Learning Decision Making Strategies of Non-experts: A NEXT-GAIL Model for Taxi Drivers

Menghai Pan*
mpan@wpi.edu
Worcester Polytechnic Institute

Xin Zhang*
xzhang17@wpi.edu
Worcester Polytechnic Institute

Yanhua Li
yli15@wpi.edu
Worcester Polytechnic Institute

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

Jun Luo
jluo1@lenovo.com
Lenovo Group Limited

ABSTRACT

Thanks to the rapid development of mobile sensing techniques, massive human-generated spatial-temporal data (HSTD) are generated from the urban areas, e.g., passenger-seeking trajectories from taxi drivers, and public transit trips from urban dwellers. These HSTD record sequential decisions made by human agents. Studying human behavior from HSTD provides benefits to many aspects, for example, studying passenger-seeking strategies from experienced taxi drivers can help improve the operation efficiencies of those new drivers. One common method to analyze human behavior from HSTD is Imitation Learning (IL). Existing IL approaches rely on data collected from experts. However, human agents who generate HSTD may have diverse expertise levels across geographical regions, i.e., with good policies in some regions and poor policies in less experienced regions. The problem of how to infer the optimal policy for agents in their unfamiliar or less-experienced regions remains open. In this paper, we propose the novel Generative Adversarial Imitation Learning for Non-experts (NEXT-GAIL) framework to first disentangle expert knowledge, which is irrelevant to spatial-temporal regions, from the demonstration data. Then, such knowledge can be transferred to regions, where the agent does not possess an expert policy. We take the real-world taxi trajectory data as an example to evaluate the performance of our proposed framework. The comparison results illustrate that our proposed NEXT-GAIL outperforms existing state-of-the-art approaches regarding the accuracy of the inferred optimal policy for non-experts.

CCS CONCEPTS

• **Computing methodologies** → **Inverse reinforcement learning**.

KEYWORDS

spatial-temporal data mining, human decision analysis, imitation learning

*Both authors contributed equally to this research.

SIGSPATIAL '21, November 2–5, 2021, Beijing, China

ACM ISBN 978-1-4503-8664-7/21/11…$15.00
https://doi.org/10.1145/3474717.3483924

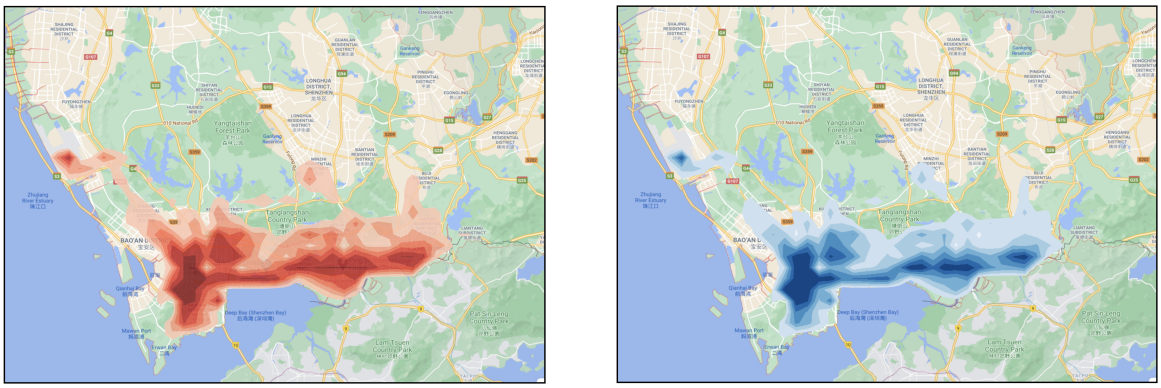
ACM Reference Format:
Menghai Pan, Xin Zhang, Yanhua Li, Xun Zhou, and Jun Luo. 2021. Learning Decision Making Strategies of Non-experts: A NEXT-GAIL Model for Taxi Drivers. In *29th International Conference on Advances in Geographic Information Systems (SIGSPATIAL '21), November 2–5, 2021, Beijing, China.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3474717.3483924

1 INTRODUCTION

Massive spatial-temporal data are being generated by human in the urban environment everyday, for example, the vehicle GPS trajectory data in ride-sharing services (e.g. Lyft [27], Bluegogo [4] and Spin [37]) and traditional taxi services, and the mobility data of urban commuters from automatic fare collection systems, etc. Given these enormous amount of *human generated spatial-temporal data (HSTD)*, how to extract applicable information from them, and utilize them to benefit the urban dwellers is of great practical importance. Among all applications, one aspect of using HSTD is studying the decision-making behavior of human agents, which can benefit people in many respects. For instance, studying the behavior of expert taxi drivers can potentially improve the operation efficiency of new drivers, understanding the decision-making behavior of urban commuters can assist the road network and public transit infrastructure design and resource allocation for the urban planners, etc.

To study the decision-making behaviors of human agents, people usually model human decision-making processes as Markov Decision Processes (MDPs) [24, 26, 31, 33, 35, 43, 46, 48, 51], where human decision-making behaviors are captured by sequences of human decisions. Each human agent aims to maximize the accumulated "rewards" when making decisions, where in practice, the "rewards" are usually unknown and intractable [13, 52, 53].

State-of-the-art approaches. To study how human make decisions when "reward" is unknown, Imitation Learning (IL) servers as a promising technique to recover human decision-making strategies from the observed data, e.g., HSTD. For example, Pan et al. employed Explainable Generative Adversarial Imitation Learning (xGAIL) to recover the optimal policy of the expert taxi drivers from the observed data, and interpret the model to understand how and why taxi drivers make decisions [32]. Zhang et al. extended Generative Adversarial Imitation Learning (GAIL) [15] to conditional GAIL (cGAIL) and unveiled multiple taxi drivers' policies by transferring knowledge across taxi drivers and locations [48]. Li et al. proposed InfoGAIL to imitate experts' behaviors while identifying salient latent factors of variation in the demonstrations [25]. In [31, 33], the authors apply Relative Entropy Inverse Reinforcement Learning



(a) Visitation frequency heat map.

(b) Earning efficiency heat map.

Figure 1: Visitation frequency & earning efficiency heat maps of a taxi driver in Shenzhen, China. Darker color means a higher value.

[5] to recover the linear reward function of the expert taxi drivers and analyze their decision preference dynamics over time. *All of these IL approaches builds upon the assumption that demonstrations to learn from are obtained from expert agents.*

Motivation. However, in real-world applications, it is hard to guarantee that the collected data are all from experts. Taking the taxi drivers as an example as is shown in Figure 1, a driver may be more familiar with some particular regions of the city (i.e., regions with darker red color in Figure 1a) and therefore has a better performance (e.g., higher earning efficiency) in these regions (e.g., due to frequent visits) as is shown in Figure 1b. For these regions, the driver can be viewed as an expert. However, he/she may not have enough experiences in other regions of the city, where the driver is considered as a non-expert. Similar observations are made in other taxi drivers. As a result, none of existing IL works can provide the optimal policy from the non-expert data neither from HSTD. Therefore, we are motivated to extract expert knowledge from non-expert demonstrations, design an IL algorithm that makes use of them and infer human agents' decision strategies from them.

