xGAIL: Explainable Generative Adversarial Imitation Learning for Explainable Human Decision Analysis

Menghai Pan[1], Weixiao Huang[1], Yanhua Li[1], Xun Zhou[2], Jun Luo[3]
[1]Worcester Polytechnic Institute, [2]University of Iowa, [3]Lenovo Group Limited
[1]{mpan, whuang2, yli15}@wpi.edu, [2]xun-zhou@uiowa.edu, [3]jluo1@lenovo.com

ABSTRACT

To make daily decisions, human agents devise their own "strategies" governing their mobility dynamics (e.g., taxi drivers have preferred working regions and times, and urban commuters have preferred routes and transit modes). Recent research such as generative adversarial imitation learning (GAIL) demonstrates successes in learning human decision-making strategies from their behavior data using deep neural networks (DNNs), which can accurately mimic how humans behave in various scenarios, e.g., playing video games, etc. However, such DNN-based models are "black box" models in nature, making it hard to explain *what* knowledge the models have learned from human, and *how* the models make such decisions, which was not addressed in the literature of imitation learning. This paper addresses this research gap by proposing xGAIL, the first explainable generative adversarial imitation learning framework. The proposed xGAIL framework consists of two novel components, including Spatial Activation Maximization (SpatialAM) and Spatial Randomized Input Sampling Explanation (SpatialRISE), to extract both global and local knowledge from a well-trained GAIL model that explains how a human agent makes decisions. Especially, we take taxi drivers' passenger-seeking strategy as an example to validate the effectiveness of the proposed xGAIL framework. Our analysis on a large-scale real-world taxi trajectory data shows promising results from two aspects: i) global explainable knowledge of what nearby traffic condition impels a taxi driver to choose a particular direction to find the next passenger, and ii) local explainable knowledge of what key (sometimes hidden) factors a taxi driver considers when making a particular decision.

CCS CONCEPTS

• **Computing methodologies → Inverse reinforcement learning**; *Neural networks*; Markov decision processes.

KEYWORDS

explainable artificial intelligence; generative adversarial imitation learning; human behavior analysis

ACM Reference Format:
Menghai Pan[1], Weixiao Huang[1], Yanhua Li[1], Xun Zhou[2], Jun Luo[3]. 2020. xGAIL: Explainable Generative Adversarial Imitation Learning for Explainable Human Decision Analysis. In *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'20), August 23–27, 2020, Virtual Event, CA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3394486.3403186

1 INTRODUCTION

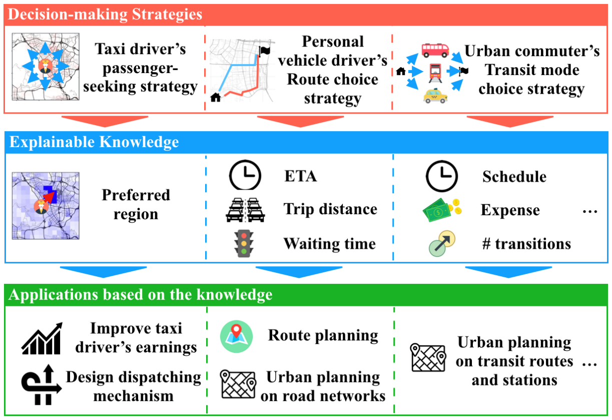

Decision-making Strategies
Taxi driver's passenger-seeking strategy
Personal vehicle driver's Route choice strategy
Urban commuter's Transit mode choice strategy
Explainable Knowledge
Preferred region
ETA
Trip distance
Waiting time
Schedule
Expense
# transitions
...
Applications based on the knowledge
Improve taxi driver's earnings
Design dispatching mechanism
Route planning
Urban planning on road networks
Urban planning on transit routes and stations
...

Figure 1: Applications with Explainable Knowledge from Human Decision-Making Strategies

Humans make daily decisions, based on their own "strategies" (such as taxi drivers' passenger-seeking processes and commuters' transit mode choices). It is crucial to understand what factors humans think about when making decisions, which can greatly facilitate many applications. As three examples shown in Fig. 1, understanding the decision-making strategies from taxi drivers, personal vehicle drivers, and urban commuters can facilitate the service providers (e.g., taxi/ride-hailing companies) to better serve the passengers, and enable the urban planners to design better road networks and transit routes to meet the needs of urban travelers.

