIL always outperforms baselines for all taxi drivers. To be precise, TrajGAIL has the highest trajectory sharing percentage on average of 76.68%, i.e., 8.72% - 46.61% higher than other baselines (from Fig. 4(a)), the highest test trajectory log likelihood on average about -1.21, i.e., 26.99% -72.88% higher than other baselines (from Fig. 4(b)), and the lowest length distribution difference on average of 0.56, i.e., 48.30% - 81.56% lower than other baselines (from Fig. 4(c)). Moreover, GAIL works the worst among all baselines, because it follows Markov property in design, and completely ignores the decision dependency. However, the passenger-seeking decision-making process is complex and involves long-term decision dependency that GAIL fails to characterize. The performance gets improved in GAIL$_\ell$ with increasing history length ℓ. This is because a large ℓ allows GAIL$_\ell$ to capture more decision dependency. On the other hand, LSTM-GAIL yeilds performances in-between GAIL$_{15}$ and GAIL$_{45}$, because LSTM-GAIL is capable of capturing relatively short-term decision dependency, but not long-term, due to the memory decay [33].

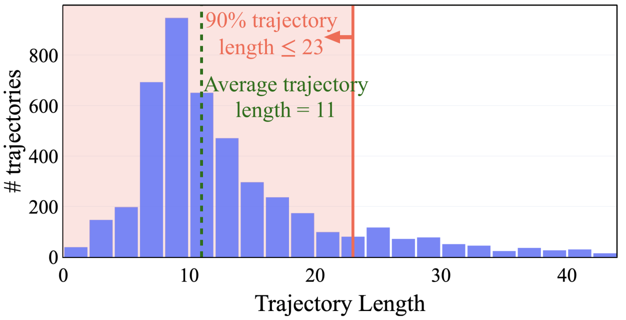


Fig. 6: Trajectory length distribution in passenger-seeking data.

Evaluating the long-term decision dependency in passenger-seeking scenario. Now, we further look into the taxi passenger-seeking decision data, and evaluate how long the decision dependency lasts along the decision-making trajectories. We examine the performances by changing the history length $\ell \in [1, 45]$ in GAIL$_\ell$, where 45 the maximum length of trajectories from our dataset. Fig. 5 shows that with the increasing amount of history information carried in a state (shown in the x-axis), the LDD score drops and TS and LL increases. This indicates that the taxi drivers' decisions are highly dependent on all previous steps along the trajectory.

However, it is infeasible to implement GAIL$_\ell$ in practice

TABLE III: Performances from transit trip data.

	GAIL	GAIL$_9$	LSTM-GAIL	TrajGAIL
TS	64.58%	74.04%	73.90%	**76.25%**
LL	-1.05	-0.23	-0.25	**-0.22**
LDD	6.16	3.42	3.62	**2.55**
PT (s)	**2.30**	4.02	13.09	3.56

due to two reasons: i) it is hard to pre-define the fixed length ℓ, since the maximum length may change over time; ii) it is a waste of storage space in state variable, since trajectories have variable lengths and most are with much shorter length than the maximum. This is validated in Fig. 6 showing the trajectory length distribution from passenger-seeking data. Overall, the average trajectory length is about 11, worth of 55 minutes, and more than 90% trajectories with length less than 23, worth of 115 minutes.

D. Results with Transit Trip Dataset

Table III shows the results of learning passengers' strategies on public transit mode selection, using a large set of public transit trip data including subway trains and buses. The results show that TrajGAIL works the best in learning the decision policies from human commuters, with the highest trajectory sharing percentage and test trajectory log likelihood, and the smallest length distribution difference. Comparing to baselines, TrajGAIL increases the trajectory sharing percentage by 2.21% – 11.67%, test trajectory log likelihood by 4.35%–79.04%, and decreases the length difference distribution by 25.43%–58.60%. In terms of the processing time, TrajGAIL is a little bit higher than GAIL, but is much lower than GAIL$_\ell$, and LSTM-GAIL. Overall, the performance improvement over baselines is great but not as significant as that in passenger-seeking scenario. This is because the trajectory length in transit trip data is smaller, say, with an average length of 1.1 in transfers between different transit modes/routes, while the trajectory length in passenger-seeking scenario is on average 11 as shown in Fig. 6. As a result, the effort of decision dependency in transit selection scenario is weaker than that of passenger-seeking scenario.