Our NEXT-GAIL. In this paper, we make the first attempt to use non-expert demonstrations, i.e., HSTD, to infer expert decision making strategies by proposing NEXT-GAIL, a novel Generative Adversarial Imitation Learning for Non-experts model. First, NEXT-GAIL "cleans up" the data from non-experts, to extract the portion of data that presents the agent's expert knowledge. Then, NEXT-GAIL learns the expert knowledge representation (irrelevant to the spatial-temporal context) by conducting feature disentanglement. Using the extracted expert knowledge, NEXT-GAIL can infer the optimal policy for spatial-temporal regions, where the agent does not possess an optimal policy. Overall, the proposed NEXT-GAIL provides a complete solution to infer the optimal policy for non-experts, which mainly consists of three components: 1) Expertise Recognition, 2) Expert Knowledge Disentangled Imitation Learning, and 3) Optimal Policy Inference for Non-experts. We use the real-world taxi drivers' passenger-seeking data as an example to demonstrate the performance of our proposed NEXT-GAIL. Our main contributions are summarized as follows:

- We propose an Expert Knowledge Disentangled Generative Adversarial Imitation Learning framework to disentangle the expert knowledge from expert data. The expert knowledge disentangled is irrelevant to the spatial-temporal information of the familiar states, so that it can be utilized and transferred to infer the optimal policy in the non-expert or unfamiliar regions.
- We propose an inference mechanism to infer the optimal policy for the non-experts utilizing the expert knowledge learned from the expert data. The inferred policy can guide the human agent to improve his/her strategies directly.
- We employ real-world taxi trajectory data to evaluate the performance of our proposed NEXT-GAIL. The comparison results illustrate that NEXT-GAIL outperforms the state-of-the-art baseline approaches in inferring the optimal policy for non-experts. *We make our code and unique data set available to contribute to the research community in a Dropbox link* [1].

The remainder of the paper is organized as follows. In Section *Overview*, we introduce the preliminaries and formally define our problem and outline our solution framework. Section *Phase 1: Data Preprocessing* presents our approach for data preprocessing. We elaborate expert knowledge disentangled GAIL in Section *Phase 2: Expert Knowledge Disentangled Imitation Learning*. Section *Phase 3: Inferring Optimal Policy for Non-experts* introduces the inference framework to learn the optimal policy for the non-experts, and Section *Evaluation* evaluates our framework with real-world data. We differentiate our solution with other approaches and introduce relevant works in Section *Related Work* and concludes the paper in Section *Conclusion*.

2 OVERVIEW

In this section, we introduce the preliminaries, formally define the non-expert strategy learning problem, and highlight the research challenges. For brevity, we present a table of notations in Table 1.

2.1 Preliminaries

Markov Decision Processes (MDPs). Markov decision processes (MDPs) [38] provides a mathematical framework to model decision-making processes, where a decision maker (a.k.a. an agent) interacts with an environment in a sequential process. An MDP is

[1]The code is available at https://www.dropbox.com/sh/xjwankbnoz5qo81/AAACjXK8g62rtCK4K-vC7IBUa?dl=0

Table 1: Notations.

Notations	Descriptions
$\mathcal{S} = \{s\}$	State space.
$\mathcal{A} = \{a\}$	Action space.
$\mathcal{T}_E = \{\tau_E\}$	Expert trajectory set.
$\mathcal{T}_{NE} = \{\tau_{NE}\}$	Non-expert trajectory set.
$\mathcal{T} = \{\tau_E, \tau_{NE}\}$	Expert and non-expert mixed trajectory set.
$\pi(a\|s)$	Policy function.
$R(s,a)$	Reward function.
$\pi_E(a\|s)$	Empirical policy from expert trajectory data.
$P(s_{t+1}\|s_t, a_t)$	Transition probability.
γ	The discount factor.
η	Initial state distribution.
$\mathbf{f}_s = [f_1, f_2, f_3, f_4]$	State feature tensor.
g, t	Grid & time identifiers.
v_{gt}	Grid & time related hidden code.
v_k	Grid & time irrelevant hidden code.

represented as a 5-tuple $\langle \mathcal{S}, \mathcal{A}, T, R, \gamma \rangle$, where $\mathcal{S}$ defines the state space, $\mathcal{A}$ the action space, $T : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \mapsto [0, 1]$ characterizes the probability $P(s_{t+1}|s_t, a_t)$ of transiting to state s_{t+1} from s_t after taking action a_t, $R : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$ is the reward function of each state-action pair, and $\gamma \in (0, 1]$ is the discount factor. At a state $s \in \mathcal{S}$, an agent makes a decision and takes an action $a \in \mathcal{A}$ following his/her strategy, i.e., a memoryless stochastic policy π. The memoryless stochastic policy π defines a mapping from the state space $\mathcal{S}$ to the probability over action set $\mathcal{A}$ as $\pi : \mathcal{S} \times \mathcal{A} \mapsto [0, 1]$. It specifies a probability distribution on the action to be executed at each state. A decision-making process forms a **trajectory** $\tau = ((s_0, a_0), \cdots, (s_L, a_L))$, where L is the terminal time step, and the set of all trajectories is denoted as $\mathcal{T} = \{\tau\}$. We denote τ_E as expert trajectory and $\mathcal{T}_E$ as expert demonstrations, and τ_{NE} and $\mathcal{T}_{NE}$ for non-experts'. We denote the expectation with respect to a policy π to represent an expectation with respect to the trajectories it generates, i.e., $\mathbb{E}_\pi[h(s,a)] = \mathbb{E}[\sum_{t=0}^{L} \gamma^t h(s_t, a_t)]$, where $s_0 \sim \eta$, $a_t \sim \pi(\cdot|s_t)$, $s_t \sim P(s_{t+1}|s_t, a_t)$ and h is a function of interest. Each agent aims to maximize its expected reward $\mathbb{E}_\pi[r(s,a|h)]$.

Imitation Learning (IL). Given a large amount of trajectory data from an expert human agent (e.g., an experienced taxi driver), $\mathcal{T}_E$, IL aims at learning a policy function $\pi(a|s)$ to finish a task. There are mainly two paradigms of IL, namely, the inverse reinforcement learning (IRL) [5, 29, 52, 53], and apprenticeship learning (AP) [1, 10, 15].

IRL aims at inversely learning a reward function $R(s,a)$ and recovering expert policy $\pi(a|s)$ from $R(s,a)$ based on various principles, e.g., maximum entropy principle [53], maximum causal entropy principle [52], and relative entropy principle [5]. They all assume a reward function as a linear combination of the features associated with state-action pairs. For example, maximum causal entropy IRL tries to solve the following constrained optimization problem to uncover expert strategy, namely, looking for a policy π with maximal causal entropy (Eq. (1)), and searching for the reward function R such that the expected reward of a trajectory generated under π matches that under the empirical policy π_E from observed data (i.e., enforcing Eq. (2)) when $\pi(a|s)$ expresses a probability distribution in Eq. (3),

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

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

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

Here $H(\pi) = \mathbb{E}_\pi[\sum_{t=0}^{L} \gamma^t (-\log \pi(a_t|s_t))]$ is the γ-discounted causal entropy of π measuring the uncertainty present in a causally conditioned policy distribution $\pi(a|s)$, π represents a learner policy, and π_E (empirical policy) represents the policy observed from the collected expert data.

AP [1, 10, 15], on the other hand, learns expert policy $\pi(a|s)$ directly from expert demonstrations and extends IRL solutions via modeling expert policy π and reward signal R using neural networks. A state-of-the-art AP approach is Generative adversarial imitation learning (GAIL) [15]. GAIL [15] shows that the above constrained optimization strategy learning problem in Eq. (1-3) is equivalent to solving a min-max problem with the objective of minimizing the Jensen-Shannon (JS) divergence D_{JS} between the trajectory distribution induced by policy π and empirical policy π_E, i.e.,

$$\min_{\pi \in \Pi} -\lambda H(\pi) + D_{JS}(\pi, \pi_E), \text{with}$$

$$D_{JS}(\pi, \pi_E) = \max_R \mathbb{E}_\pi[\log(R(s,a))] + \mathbb{E}_{\pi_E}[\log(1 - R(s,a))],$$

with Π as the policy probability simplex space, guaranteeing constraint Eq. (3), and λ as the Lagrangian multiplayer. As a result, GAIL inversely learns both the policy function $\pi(a|s)$ and a reward signal $R(s,a)$ employed by the expert agent from the his/her trajectories using a Generative Adversarial Net (GAN) [13] structure. It solves the strategy learning problem with a generator network G (representing the policy function π) and a discriminator network D (representing the reward function R). However, *both IRL approaches and GAIL requires the demonstrations (i.e., trajectories) to be generated from expert agents [15, 53], thus non-expert demonstrations fail to be applied under such a framework.* To learn from non-expert demonstrations, we model the human decision making process as MDPs and formulate our problem in the following sections.