Many real-world humans' decision-making processes (e.g., taxi passenger-seeking and transit mode choices) can be modeled as Markov Decision Processes (MDPs) [14, 15, 18, 19, 23, 27, 30–32]. In the MDP model, the human agents' decision-making strategies (e.g., the passenger-seeking strategies) can be captured by sequences of human decisions, which aims to maximize his/her total "rewards". In the literature, inverse reinforcement learning (IRL) and imitation learning (IL) techniques have been applied to recover such reward functions to learn how humans make decisions. For example, Pan et al. propose to use relative entropy based IRL to recover linear reward functions and to dissect drivers' preference dynamics over time [19]. Zhang et al. extend Generative Adversarial Imitation

KDD '20, August 23–27, 2020, Virtual Event, CA, USA

ACM ISBN 978-1-4503-7998-4/20/08…$15.00
https://doi.org/10.1145/3394486.3403186

Learning (GAIL) [10] to conditional GAIL (cGAIL) to unveil taxi drivers' policies by transferring knowledge across taxi drivers and locations [31]. A GAIL model [10] consists of two deep neural networks (DNNs), i.e., a policy net (learning a non-linear policy function) and a reward net (learning a non-linear reward function).

However, there are significant limitations in these solutions. The IRL approaches [4, 33, 34] manually extract features to represent the linear reward function. It is likely to neglect some counter-intuitive while effective features [10]. On the other hand, although a GAIL model [10] is able to integrate high dimensional rich feature sets and better imitate a human agent's strategy, it is hard to explain what knowledge the model has learned from human. This is due to the "black-box" nature of deep neural networks. Therefore, both types of approaches are not much helpful in understanding human agents' strategies. In recent years, a number of approaches have been proposed to interpret machine learning models, such as classifier interpretations [20, 21, 26]. However, none of them focuses on the explanation of knowledge learned by imitation learning models (e.g., GAIL) from *human-generated spatial-temporal* data, such as vehicle trajectories.

In this paper, we make the first attempt to address the above challenges by proposing xGAIL, a novel explainable Generative Adversarial Imitation Learning model for learning both i) human decision-making strategies (as deep neural networks) to mimic how a human behaves, and ii) human-understandable knowledge to explain how a human (and the learned model) makes decisions. The proposed xGAIL framework consists of two novel components, Spatial Activation Maximization (SpatialAM) and Spatial Randomized Input Sampling Explanation(SpatialRISE), which extracts both global and local knowledge from a pre-trained GAIL model that has learned a human agent's decision-making strategy. Especially, we take taxi drivers' passenger-seeking strategies as an example to validate the effectiveness of our proposed xGAIL framework. Our main contributions are summarized as follows:

- We formulate human agents' decision-making processes (using taxi drivers' passenger-seeking processes as an example) as Markov Decision Processes (MDPs), and to inversely learn each agent's decision-making strategy by a Generative Adversarial Imitation Learning (GAIL) model.
- We propose an explanation framework with both global and local interpretation mechanisms, i.e., Spatial Activation Maximization (SpatialAM) and Spatial Randomized Input Sampling Explanation (SpatialRISE), to explain what knowledge a GAIL model learns so as to generate a specific decision-making strategy.
- We conduct a case study using real-world taxi driver's trajectory data to validate our framework. Our analysis shows interesting results from two facets: i) global explainable knowledge of what nearby traffic condition impels a taxi driver to choose a particular direction to find the next passenger, and ii) local explainable knowledge of what key (sometimes hidden) factors a taxi driver considers when making a particular decision. We made our code and unique data set[1] available to contribute to the research community.

The remainder of the paper is organized as follows. In Section 2, we define our problem and outline our system framework. Section 3 presents our approach for data preprocessing. We elaborate GAIL and the evaluation method for inverse policy learning in Section 4. Section 5 introduces the xGAIL framework to explain the learned GAIL model, and Section 6 evaluates our framework. Section 7 presents the related work and Section 8 concludes the paper.

2 OVERVIEW

In this section, we model the sequential human decision-making process as a Markov Decision Process (MDP) and introduce Generative Adversarial Imitation Learning (GAIL) model as a way to learn the human decision-making strategy using deep neural networks. We also define the strategy explanation problem, and outline our proposed xGAIL solution framework. Along with the paper, we use a concrete example, i.e., taxi driver passenger-seeking process as an example to illustrate the strategy explanation problem. Actually, **without loss of generality, our proposed xGAIL model can be applied to any general human sequential decision analysis problems**, such as commuter transit mode choice, etc.