V. Related work

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

Human Decision Analysis aims to provide logical and systematic analysis on the representation of decision-maker's information and preferences towards uncertain, complex and dynamics features of the decision problem to enhance effective decision making [41], [42]. It has been applied in many areas such as business [43], public health [44] and urban computing [45] with human generated behavior data. Human decision analysis using human behavior data has been applied in various problems. In taxi operation management, works are focusing on dispatching [46], [47], and passenger seeking [8], [9], [48]–[51]. In public transportation selection, [6] focuses on commuter's daily transit planning and [52] works on traveler's travel mode preferences. These works target finding an optimal actionable solution to improve the performance/revenue of individual decision-makers. However, all of these works focus on finding the optimal strategies with specific problems. By contrast, our work focuses on providing a unifying framework to model and learn the decision-making strategies from human generated spatial-temporal data, that are applicable to a wide range of scenarios such as taxi driver passenger-seeking processes, commuter transit mode choice processes, food delivery work decision processes, etc.

Inverse Reinforcement Learning (Imitation Learning) aims to inversely learn the reward function and policy of experts from their demonstrations [22]–[25], [36], [53]–[55], which models the agent's decision making process as Markov Decision Processes (MDPs). Based on such Markov perproty assumption, MaxEnt IRL [22], MaxCausalEnt IRL [23], and RelEnt IRL [36] 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 [24], [56] extends the above approaches (especially MaxCausalEnt IRL) to general non-linear reward function by using generative adversarial networks (GANs) framework. Moreover, [25] applies GAIL model in applications, such as autonomous driving. None of these works *explicitly* capture the *long-term* decision dependency of real world human decision-making processes. We develop a trajectory generative adversarial imitation learning framework (TrajGAIL) to address this challenge.

VI. Conclusion

In this paper, we propose a trajectory generative adversarial imitation learning (TrajGAIL) framework that inversely learn the human decision-making strategy (as policy and reward functions) from their generated spatial-temporal data (HSTD). Unlike existed models which makes Markov property assumption on human decision-making processes, we show that such Markov Property does not hold in real world scenarios, and develop TrajGAIL to i) explicitly capture the decision dependency, and ii) jointly learn both reward and policy functions as two deep neural networks. Moreover, we evaluate the performances of TrajGAIL framework using two real-world human-generated spatial-temporal datasets, including taxi driver passenger-seeking decision data and a public transit trip data. When comparing to baselines with Markov property assumption, our results show significant accuracy improvement in learning human decision-making strategies.

VII. ACKNOWLEDGEMENTS

Xin Zhang and Yanhua Li were supported in part by NSF grants IIS-1942680 (CAREER), CNS-1952085, CMMI-1831140, and DGE-2021871. Ziming Zhang was supported in part by NSF CCF-2006738. Xun Zhou was funded partially by Safety Research using Simulation University Transportation Center (SAFER-SIM). SAFER-SIM is funded by a grant from the U.S. Department of Transportation's University Transportation Centers Program (69A3551747131). However, the U.S. Government assumes no liability for the contents or use thereof.

References

[1] DoorDash, "Doordash services." https://www.doordash.com/, 2019.

[2] Postmates, "Postmates services." https://postmates.com/, 2019.

[3] TaskRabbit, "Taskrabbit services." https://www.taskrabbit.com/, 2019.

[4] Z. Xu, Z. Li, Q. Guan, D. Zhang, Q. Li, J. Nan, C. Liu, W. Bian, and J. Ye, "Large-scale order dispatch in on-demand ride-hailing platforms: A learning and planning approach," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 905–913, 2018.

[5] J. Bao, T. He, S. Ruan, Y. Li, and Y. Zheng, "Planning bike lanes based on sharing-bikes' trajectories," in *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1377–1386, 2017.

[6] 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.

[7] M. Yang, Y. Li, X. Zhou, H. Lu, Z. Tian, and J. Luo, "Inferring passengers' interactive choices on public transits via ma-al: Multi-agent apprenticeship learning," in *2020 The Web Conference (WWW)*, 2020.

[8] M. Pan, Y. Li, X. Zhou, Z. Liu, S. Rui, H. Lu, and J. Luo, "Dissecting the learning curve of taxi drivers: A data-driven approach," in *2019 SIAM International Conference on Data Mining*, SDM, 2019.

[9] 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.

[10] R. Durrett, *Probability: theory and examples*, vol. 49. Cambridge university press, 2019.

[11] E. A. Locke and G. P. Latham, "Work motivation and satisfaction: Light at the end of the tunnel," *Psychological science*, vol. 1, no. 4, pp. 240–246, 1990.