2.2 Human Decision Making as Markov Decision Processes

Human-generated spatial-temporal data (HSTD) record human mobility trajectories which embed sequential human decisions. For example, taxi drivers' sequential decisions when seeking for a passenger are recorded in taxi GPS traces; urban commuters' sequential decisions when deciding on transit modes are inferred from automated fare collection devices on public transportation. Therefore, HSTD contains human sequential decision making trajectories as sequences of human agent traversed spatial-temporal states following his/her decision strategy. HSTD is mixed with both expert trajectories and non-expert trajectories. In this sense, we formulate human decision making from HSTD as MDPs, and formally define their state and action spaces as below:

- *State* $s \in \mathcal{S}$: A state s from HSTD can be uniquely defined by the geographical location (e.g. latitude and longitude) and temporal information (e.g. time stamp). It also relates to a set of decision

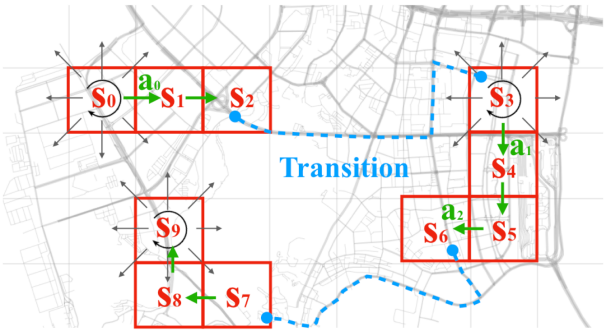


Figure 2: MDP of taxi driver's decision making process

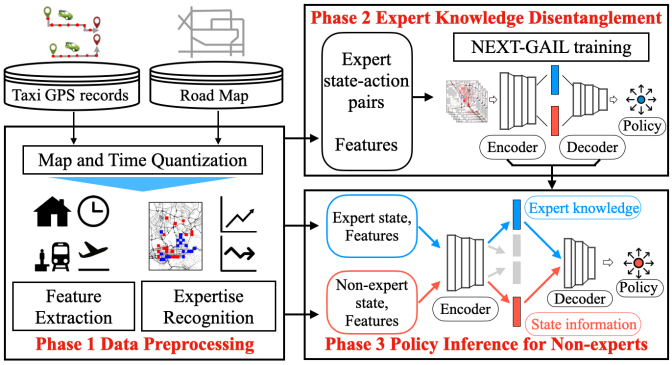


Figure 3: NEXT-GAIL Solution Framework.

making related features, e.g., traffic speed and passenger flow of surrounding regions. These features reflects what a humen agent would consider when making decisions.

- *Action* $a \in \mathcal{A}$: An action a from HSTD represents a decision made by a human agent at a spatial-temporal state s when performing a task. An action a made at state s will affect the next state transited to. For example, a passenger near a bus station A (as a state) takes a bus (as an action) and is transferred to a next bus station B (a next state).

An illustration example. In taxi drivers' passenger-seeking processes shown in Figure 2, a taxi driver (when the taxi is empty) is the agent, he/she makes a sequence of decisions about which directions (as actions a_i's) to go based on his/her own decision-making strategy at different spatial-temporal states s_i's. Below, we will simply use state for spatial-temporal state for brevity.

When at different geographical locations, each human agent has her own decision strategies to follow to choose an action for completing a task. The human decision-making strategy are modeled and characterized by the *policy function* and the *reward function* defined below:

- *Reward* $R(s, a)$: The reward function $R(s, a)$ reflects the "reward" a human agent obtains when taking action a at state s. It quantifies how satisfied a human agent is towards his/her situation and response.
- *Policy* $\pi(a|s)$: A policy function $\pi(a|s)$ of a human agent is a mapping from a state s to probabilities over actions $a \in \mathcal{A}$, i.e., the probability distribution of choosing an action a given a state s. It governs what decisions a human agent to take at different situations.

As a result, a human agent's (e.g., taxi driver's) decision-making strategy can be characterized by i) the *policy function* $\pi(a|s)$ controlling how the agent chooses an action, and ii) the *reward function* $R(s, a)$ governing how the agent evaluates states and actions.

2.3 Non-Expert (NEXT) Strategy Learning Problem, Challenges & Solution Framework

Problem Definition. Given mixed demonstrations from both experts and non-experts, i.e., $\mathcal{T} = \{\tau_E, \tau_{NE}\}$, we aim to distinguish $\mathcal{T}_E$ and $\mathcal{T}_{NE}$ from $\mathcal{T}$, disentangle expert knowledge from $\mathcal{T}_E$, extract state-related information from $\mathcal{T}_{NE}$, and infer the optimal decision-making strategy, namely, the policy $\pi(a|s)$ for the non-experts.

Challenges. The proposed NEXT strategy learning problem is challenging in three aspects: (**C1**) How to quantify the expertise of a human agent at different spatial regions? (**C2**) How to disentangle expert knowledge from mixed demonstrations in HSTD (as observed in Figure 1)? (**C3**) Given disentangled expert knowledge, how to design an IL algorithm such that expert knowledge can be transferred and utilized to learn optimal strategies for non-experts?

Solution Framework. To respond to the above challenges and solve the NEXT strategy learning problem, we propose the novel framework of Generative Adversarial Imitation Learning for Non-experts (in short, NEXT-GAIL). Figure 3 illustrates the solution framework. Note that, in this paper, we use the taxi drivers' passenger-seeking problem as an example and application to evaluate our proposed NEXT-GAIL. In the framework, NEXT-GAIL consumes two sources of data and consists of three phases: (1) Data Preprocessing to tackle **C1**, (2) Expert Knowledge Disentangled Imitation Learning to tackle **C2**, and (3) Inferring optimal policy for non-experts to tackle **C3**.

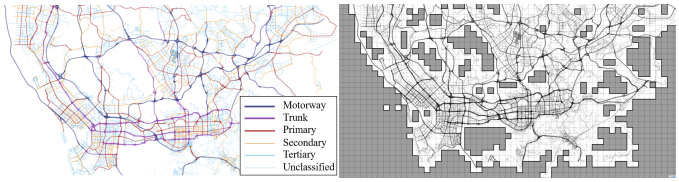
3 PHASE 1: DATA PREPROCESSING

In this section, we tackle **C1** via demonstrating the data preprocessing procedure with the taxi driver passenger-seeking process as an example, and introducing the expertise recognition mechanism.

3.1 Data Description

We use two data sources for the analysis of taxi driver passenger-seeking process and treat them as input, including (1) taxi trajectory data and (2) road map data. Both datasets are collected in Shenzhen, China in 2014 and 2016 for consistency.

Taxi trajectory data were collected from taxis equipped with GPS devices in Shenzhen, China during 2014 and 2016. It contains GPS records from a total of 17, 877 taxis, each of which generates a GPS point every 40 seconds on average. Overall, 51,485,760 GPS records were collected on a daily basis, and each GPS record carries five key attributes, including a unique taxi plate ID, time stamp, passenger indicator, latitude and longitude. The passenger indicator bears a binary value with 1 indicating a passenger on board, and 0 otherwise.