2.1 Sequential Human Decision-Making Processes as MDPs

Markov decision processes (MDPs) [25] provide a mathematical framework for modeling decision-making processes. An MDP includes an agent as the decision maker and an environment that interacts with the agent. An MDP is defined as a 5-tuple $\langle S, A, T, R, \gamma \rangle$, where S is the state space, A is the action space, $T : S \times A \times S \mapsto [0, 1]$ represents 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, $R : S \times A \mapsto \mathbb{R}$ is the reward function of each state-action pair, and $\gamma \in (0, 1]$ is the discount factor. An agent at a state $s \in S$ makes a decision of taking an action $a \in A$ following a memoryless policy π. The memoryless policy π is a function that specifies a probability distribution on the action to be executed in each state, defined as $\pi : S \times A \mapsto [0, 1]$. Taking a passenger-seeking process as an example, a taxi driver makes a sequence of decisions about which directions (as actions) to go based on his/her own decision-making strategy. The MDP components of a passenger-seeking process are highlighted as follows.

- *State $s \in S$*: A state of a taxi driver can be uniquely defined by the spatial location and time stamp.
- *Action $a \in A$*: There are 9 possible actions that a taxi driver can choose at a state s, including traveling to 8 neighboring directions, and staying at the current location.
- *Reward $R(s, a)$*: The reward that a taxi driver receives follows an inherent function $R(s, a)$ to evaluate an action a taken at a state s.
- *Policy $\pi(a|s)$*: A policy $\pi(a|s)$ of a taxi driver is a mapping from a state s to an action a, i.e., the probability distribution of choosing an action a given a state s.

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

Decision-making Strategy Learning with Generative Adversarial Imitation Learning (GAIL). Given a large amount of trajectory data from a human agent (e.g., a taxi driver), each **trajectory**

[1] https://github.com/paperpublicsource/xgail

is defined as a sequence of decisions, namely, state-action pairs, $\tau = [(s_0, a_0), (s_1, a_1), ..., (s_L, a_L)]$, with L as the trajectory length. Generative adversarial imitation learning (GAIL) [10, 33] was proposed to inversely learn both the policy function $\pi(a|s)$ and reward function $R(s, a)$ employed by the agent. As defined in [10], the strategy learning problem can be modeled as the following constrained optimization problem, namely, finding the policy π with maximum causal entropy (eq.(1)), and finding the reward function R such that the expected reward of a trajectory under π matches that under the empirical policy π_E from observed data (enforcing eq.(2)).

MaximumCausalEntropy Inverse Reinforcement Learning

$$\max_{R} \min_{\pi} : \; -H(\pi), \tag{1}$$

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

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

where $H(\pi) = \mathbb{E}_{\pi}[\sum_{t=0}^{T} \gamma^t(-\log \pi(a_t|s_t))]$ is the γ-discounted causal entropy π, $\mathbb{E}_{\pi}[R(s, a)] = \mathbb{E}_{\pi}\left[\sum_{t=0}^{T} \gamma^t R(s_t, a_t)\right]$ represents the expected reward of a trajectory under the policy π, and π_E (empirical policy) represents the policy observed from the collected data. GAIL [10] proves that the above maximum causal entropy inverse reinforcement learning problem is equivalent to solving a minimax problem (eq.(4)) with the objective as a Jensen-Shannon (JS) divergence as follows.

$$\max_{R} \min_{\pi \in \Pi} -\lambda H(\pi) + \mathbb{E}_{\pi}[\log(R(s, a))] + \mathbb{E}_{\pi_E}[\log(1 - R(s, a))], \tag{4}$$

with Π as the policy probability simplex space, guaranteeing constraint eq.(3), λ as the Lagrangian multiplayer. As a result, a generative adversarial networks (GANs) framework [8] is naturally employed to solve the strategy learning problem with a generator network G (equivalent to the policy function π) and a discriminator network D (equivalent to the reward function R). However, the policy and reward functions are learned as two deep neural networks, thus it is hard to explain what knowledge and aggregated features the two networks have learned from human agents' trajectory data, i.e., depending on what complex factors, human agents make decisions. Below, we formally define the strategy explanation problem and outline our solution framework.

2.2 Strategy Explanation Problem and Solution

Problem Definition. We aim to extract human understandable knowledge from the learned policy (π) and reward (R) nets in the GAIL model to understand why and how a human agent (e.g., a taxi driver) makes a certain decision a at a state s.

Solution Framework. The proposed strategy explanation problem is challenging, because the policy function π learned from GAIL is a deep neural network (DNN), which as a blackbox model is hard to explain. Fig. 2 outlines our proposed explainable generative adversarial imitation learning (xGAIL) framework (using taxi driver passenger-seeking strategy as an example). xGAIL takes two sources of data as inputs and consists of three stages, including (1) data preprocessing, (2) GAIL, and (3) Strategy Explanation Module, which will be detailed below from Sec 4 to Sec 5.