[12] R. F. Baumeister, "Ego depletion and self-control failure: An energy model of the self's executive function," *Self and identity*, vol. 1, no. 2, pp. 129–136, 2002.

[13] TrajGAIL, "Trajectory generative adversarial imitation learning." https://github.com/TrajGAIL/TrajGAIL, 2020.

[14] X. Liang, T. Wang, L. Yang, and E. Xing, "Cirl: Controllable imitative reinforcement learning for vision-based self-driving," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 584–599, 2018.

[15] G. Li, M. Mueller, V. Casser, N. Smith, D. L. Michels, and B. Ghanem, "Oil: Observational imitation learning," *arXiv preprint arXiv:1803.01129*, 2018.

[16] J. Hawke, R. Shen, C. Gurau, S. Sharma, D. Reda, N. Nikolov, P. Mazur, S. Micklethwaite, N. Griffiths, A. Shah, *et al.*, "Urban driving with conditional imitation learning," *arXiv preprint arXiv:1912.00177*, 2019.

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

[19] P. Bühlmann, A. J. Wyner, *et al.*, "Variable length markov chains," *The Annals of Statistics*, vol. 27, no. 2, pp. 480–513, 1999.

[20] A. Y. Ng, S. J. Russell, *et al.*, "Algorithms for inverse reinforcement learning.," in *Icml*, vol. 1, p. 2, 2000.

[21] U. Syed and R. E. Schapire, "A game-theoretic approach to apprenticeship learning," in *Advances in neural information processing systems*, pp. 1449–1456, 2008.

[22] 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.

[23] B. D. Ziebart, J. A. Bagnell, and A. K. Dey, "Modeling interaction via the principle of maximum causal entropy," 2010.

[24] J. Ho and S. Ermon, "Generative adversarial imitation learning," in *NeurIPS*, pp. 4565–4573, 2016.

[25] T. Fernando, S. Denman, S. Sridharan, and C. Fookes, "Learning temporal strategic relationships using generative adversarial imitation learning," in *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 113–121, International Foundation for Autonomous Agents and Multiagent Systems, 2018.

[26] C. Wu and X.-C. Tai, "Augmented lagrangian method, dual methods, and split bregman iteration for rof, vectorial tv, and high order models," *SIAM Journal on Imaging Sciences*, vol. 3, no. 3, pp. 300–339, 2010.

[27] W. Wang and M. A. Carreira-Perpinán, "Projection onto the probability simplex: An efficient algorithm with a simple proof, and an application," *arXiv preprint arXiv:1309.1541*, 2013.

[28] P. J. Liu, M. Saleh, E. Pot, B. Goodrich, R. Sepassi, L. Kaiser, and N. Shazeer, "Generating wikipedia by summarizing long sequences," *arXiv preprint arXiv:1801.10198*, 2018.

[29] T. Mikolov, "Statistical language models based on neural networks," *Presentation at Google, Mountain View, 2nd April*, vol. 80, 2012.

[30] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *arXiv preprint arXiv:1409.1259*, 2014.

[31] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[32] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting advances in neural information processing systems. 802–810," 2015.

[33] Y. Su and C.-C. J. Kuo, "On extended long short-term memory and dependent bidirectional recurrent neural network," *Neurocomputing*, vol. 356, pp. 151–161, 2019.

[34] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[35] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *International conference on machine learning*, pp. 1889–1897, 2015.

[36] A. Boularias, J. Kober, and J. Peters, "Relative entropy inverse reinforcement learning," in *AISTATS*, pp. 182–189, 2011.

[37] B. Ibarz, J. Leike, T. Pohlen, G. Irving, S. Legg, and D. Amodei, "Reward learning from human preferences and demonstrations in atari," in *Advances in neural information processing systems*, pp. 8011–8023, 2018.

[38] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[39] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, pp. 5998–6008, 2017.

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

[41] P. Goodwin and G. Wright, *Decision Analysis for Management Judgment 5th ed.* 2014.

[42] R. A. Howard, "An assessment of decision analysis," *Operations Research*, vol. 28, no. 1, pp. 4–27, 1980.

[43] R. T. Clemen, *Making hard decisions: an introduction to decision analysis*. Brooks/Cole Publishing Company, 1996.

[44] M. C. Weinstein, B. O'Brien, J. Hornberger, J. Jackson, M. Johannesson, C. McCabe, and B. R. Luce, "Principles of good practice for decision analytic modeling in health-care evaluation: report of the ispor task force on good research practices—modeling studies," *Value in health*, vol. 6, no. 1, pp. 9–17, 2003.