(a) Shenzhen road map **(b) Map gridding**

Figure 4: Shenzhen map data

Shenzhen Road map data were collected from OpenStreetMap [30]. It covers the area defined in between 22.44° to 22.87° in latitude and 113.75° to 114.63° in longitude. It contains information of 21,000 roads categorized in six levels, namely, motorway, trunk road, primary road, secondary road, tertiary road and unclassified road, as demonstrated in Figure 4a.

3.2 Map and Time Quantization

For better characterization of human agents' (i.e., taxi drivers') activities and ease of decision strategy analysis, we define the spatial and temporal spaces a taxi driver traverses by i) dividing and partitioning Shenzhen city into equal side-length *(spatial)* grid cells with a fixed side-length $l = 0.01°$ in latitude and longitude, ii) discretizing a day into 288 five-minute *(temporal)* intervals. There are a total of 1,934 cells (eliminating inaccessible cells in the ocean and unreachable regions) connected among each other by the road network, as shown in Figure 4b. We thus represent each cell as $\ell = (x, y)$, where x and y are longitudinal and latitudinal cell indexes, respectively. A spatial-temporal state s is then uniquely defined by a spatial gird cell ℓ, a time interval t, and the day of the week d, i.e., $s = (x, y, t, d)$.

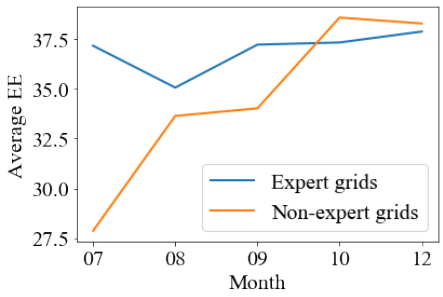
3.3 Feature Extraction

Taxi drivers' decisions (e.g., which direction to go) are affected by various features (such as traffic speed, congestion condition and so on) observed in the surrounding urban environment of the target area. These features are referred to as taxi drivers' state observations. We model a taxi driver's observations at a spatial-temporal state s as the state feature denoted as $\mathbf{f}_s = [f_1, f_2, f_3, f_4]$. It is a tensor including four statistic matrices for the surrounding 5×5 grid cells of s. Specifically, each statistic matrix f_i with $i = 1, \cdots, 4$ is a feature map of surrounding 5×5 grids centering s in one aspect of feature, where f_1 is the number of pickups matrix, f_2 is the traffic volume matrix, f_3 is the traffic speed matrix, and f_4 is the waiting time matrix.

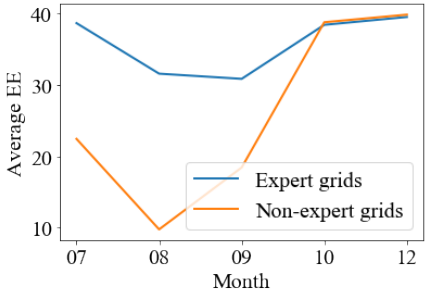
3.4 Expertise Recognition

It is hard to guarantee that the demonstrations collected from human agents are always expert. Taking the taxi drivers as an example, each driver can be expert in the regions where he/she is particularly familiar with, whereas in some other regions, his/her "performance" may not be superior. Here in this paper, we quantify the performance of taxi drivers in each grid by their *Earning Efficiency* (EE), i.e., hourly earnings, inside each grid[2]. Then, we recognize and distinguish the expertise of each grid for each taxi driver to the following categories:

[2]There are other ways of defining human agents' expertise level, we follow [31, 48, 49] and choose to use taxi drivers' earning efficiencies.



(a) Driver 1.



(b) Driver 2.

Figure 5: Earning efficiency trends of expert and non-expert grids.

- **Expert grids:** the earning efficiency of the driver inside these grids remains similar from July to December. This is selection is based on the observation in [31] where expert agents tend to have consistent strategies and remain similar earning efficiencies over time.
- **Non-expert grids:** the earning efficiency of the driver inside these grids increases from July to October, and remains stable from October to December. We consider these grids non-expert from July to September, and they become expert after October. The criterion of defining non-expert grids is that we can use the data in these grids after October as the ground truth to evaluate our proposed NEXT-GAIL.

As a result, demonstrations traversing expert grids are viewed expert demonstrations $\mathcal{T}_E$ where we extract expert knowledge from, and those traversing non-expert grids are viewed non-expert demonstrations $\mathcal{T}_{NE}$ where we elicit state-related information from. Note that we do not consider grids in the data whose earning efficiency fluctuates over time and shows no sign of high expertise. This naturally stems from the fact that these drivers tend to be constantly learning to locate a passenger and have not reached expert level where no ground truth data can be obtained to test against and evaluate from. Figure 5 shows the trends of the average earning efficiencies in the expert grids and non-expert grids of two drivers.

4 PHASE 2: EXPERT KNOWLEDGE DISENTANGLED IMITATION LEARNING

In this section, we tackle **C2** and introduce the design of our proposed NEXT-GAIL model in learning the expert behavior with knowledge disentanglement.

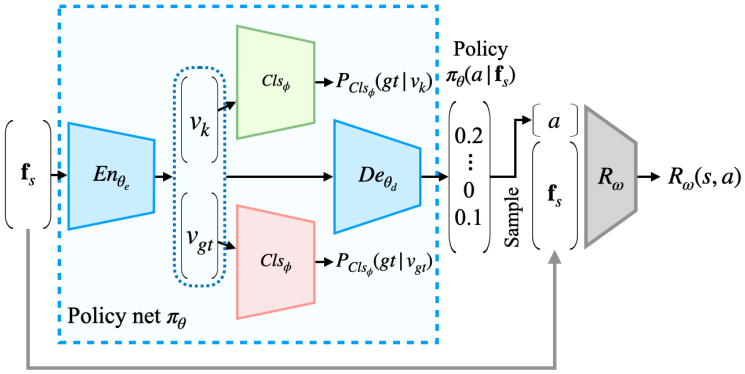


Figure 6: NEXT-GAIL training framework.

Algorithm 1 NEXT-GAIL Training Process

Input: Taxi drivers' decision-making data as state-action pairs $\mathcal{T}_E = \{(\mathbf{f}_s(s), a)\}$ in expert grids, and in all grids $\mathcal{T}$; initialize policy net, reward net, classifier parameters $\theta = \{\theta_e, \theta_d\}$ ω, and ϕ; batch size B.
Output: Learned policy π_θ, reward R_ω and classifier Cls_ϕ.
1: **for** Each Epoch $i = 0, 1, \cdots$ **do**
2: Generate trajectories $\tilde{\mathcal{T}}^i$ from π_{θ^i}.
3: Sample state-action sequences from $\mathcal{T}_E$ and $\tilde{\mathcal{T}}^i$ each with batch size B to evaluate Eq.(4).
4: Update $\theta^i = \{\theta_e^i, \theta_d^i\}$ to minimize ① in Eq.(4).
5: Update ω^i to maximize ① in Eq.(4).
6: Update ϕ^i to minimize ② in Eq.(4).
7: **end for**

NEXT-GAIL algorithm. We employ Generative Adversarial Imitation Learning (GAIL)[15] framework to learn the optimal policy from the expert data, i.e., data collected from the expert regions(/grids) $\mathcal{T}_E$. The generator of GAIL is the policy net π_θ parameterized by θ, which consumes the observations of a state, i.e., state feature maps $\mathbf{f}_s$ and outputs the policy. The discriminator is the reward net R_ω parameterized by ω, which takes both the state feature $\mathbf{f}_s$ of state s, and the sampled action a as input, and outputs the reward signal which indicates to what degree the generated state-action pair matches the demonstrated expert behavior.