3 STAGE 1: DATA PREPROCESSING

In this section, we take a taxi driver passenger-seeking process as an example to illustrate the data preprocessing mechanism. The novelty of the data preprocessing is our design of the state observation. Note that the data preprocessing approaches can be easily applied to other scenarios, such as commuter transit mode choice, personal vehicle route choice, etc.

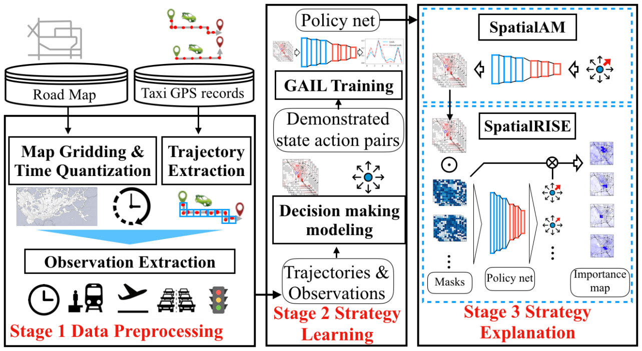

Road Map
Taxi GPS records
Map Gridding & Time Quantization
Trajectory Extraction
Observation Extraction
Stage 1 Data Preprocessing
Policy net
GAIL Training
Demonstrated state action pairs
Decision making modeling
Trajectories & Observations
Stage 2 Strategy Learning
SpatialAM
SpatialRISE
⊙
⊗
Masks
Policy net
Importance map
Stage 3 Strategy Explanation

Figure 2: xGAIL Solution Framework

3.1 Data Description

We employ two datasets, the GPS dataset and the road map dataset, in this study.

GPS dataset. Taxi trajectory dataset records were collected from July to September in 2016 in Shenzhen, China. The dataset recorded the traces from 17,877 unique taxis. For each taxi, a GPS point was collected every 30 seconds on average. There were a total of 51,485,760 GPS points generated in a day. Each GPS point contains five attributes, a unique taxi ID, a timestamp, a latitude, a longitude, and a passenger indicator. The passenger indicator is a binary value with 1, indicating the taxi was occupied, and 0, indicating it was vacant.

Road map dataset. The road map data were collected from OpenStreetMap [1] for the region of Shenzhen in China, ranging from 22.44° to 22.87° in latitude and 113.75° to 114.63° in longitude. There are 455,944 road segments collected in this region.

3.2 Map and Time Quantization

The human agents (i.e., taxi drivers) traverse the spatial and temporal spaces when seeking and serving passengers. We define those states that a driver can visit by i) dividing and discretizing Shenzhen city into equal side-length *(spatial)* grid cells with a given side-length $l = 0.01°$ in latitude and longitude, ii) a day into 288 five-minute *(temporal)* intervals. By eliminating grid cells in the ocean and unreachable regions in the city, there are a total of 1,934 remaining cells that are well-connected by the road network. Each cell is represented 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 State Observation of A Human Agent

Each human agent makes a sequence of decisions to traverse geographical locations over time when seeking for passengers. Each decision (e.g., which direction to go) made by the driver is based on various features (such as traffic) in the surrounding urban environment of the nearby area, referred to as the state observation of the driver. Given a spatial-temporal state s, we model a taxi driver's observations as the state observation $\boldsymbol{O}_s = [O_1, O_2, O_3, O_4, O_5]$, with

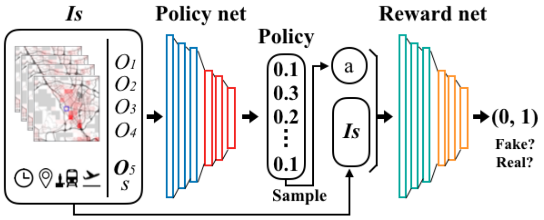


Figure 3: GAIL model with a policy net π and a reward net R trained as a GAN.

five statistics for the surrounding 15 × 15 grid cells of s, including O_1 the number of pickups, O_2 the traffic volume, O_3 traffic speed, O_4 the waiting time; and O_5 the distances to points of interests (POIs), such as train stations, airports, shopping malls, ports, and hospitals in the city. For example, Fig. 4a shows an example of the traffic volume observation map O_2 of a state (the blue box at the center).

4 STAGE 2: STRATEGY LEARNING WITH GAIL

Now, we introduce the structure of the generative adversarial imitation learning (GAIL) model [10] for leaning a human agent's decision-making strategy (from his/her generated data).

A GAIL model trains a generator network for the policy function $\pi(a|s)$, and a discriminator network for the reward function $R(s, a)$ (see Fig. 3).