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

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

[47] 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.

[48] 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.

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

[50] 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.

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

[52] B. Verplanken, H. Aarts, A. Van Knippenberg, and C. van Knippenberg, "Attitude versus general habit: Antecedents of travel mode choice 1," *Journal of applied social psychology*, vol. 24, no. 4, pp. 285–300, 1994.

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

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

[55] J. Fu, K. Luo, and S. Levine, "Learning robust rewards with adversarial inverse reinforcement learning," *arXiv preprint arXiv:1710.11248*, 2017.

[56] 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.

Appendix

A. Conditional Occupancy Measure

Definition A.1 (Conditional Occupancy Measure)**.** *The conditional occupancy measure of a policy $\pi \in \Pi$ is $\rho_\pi : \mathcal{S} \times \mathcal{A} \times \mathcal{H} \mapsto \mathbb{R}$ as $\forall s \in \mathcal{S}, \forall a \in \mathcal{A}, t \in [T]$, we have*

$$\rho_\pi(s_t, a_t|h_{t-1}) = \gamma^t \pi(a_t|s_t, h_{t-1}) P(s_t|h_{t-1}),$$

The conditional occupancy measure can be interpreted as the distribution of state-action pairs at step t when following h_{t-1} under the policy π.

Below, we describe more details about the dataset preparation and experiment settings to help reproducibility. The data and code for trajGAIL and baselines are available at [13].

B. Proof to Theorem III.1

The proof to theorem III.1 follows the procedure of firstly showing the relationship and mutual equivalence of a policy π and its conditional occupancy measure ρ_π, and using this result to show that reward r and conditional occupancy measure ρ_π form a saddle point in the optimization problem of eq. (4), and finally, with a specific choice of reward r regularizer ψ, we can show the result in Theorem III.1.

Lemma A.1. *(Theorem 2 of [54]) If $\rho \in \mathcal{U}$, ρ is the occupancy measure for $\pi_\rho(a_t|s_t, h_{t-1}) \triangleq \frac{\rho(a_t, s_t|h_{t-1})}{\sum_{a_t\prime} \rho(a_t\prime, s_t|h_{t-1})}$, and π_ρ is the only policy whose conditional occupancy measure is ρ.*

Lemma A.2. *Let*

$$\bar{H}(\rho) = -\sum_{t=1}^{T} \sum_{h_t} \rho(s_t, a_t|h_{t-1}) \log(\frac{\rho(s_t, a_t|h_{t-1})}{\sum_{a_t\prime} \rho(s_t, a_t\prime|h_{t-1})}).$$

Then, $\bar{H}$ is strictly concave, and for all $\pi \in \Pi$ and $\rho \in \mathcal{U}$, we have $H(\pi) = \bar{H}(\rho_\pi)$ and $\bar{H}(\rho) = H(\pi_\rho)$.

Proof (Sketch). The concavity can be proved following the concave function definition, that is, for any $\alpha \in [0, 1]$, we have ρ and ρ' as two conditional occupancy measures, such that $\bar{H}(\alpha\rho+(1-\alpha)\rho) \geq \alpha\bar{H}(\rho)+(1-\alpha)\bar{H}(\rho')$. The following two equivalences are shown via expanding $H(\pi)$ and $\bar{H}(\rho)$ with policy π and ρ respectively and replace them with the results of Lemma A.1 in ρ_π and π_ρ respectively. □

The two lemmas together allow us to freely switch between π and ρ_π when considering functions involving causal entropy and expected rewards, as in the following lemma:

Lemma A.3. *If $L(\pi, r) = -H(\pi) - \mathbb{E}_\pi[r(s_t, a_t|h_{t-1})]$ and $\bar{L} = -\bar{H}(\rho) - \sum_{t=1}^{T} \sum_{h_{t-1}} \rho(s_t, a_t|h_{t-1}) r(s_t, a_t|h_{t-1})$, then, for all reward functions r, $L(\pi, r) = \bar{L}(\rho_\pi, r)$ for all policies $\pi \in \Pi$, and $\bar{L}(\rho, r) = L(\pi_\rho, r)$ for all conditional occupancy measures $\rho \in \mathcal{U}$.*

Proof. This lemma is a result of lemma A.1 and A.2. □

Proof to Theorem III.1. Let $\tilde{r}$, $\tilde{\pi}$ be the solution to eq. (4), and π_A be the solution to eq. (6), we have

$$\begin{aligned} \pi_A &\in \arg\min_{\pi\in\Pi} -H(\pi) + \psi^*(\rho_{\pi_E} - \rho_\pi) \\ &= \arg\min_{\pi\in\Pi} \max_{r} -H(\pi) - \psi(r) \\ &+ \sum_{t=1}^{T} \sum_{h_{t-1}} r(s_t, a_t|h_{t-1}) \big(\rho_E(s_t, a_t|h_{t-1}) - \rho(s_t, a_t|h_{t-1})\big), \end{aligned}$$

where ρ_E is the conditional occupancy measure of expert policy π_E.