Meanwhile, in the policy net π_θ, we aim to disentangle the knowledge of the driver into two parts, i.e., the knowledge related to the specific regions(/grids) and time slots, and the knowledge irrelevant to the location and time, which is considered as the high-level knowledge of the driver and can be transferred to other locations and time slots. To accomplish this goal of knowledge disentanglement, we design the policy net as an auto-encoder-decoder, i.e., $\pi_\theta(s) : De_{\theta_d}(En_{\theta_e}(s))$ with $\theta = \{\theta_e, \theta_d\}$. The framework is illustrated in Figure 6. To enforce expert knowledge and spatial-temporal information disentanglement, we design a spatial-temporal classifier denoted as Cls_ϕ parameterized by ϕ to classify the location and time information from the hidden latent codes v_{gt} (related to grid and time) and v_k (irrelevant to grid and time). The encoder En_{θ_e} is trained to produce v_k as adversarial inputs to the classifier Cls_ϕ containing little spatial-temporal information, and produce v_{gt} as positive samples to promote the performance of the classifier Cls_ϕ to make correct prediction on spatial-temporal information g and t as is shown in Figure 6. Therefore, the objective function of NEXT-GAIL is

$$\min_{\pi_\theta, Cls_\phi} \underbrace{-\lambda H(\pi_\theta(s)) + D_{JS}(\pi_\theta, \pi_E)}_{①} \tag{4}$$
$$+ \underbrace{\mathbb{E}_{\pi_E}\left[\log P_{Cls_\phi}(g, t|v_k, En_{\theta_e}(s)) - \log P_{Cls_\phi}(g, t|v_{gt}, En_{\theta_e}(s))\right]}_{②},$$

where $D_{JS}(\pi_\theta, \pi_E) = \max_{R_\omega} \mathbb{E}_{\pi_\theta}[\log(R_\omega(s,a))] + \mathbb{E}_{\pi_E}[\log(1 - R_\omega(s,a))]$. Here, ① in Eq. (4) is the same as the original GAIL, and ② is designed to train the encoder to produce v_k to fool the classifier, and produce v_{gt} to improve the performance of the classifier.

Now we are in a position to present our NEXT-GAIL training algorithm as is shown in Alg. 1. NEXT-GAIL applies the Adam [18] optimizer for gradient update on θ, ω and ϕ, and utilizes the Trusted Region Policy Optimization (TRPO) [36] for updating θ to decrease eq. (4) with respect to π_θ. In each training epoch i, we use current policy π_{θ^i} generate learner policy trajectories denoted as $\tilde{\mathcal{T}}^i$ (line 2). With $\tilde{\mathcal{T}}^i$ and given demonstration in expert grids $\mathcal{T}_E$, we sample the state-action sequences from them each with a batch size of B, and evaluate the objective in Eq. 4 (line 3). We then update the policy net π_{θ^i} with a TRPO step whose reward is R_{ω^i} (line 4), and update the reward net R_{ω^i} and the classifier Cls_{ϕ^i} sequentially with the Adam optimizer.

Interpretation of Cls_ϕ. In fact, the classifier Cls_ϕ with input v_{gt} can maximize the mutual information between v_{gt} and the intrinsic location and time information of the input state, which encourages the encoder to push the information related to location and time to v_{gt}. Similarly, the classifier Cls_ϕ with input v_k can minimize the mutual information between v_k and the intrinsic location and time information, which makes the encoder dispel the information about location and time out from v_k. Below, we will take the classifier Cls_ϕ with input v_{gt} as an example to show the connection between the classifier objective and the mutual information maximization/minimization.

Mutual information between X and Y, $I(X;Y)$, measures the "amount of information" learned from a random variable Y after observing the other random variable X [6]. Here, we study the mutual information between the intrinsic location and time information g, t and the output vector v_{gt} of the encoder $En(s|gt)$, i.e., $I(gt; En(s|gt))$. Using Variational Information Maximization [3], the mutual information $I(gt; En(s|gt))$ is lower bounded by

$$I(gt; En(s|gt)) \geq \mathbb{E}_{x\sim En(s|gt)}[\mathbb{E}_{g't'\sim P(gt|x)}[\log Q(g't'|x)]] + H(gt)$$

Then, according to Lemma 5.1 in [6], the above lower bound can be rewritten as below:

$$L_I(En, Q) = \mathbb{E}_{x\sim En(s|gt)}[\mathbb{E}_{g't'\sim P(gt|x)}[\log Q(g't'|x)]] + H(gt). \tag{5}$$

Now, instead of maximizing $I(gt; En(s|gt))$ directly, we maximize the lower bound $L_I(En, Q)$. Note that in Eq.(5), maximizing $Q(gt|x)$ leads to the maximum of the lower bound. Here in our proposed

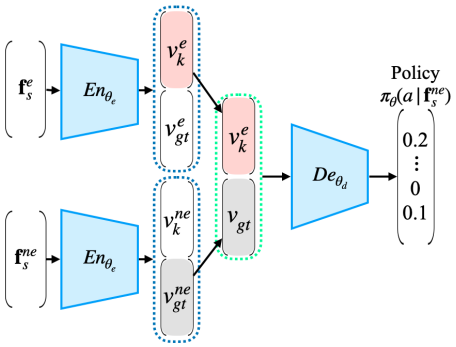


Figure 7: NEXT-GAIL inferring framework.

framework, we employ a classifier to maximize the probability of g, t given x, i.e., v_{gt}.

5 PHASE 3: INFERRING OPTIMAL POLICY FOR NON-EXPERTS

After we obtain the well-trained NEXT-GAIL model from the expert data, we want to transfer the expert knowledge v_k in the expert grid to the target state at a non-expert grid and tackle **C3**.

As shown in Figure 7, we want to infer the optimal policy for state s_{ne} which is from the non-expert region or time slot, while s_e is the expert state from expert grid and time slot. This expert state s_e can be determined by selecting the expert state from the observed data whose feature map $\mathbf{f}_s^e$ has the minimum L_2 distance to the target non-expert state $\mathbf{f}_s^{ne}$. Then, we utilize the encoder obtained in Phase 2 to disentangle the knowledge into v_k and v_{gt} for s_e and s_{ne} as illustrated below

$$\{v_k^e, v_{gt}^e\} = En_{\theta_e}(\mathbf{f}_s^e),\ \{v_k^{ne}, v_{gt}^{ne}\} = En_{\theta_e}(\mathbf{f}_s^{ne}).$$

Then we concatenate the v_k^e of the expert state and the v_{gt}^{ne} of the non-expert state, and feed it to the decoder to obtain the optimal policy of the non-expert state, i.e., $\pi_\theta(s_{ne}) = De_{\theta_d}(v_k^e, v_{gt}^{ne})$. This fulfills non-expert policy inference.

6 EVALUATION

In this section, we evaluate the performance of our proposed NEXT-GAIL using the real-world taxi trajectory data collected in Shenzhen, China from July to October 2016. We compare our proposed framework with state-of-the-art baseline models, and analyze the learning curve of NEXT-GAIL to illustrate the effectiveness of knowledge disentanglement. We make our code and unique data set available to contribute to the research community in the supplementary material of this paper.

6.1 Evaluation Plan

We conduct two sets of experiments utilizing the real-world taxi trajectory data.

- Baseline methods comparison: We compare the accuracy of the inferred policy for the non-experts by our proposed NEXT-GAIL with the state-of-the-art baseline methods GAIL[15] and cGAIL[48].
- Expert knowledge disentanglement analysis: We analyze the learning curve of our proposed NEXT-GAIL, and illustrate the effectiveness of the expert knowledge disentanglement.

6.2 Evaluation Metrics

Ground truth optimal policy for non-experts. We analyze each driver's expertise for each grid via a data-driven approach. Specifically, in the expert grids, an expert driver's earning efficiency remains stable and ranks at the top 15% among all taxi drivers from July to December. The non-expert grids for a driver in July are those in which the driver's earning efficiencies increase from July to October, and remains stable at the top 15% from October to December among all drivers. In experiment, we study grids that are non-expert in July, and turns expert after October for a driver. We use the data from each driver in their July's expert grids to train our proposed NEXT-GAIL and all the baseline models, and the data in the non-expert grids in July to test the performance of optimal policy inference for the non-experts. Data collected in October in the same grids as the July's non-expert grids are used to extract the ground-truth optimal policy for non-experts. The ground-truth optimal policy for non-experts are empirically calculated for each grid, i.e., calculating the percentage of choosing each action in each grid via data-driven approach.

Metrics. In order to measure the accuracy of the inferred policy compared with the empirical ground-truth optimal policy, we employ the Kullback-Leibler (KL) divergence and L_2-distance metrics. KL divergence [22] measures how one distribution P is different from a ground truth distribution Q as

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

L_2-distance [2] is also called the Euclidean distance, 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.,

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

Smaller KL divergence and L_2-distance indicate higher accuracy in inferring the optimal policy for the non-experts.

6.3 Experiment Setups

We use the data in the expert grids in July to train all models and test the inference performance on the non-expert grids. All experiments were run on Red Hat Enterprise Linux 7.2 and written in Python 3.7.3. The implementation of neural networks is based on PyTorch 1.0.1[3]. We also employ Numpy 1.16.4 and Scipy 1.3.0 in the implementation.

NEXT-GAIL implementation. The implementation details of NEXT-GAIL are as follows:

- The encoder network. The encoder net consists of 2 convolutional layers and 1 fully-connected layer. Between the 2 convolutional layers, there is a max pooling layer with a filter size of 2×1. The number of filters in the 2 convolutional layers is 3 and

[3]https://pytorch.org/get-started/previous-versions/

6, respectively. We use a kernel size of 2×2 for the convolutional layers. The output dimension of the fully-connected layer is 24.

- The decoder network. The input of the decoder network is the combination of hidden vectors v_k and v_{gt} of size 24 in total. The decoder network consists of 3 fully-connected layers with output dimensions of 48, 84, and 10, respectively. There is a softmax layer after the 3 fully-connected layers, which outputs the policy for the input state of the encoder. Together the encoder and decoder form the generator, i.e., policy net, in the NEXT-GAIL framework.
- The classifier network. The Spatial-temporal classifier (in short, ST-classifier) and the adversarial ST-classifier share the same network, i.e., the structure and parameters are shared. There are 2 fully-connected layers in the classifier net. The output dimension of the first fully-connected layer is 48, and that of the second fully-connected layer equals to the number of valid spatial-temporal regions. A softmax layer is added in the end to output the possibility of classifying to the spatial-temporal regions.
- The discriminator network. In the discriminator network, the input is an input state of size $4 \times 5 \times 5$ and an action, before the covolutional layers, we use a fully-connected layer to map the input to the dimension of $1 \times 9 \times 9$, followed by 2 convolutional layers and 3 fully-connected layers. The filter size of the 2 convolutional layers is 2×2, and the numbers of filters are 2 and 6, respectively. The output dimensions of the 3 fully-connected layers are 120, 84, 1.

During the training process, we apply batch gradient descent approach to update the generator network and discriminator network, with a predefined $1,000$ epochs. We employ ADAM[18] with a learning rate of $2e^{-3}$ to update the parameters in the encoder, decoder, classifier, and the discriminator networks.

Baselines. We compare our proposed NEXT-GAIL with two state-of-the-art imitation learning approaches, i.e., Generative Adversarial Imitation learning (GAIL) [15] and Conditional Generative Adversarial Imitation Learning (cGAIL) [48]. The detailed settings are as following:

- GAIL [15]. The generator is a covolutional neural network consuming the state feature map $\mathbf{f}_s$ and producing the policy, and the discriminator network (i.e., reward) takes both the state feature map $\mathbf{f}_s$ of state s, and the sampled action a as input, and outputs the reward signal which indicates to what degree the generated state-action pair matches the demonstrated trajectories. GAIL's input state is the same as in NEXT-GAIL. GAIL's generator has the same structure as the NEXT-GAIL, i.e., a combination of encoder and decoder nets. GAIL's discriminator also has the same structure as that in NEXT-GAIL. The difference to NEXT-GAIL is that GAIL does not have the classifier net for feature/knowledge disentanglement.
- cGAIL [48]. Differing from GAIL, the generator of cGAIL consumes the state feature map together with a grid label which is served as a condition, and outputs the policy. The generator takes the state-action pair and the grid label condition as input, and outputs the signal. Here, the condition of cGAIL is the grid label of the input state, which is embedded into a channel of size 5×5, and then concatenated with the $4 \times 5 \times 5$ state feature map. Other than the input channels of the first convolutional layers (cGAIL has 1 more channel than GAIL), the generator and discriminator of cGAIL are the same as those in GAIL.

6.4 Comparison Results

Figure 8 shows the KL divergence and the L_2-distance between the inferred policy and the empirical ground truth policy for the non-expert grids in July. We randomly select 10 drivers from our data set. The x-axis is the driver ID, and y-axis is the average KL divergence (Figure 8a) and L_2-distance (Figure 8b) over the non-expert grids for each taxi driver. Figure 8a illustrates that our proposed NEXT-GAIL outperforms cGAIL and GAIL in all cases. On average, NEXT-GAIL can reach a 25% and 12% lower KL divergence comparing with GAIL and cGAIL respectively. When looking into the results of L_2-distance, the advantage of our model is more distinct with an average 33% and 23% lower L_2-distance comparing with GAIL and cGAIL respectively. These results can illustrate that our proposed NEXT-GAIL framework can outperform the state-of-the-art imitation learning approaches in inferring the optimal policy for the non-experts.

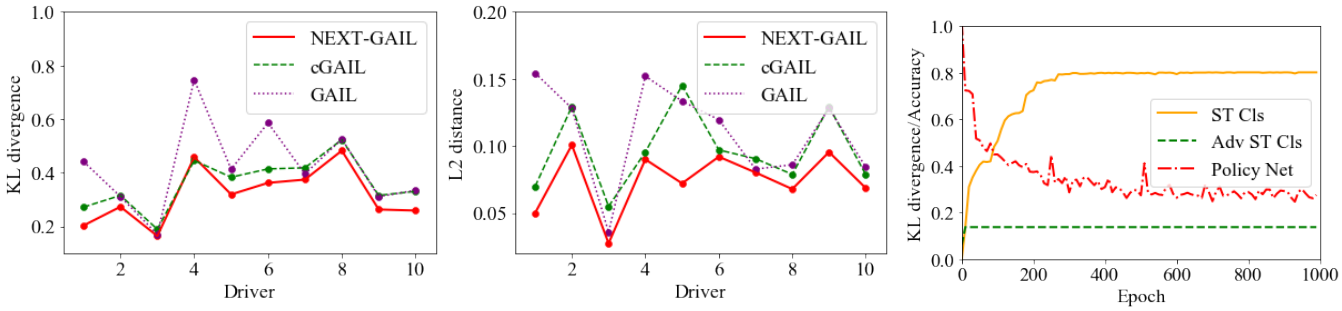
6.5 Expert Knowledge Disentanglement Analysis