We wish to show that $\pi_A = \tilde{\pi}$. To do this, let ρ_A be the conditional occupancy measure of π_A, let $\tilde{\rho}$ be the conditional occupancy measure of π, and define $\bar{L} : \mathcal{U} \times \mathbb{R}^{\mathcal{S}\times\mathcal{A}} \mapsto \mathbb{R}$ by

$$\bar{L}(\rho, r) = -\bar{H}(\rho) - \psi(r) + \sum_{t=1}^{T} \sum_{h_{t-1}} r(s_t, a_t|h_{t-1}) \big(\rho_E(s_t, a_t|h_{t-1}) - \rho(s_t, a_t|h_{t-1})\big).$$

The following relationships then hold, due to Lemma A.3:

$$\begin{aligned} \rho_A &\in \arg\min_{\rho\in\mathcal{U}} \max_{r} \bar{L}(\rho, r), \\ \tilde{r} &\in \arg\max_{r} \min_{\rho\in\mathcal{U}} \bar{L}(\rho, r), \\ \tilde{\rho} &\in \arg\min_{\rho\in\mathcal{U}} \bar{L}(\rho, \tilde{r}). \end{aligned}$$

Now $\mathcal{U}$ is compact and convex and $\mathbb{R}^{\mathcal{S}\times\mathcal{A}\times\mathcal{H}}$ is convex due to lemma A.1; furthermore, due to convexity of $-\bar{H}$ (lemma A.2) and ψ, we also have that $\bar{L}(\cdot, r)$ is convex for all r, and that $\bar{L}(\rho, \cdot)$ is concave for all ρ. Therefore, we can use minimax duality:

$$\min_{\rho\in\mathcal{U}} \max_{r} \bar{L}(\rho, r) = \max_{r} \min_{\rho\in\mathcal{U}} \bar{L}(\rho, r).$$

Hence, $(\rho, \tilde{r})$ is a saddle point of $\bar{L}$, which implies that

$$\rho_A \in \arg\min_{\rho\in\mathcal{U}} \bar{L}(\rho, \tilde{r}).$$

Since $\bar{L}(\cdot, r)$ is strictly convex for all r, the above equations imply $\rho_A = \tilde{\rho}$. Since policies corresponding to occupancy measures are unique we get $\pi_A = \tilde{\pi}$.

Then, we choose ψ as

$$\psi(r) = \begin{cases} \mathbb{E}_{\pi_E}[g(r(s, a|h))] & \text{if } r(s, a|h) > 0, \\ +\infty & o.w. \end{cases}$$

where

$$g_\phi(x) = \begin{cases} x - \log(1 - e^x) & \text{if } x > 0, \\ +\infty & o.w. \end{cases}$$

With the above choice, we derive the conjugate of ψ to obtain

$$\begin{aligned} \psi^*(\rho_E - \rho) = \max_{r} \sum_{t=1}^{T} \sum_{h_t} \rho(s_t, a_t|h_{t-1}) \log(\frac{1}{1 + e^{-r(s_t, a_t|h_{t-1})}}) \\ + \rho_{\pi_E}(s_t, a_t|h_{t-1}) \log(1 - \frac{1}{1 + e^{-r(s_t, a_t|h_{t-1})}}). \end{aligned} \tag{11}$$

Since function $y = \frac{1}{1+e^{-x}}, x \in (-\infty, +\infty)$ is monotonically increasing, we denote $D(s, a|h) = \frac{1}{1+e^{-r(s,a|h)}}$ as the reward signal. Therefore, we have $\psi^*(\rho_E - \rho)$ to be

$$\max_{D\in(0,1)} \mathbb{E}_{\pi_E}[\log(D(s_t, a_t|h_{t-1}))] + \mathbb{E}_\pi[\log(1 - D(s_t, a_t|h_{t-1}))].$$

For simplicity and less confusion, in the main text we still use r instead of D. Combining eq. (6) and eq. (11) completes the proof. □