To study the effectiveness of the expert knowledge disentanglement of our proposed model, we analyze the accuracy of the spatial-temporal classifier Cls_ϕ consuming v_{gt} with legend "ST Cls" consuming v_k with ledeng "Adv ST Cls" along the training process in Figure 9. The learning curve of the training phase for a driver is illustrated in Figure 9, where the x-axis is the training epoch, and the y-axis is the accuracy of the ST Cls (orange solid curve) and the Adv ST Cls (green dashed curve) and the KL-divergence of the policy net (red dashdotted curve). The red dashdotted curve indicates that the KL-divergence of the policy net decreases as the training process goes on, which means our proposed NEXT-GAIL can imitate the expert behavior in the training phase. The accuracy of the ST classifier increases as the number of training epochs increases, and reaches an accuracy of 80.20% when converges. Note that, we also train a neural network aiming to classify the grid from the input state feature map $\mathbf{f}_s$ directly using the same training data, which obtains an accuracy of 89.09%, slightly higher than that of the ST classifier in NEXT-GAIL. These results can indicate that v_{gt} can absorb most spatial information from the input state feature map $\mathbf{f}_s$. Meanwhile, the accuracy of adversarial ST classifier remains low along the whole training process. The highest accuracy of Adv ST Cls is 13.79% which is similar to performance of random guess, which has an accuracy of 4.35% given totally 23 expert grids. The performance of Adv ST Cls indicates v_k contains nearly no information about the spatial characteristics of the input state.

7 RELATED WORK

7.1 Human Decision Analysis.

Human decision analysis targets on improving decision making efficiency via learning representations of decision-maker's strategies and preferences and analyzing them from uncertain, complex and dynamical decision features [14, 16]. It has been proved to be



(a) KL divergence **(b) L2 distance**

Figure 8: Comparison results.

Figure 9: Learning curve of NEXT-GAIL.

useful in the field of public health [42], business [9], and urban computing [50] with human generated spatial-temporal data (HSTD). Given HSTD, human decision analysis is applied to improve taxi operation efficiency [11, 12, 28, 33, 35, 44, 45, 48, 49], and analyze urban dweller transportation modes [41, 43]. Specifically in the taxi scenario, [28, 45] focuses on taxi dispatching for better taxi operation management, and [11, 12, 33, 35, 44, 48] targets on passenger seeking for each individual taxi driver's well-being. However, all of these works focus on finding the optimal decision strategies with given HSTD, overlooking the HSTD's quality for application. By contrast, our work studies the quality of HSTD when applied on imitation learning, and makes appropriate adjustments on algorithm for more reasonable data usage which entails a better performance.

7.2 Imitation learning.

Imitation learning (IL), also known as learning from demonstrations, has two main paradigms, namely, inverse reinforcement learning (IRL), and apprenticeship learning (AL). IL inversely recovers the agent's policy and reward functions from the collected demonstrations. IRL approaches [5, 52, 53] have been proposed based on different principles, including maximum entropy, maximum causal entropy, and relative entropy principles [5, 52, 53]. All the approaches assume that the underlying reward function is a linear function and features have to be manually extracted. A progress in AP, the Generative adversarial imitation learning (GAIL) [15], and its extension works cGAIL [48], xGAIL [32], InfoGAIL [25], adversarial IRL [10], fGAIL [47], trajGAIL [49] learn the non-linear policy and reward functions as two deep neural networks (DNNs), with theoretical connections to generator and discriminator in generative adversarial networks (GANs) structure. All of these existing imitation learning approaches assume the observed data are from expert agents. However, it is hard to ensure that the data we collected from real-world are all expert. And these works cannot infer the optimal policy of the non-experts. In this paper, we make the first attempt to deal with this challenge with HSTD.

7.3 Feature disentanglement.

Feature disentanglement aims at learning interpretable representations with deep generative models such as generative adversarial networks (GANs) [13] and variational autoencoders (VAEs) [20]. It has been studied under different degrees of supervision. [21] applied feature disentanglement for 3D image rendering via learning invertible graphic codes with fully supervised data. [19] extended VAE for representation disentanglement in semi-supervised settings. For unsupervised situation, [6] maximized the mutual information between latent codes and synthesized data for the fulfillment of feature disentanglement. Feature disentanglement has also been applied in various domains such as in pose-invariant recognition [34, 39, 40], identity-preserving image editing [17, 23], voice conversion [8] and automatic speech recognition [7]. Though broadly applied on images, voices and speeches, few works focus on applying feature disentanglement on human-generated spatial-temporal data. In this work, we disentangle expert knowledge from demonstrated *human-generated spatial-temporal* data and enable knowledge transfer via knowledge ensemble and imitation learning. To the best of our knowledge, we are the first to apply feature disentanglement in imitation learning for the fulfillment of knowledge transfer with demonstrations from both expert and non-expert agents.

8 CONCLUSION

In this paper, we propose NEXT-GAIL, a novel Generative Adversarial Imitation Learning for Non-experts model that infers the expert policy for non-experts, by disentangling expert knowledge from their demonstrations, and transferring the knowledge across spatial-temporal regions. Our evaluation results on a real-world large scale dataset, specifically, on taxi drivers' passenger-seeking processes, illustrate that NEXT-GAIL outperforms the state-of-the-art baseline approaches by an average margin of 23% in accuracy when inferring the optimal policy for non-experts.

ACKNOWLEDGMENTS

Menghai Pan, Xin Zhang and Yanhua Li were supported in part by NSF grants IIS-1942680 (CAREER), CNS-1952085, CMMI-1831140, and DGE-2021871. 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. Jun Luo was partially supported by ARC Discovery Project (grant DB210100743).

REFERENCES

[1] Pieter Abbeel and Andrew Y Ng. 2004. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first ICML*. ACM, 1.

[2] Howard Anton and Chris Rorres. 2010. *Elementary linear algebra: applications version*. John Wiley & Sons.

[3] David Barber and Felix V Agakov. 2003. The im algorithm: a variational approach to information maximization. In *Advances in neural information processing systems*. None.

[4] Bluegogo. [n.d.]. Bluegogo bike-sharing services. https://www.bluegogo.com/us/.

[5] Abdeslam Boularias, Jens Kober, and Jan Peters. 2011. Relative entropy inverse reinforcement learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. 182–189.

[6] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. 2016. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems*. 2172–2180.

[7] Yi-Chen Chen, Chia-Hao Shen, Sung-Feng Huang, and Hung-yi Lee. 2018. Towards unsupervised automatic speech recognition trained by unaligned speech and text only. *arXiv preprint arXiv:1803.10952* (2018).

[8] Ju-chieh Chou, Cheng-chieh Yeh, Hung-yi Lee, and Lin-shan Lee. 2018. Multi-target voice conversion without parallel data by adversarially learning disentangled audio representations. *arXiv preprint arXiv:1804.02812* (2018).

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

[10] Justin Fu, Katie Luo, and Sergey Levine. 2017. Learning robust rewards with adversarial inverse reinforcement learning. *arXiv preprint arXiv:1710.11248* (2017).

[11] Yong Ge, Chuanren Liu, Hui Xiong, and Jian Chen. 2011. A taxi business intelligence system. In *KDD*. ACM, 735–738.

[12] Yong Ge, Hui Xiong, Alexander Tuzhilin, Keli Xiao, Marco Gruteser, and Michael Pazzani. 2010. An energy-efficient mobile recommender system. In *KDD*. ACM, 899–908.

[13] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*. 2672–2680.

[14] Paul Goodwin and George Wright. 2014. *Decision Analysis for Management Judgment 5th ed.*

[15] Jonathan Ho and Stefano Ermon. 2016. Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems*. 4565–4573.

[16] Ronald A Howard. 1980. An assessment of decision analysis. *Operations Research* 28, 1 (1980), 4–27.

[17] Rui Huang, Shu Zhang, Tianyu Li, and Ran He. 2017. Beyond face rotation: Global and local perception gan for photorealistic and identity preserving frontal view synthesis. In *Proceedings of the IEEE International Conference on Computer Vision*. 2439–2448.

[18] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[19] Durk P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. 2014. Semi-supervised learning with deep generative models. In *Advances in neural information processing systems*. 3581–3589.

[20] Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).

[21] Tejas D Kulkarni, William F Whitney, Pushmeet Kohli, and Josh Tenenbaum. 2015. Deep convolutional inverse graphics network. In *Advances in neural information processing systems*. 2539–2547.

[22] Solomon Kullback and Richard A Leibler. 1951. On information and sufficiency. *The annals of mathematical statistics* 22, 1 (1951), 79–86.

[23] Guillaume Lample, Neil Zeghidour, Nicolas Usunier, Antoine Bordes, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2017. Fader networks: Manipulating images by sliding attributes. In *Advances in neural information processing systems*. 5967–5976.

[24] P Li, Sandjai Bhulai, and JT van Essen. 2017. Optimization of the revenue of the New York city taxi service using Markov Decision Processes. In *6th International Conference on Data Analytics, Barcelona (Spain), November 12-16*. IARIA, 47–52.

[25] Yunzhu Li, Jiaming Song, and Stefano Ermon. 2017. Infogail: Interpretable imitation learning from visual demonstrations. In *Advances in Neural Information Processing Systems*. 3812–3822.

[26] Liang Liu, Clio Andris, Assaf Biderman, and Carlo Ratti. 2010. Revealing Taxi Driver's Mobility Intelligence through His Trace. In *Movement-Aware Applications for Sustainable Mobility: Technologies and Approaches*. IGI Global, 105–120.

[27] Lyft. [n.d.]. Lyft Services. https://www.lyft.com/.

[28] Shuo Ma, Yu Zheng, and Ouri Wolfson. 2013. T-share: A large-scale dynamic taxi ridesharing service. In *ICDE*. IEEE, 410–421.

[29] Andrew Y Ng, Stuart J Russell, et al. [n.d.]. Algorithms for inverse reinforcement learning.

[30] OpenStreetMap contributors. 2017. Planet dump retrieved from https://planet.osm.org . https://www.openstreetmap.org.

[31] Menghai Pan, Weixiao Huang, Yanhua Li, Xun Zhou, Zhenming Liu, Rui Song, Hui Lu, Zhihong Tian, and Jun Luo. 2020. DHPA: Dynamic Human Preference Analytics Framework: A Case Study on Taxi Drivers' Learning Curve Analysis. *ACM Trans. Intell. Syst. Technol.* 11, 1, Article 8 (Jan. 2020), 19 pages. https://doi.org/10.1145/3360312

[32] Menghai Pan, Weixiao Huang, Yanhua Li, Xun Zhou, and Jun Luo. 2020. xGAIL: Explainable Generative Adversarial Imitation Learning for Explainable Human Decision Analysis. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1334–1343.

[33] Menghai Pan, Yanhua Li, Xun Zhou, Zhenming Liu, Rui Song, and Jun Luo. 2019. Dissecting the Learning Curve of Taxi Drivers: A Data-Driven Approach. In *Proceedings of the 2019 SIAM International Conference on Data Mining*. SIAM.

[34] Xi Peng, Xiang Yu, Kihyuk Sohn, Dimitris N Metaxas, and Manmohan Chandraker. 2017. Reconstruction-based disentanglement for pose-invariant face recognition. In *Proceedings of the IEEE international conference on computer vision*. 1623–1632.

[35] Huigui Rong, Xun Zhou, Chang Yang, Zubair Shafiq, and Alex Liu. 2016. The rich and the poor: A Markov decision process approach to optimizing taxi driver revenue efficiency. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM, 2329–2334.

[36] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. 2015. Trust region policy optimization. In *International conference on machine learning*. PMLR, 1889–1897.

[37] Spin. [n.d.]. Spin bike-sharing services. https://www.spin.pm/.

[38] Richard S Sutton, Andrew G Barto, et al. 1998. *Introduction to reinforcement learning*. Vol. 135. MIT press Cambridge.

[39] Luan Tran, Xi Yin, and Xiaoming Liu. 2017. Disentangled representation learning gan for pose-invariant face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1415–1424.

[40] Luan Tran, Xi Yin, and Xiaoming Liu. 2018. Representation learning by rotating your faces. *IEEE transactions on pattern analysis and machine intelligence* 41, 12 (2018), 3007–3021.

[41] Bas Verplanken, Henk Aarts, Ad Van Knippenberg, and Carina van Knippenberg. 1994. Attitude Versus General Habit: Antecedents of Travel Mode Choice 1. *Journal of applied social psychology* 24, 4 (1994), 285–300.

[42] Milton C Weinstein, Bernie O'Brien, John Hornberger, Joseph Jackson, Magnus Johannesson, Chris McCabe, and Bryan R Luce. 2003. 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* 6, 1 (2003), 9–17.

[43] Guojun Wu, Yanhua Li, Jie Bao, Yu Zheng, Jieping Ye, and Jun Luo. 2018. Human-Centric Urban Transit Evaluation and Planning. In *2018 IEEE ICDM*. IEEE, 547–556.

[44] Jing Yuan, Yu Zheng, Liuhang Zhang, XIng Xie, and Guangzhong Sun. 2011. Where to find my next passenger. In *Ubicomp*. ACM, 109–118.

[45] Nicholas Jing Yuan, Yu Zheng, Liuhang Zhang, and Xing Xie. 2012. T-finder: A recommender system for finding passengers and vacant taxis. *TKDE* 25, 10 (2012), 2390–2403.

[46] C Zeng and N Oren. 2014. Dynamic taxi pricing. *Frontiers in Artificial Intelligence and Applications* 263 (01 2014), 1135–1136. https://doi.org/10.3233/978-1-61499-419-0-1135

[47] Xin Zhang, Yanhua Li, Ziming Zhang, and Zhi-Li Zhang. 2020. f-GAIL: Learning f-Divergence for Generative Adversarial Imitation Learning. *arXiv preprint arXiv:2010.01207* (2020).

[48] Xin Zhang, Yanhua Li, Xun Zhou, and Jun Luo. 2019. Unveiling Taxi Drivers' Strategies via cGAIL-Conditional Generative Adversarial Imitation Learning. In *2019 International Conference on Data Mining (ICDM)*. IEEE.

[49] Xin Zhang, Yanhua Li, Xun Zhou, Ziming Zhang, and Jun Luo. 2020. TrajGAIL: Trajectory Generative Adversarial Imitation Learning for Long-term Decision Analysis. In *2020 IEEE International Conference on Data Mining (ICDM)*. IEEE, 801–810.

[50] Yu Zheng, Licia Capra, Ouri Wolfson, and Hai Yang. 2014. Urban computing: concepts, methodologies, and applications. *TIST* 5, 3 (2014), 38.

[51] Xun Zhou, Huigui Rong, Chang Yang, Qun Zhang, Amin Vahedian Khezerlou, Hui Zheng, M Zubair Shafiq, and Alex X Liu. 2018. Optimizing Taxi Driver Profit Efficiency: A Spatial Network-based Markov Decision Process Approach. *IEEE TBD* (2018).

[52] Brian D Ziebart. 2010. Modeling purposeful adaptive behavior with the principle of maximum causal entropy. (2010).

[53] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, and Anind K Dey. 2008. Maximum Entropy Inverse Reinforcement Learning.. In *AAAI*, Vol. 8. Chicago, IL, USA, 1433–1438.