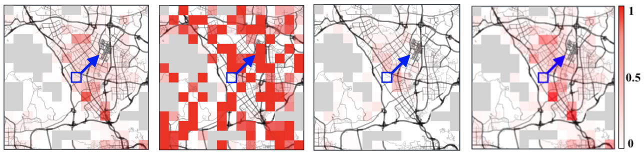
The generator network (i.e., policy) takes the state observation $\boldsymbol{O}_s$, the high dimensional feature maps, as the input, and outputs the decision-making policy $\pi(a|s)$. Based on the learned policy, an action (namely, a direction to go to find the next passenger) is then randomly chosen.

The discriminator network (i.e., reward) takes both the state observation $\boldsymbol{O}_s$ 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.

When implementing GAIL, we employ convolutional neural networks [13]. For the policy and reward nets, they both consist of three convolutional layers with ReLU activation functions. Given the input state observation with the size of 5 × 15 × 15 (5 channels are the number of pickups, traffic volume, speed, waiting time, and distances to POIs), we use a kernel size of 3×3 for the convolutional layers with padding of size 1. The sampling process for actions makes the entire network no longer differentiable, so that it is not trainable by backpropagation [9]. We, thus, use the Reinforcement Learning (RL) based approach [25] to train the network, i.e., using the output of reward net as signals to update the policy net.

5 STAGE 3: STRATEGY EXPLANATION WITH xGAIL

The trained GAIL model recovers the taxi driver's strategy. Given a state and its state observation, the policy net predicts an action just as the driver does. However, the policy and reward nets both are "black boxes". The inscrutable internal processes cause considerable difficulty in explaining why and how the nets generate that specific strategy and make that specific action. In other words, what "knowledge" the nets learned remains unknown. To solve this problem, in this section, we formally propose Explainable Generative Adversarial Imitation Learning (xGAIL), a strategy explanation framework to extract human understandable knowledge from the trained nets. Our xGAIL framework is designed to provide both global and local explanations for the learned policy and reward nets. We use the policy net as an example to illustrate xGAIL. The xGAIL framework consists of the global explanation and the local explanation. The global explanation aims to reveal the state observation $\boldsymbol{O}^*_s(a)$ which leads to the highest probability of choosing a target action a, and the local explanation extracts the most effective local features.



(a) Real state $\pi(a_1) = 0.30$ **(b) AM** $\pi(a_1) = 0.98$ **(c) L_2 AM** $\pi(a_1) = 0.42$ **(d) SpatialAM** $\pi(a_1) = 0.41$

Figure 4: O^*_2 obtained from different AM approaches

5.1 SpatialAM: Global Explanation Method for GAIL

Design Goal. Given a policy net π, the goal of the global explanation is to extract the state observation $\boldsymbol{O}^*_s(a)$ that maximizes the probability of a target action a in policy $\pi(a|s)$ among all the actions. It thus can be formulated as below,

$$\boldsymbol{O}^*_s(a) = \arg\max_{\boldsymbol{O}_s} \pi(a|\boldsymbol{O}_s). \quad (5)$$

Limitations of state-of-the-art works. This objective function has been extensively studied in the literature as an activation maximization (AM) problem [17, 24, 28]. For example, the AM model from [24] aims to find the image that maximizes the likelihood of being classified as a goose, which introduces an L_2 regularization term to guarantee the obtained image is close to a real image, without overfitting. The optimal input can be obtained by gradient ascent via back propagation.

However, for our policy net explanation problem, the input traffic state observations possess intrinsic geographic characteristics, i.e., spatial auto-correlations across grids. As a result, the activation maximization model with L_1 and L_2 norm regularization cannot preserve these spatial auto-correlations in the obtained $\boldsymbol{O}^*_s(a)$. Fig. 4(b)(c) show $\boldsymbol{O}^*_s(a)$ (in traffic volume distribution) obtained using AM without regularization and with L_2 norm regularization. Comparing to the real state observation in Fig. 4(a), neither of them captures the real traffic volume distribution. To tackle this problem, we propose Spatial Activation Maximization (SpatialAM).

Spatial Activation Maximization (SpatialAM). To enable activation maximization to output $\boldsymbol{O}^*_s(a)$ that preserves the spatial auto-correlation pattern presented in the real world observations, we introduce a new spatial regularization term into the AM problem (to capture the realness of a state observation) as

$$Realness(O_s(a)) = -Dist(\boldsymbol{O}_s(a), \bar{\boldsymbol{O}}_s(a)), \quad (6)$$

where $Dist(\boldsymbol{O}_s(a), \bar{\boldsymbol{O}}_s(a))$ is the mean square distance of $\boldsymbol{O}_s(a)$ from $\bar{\boldsymbol{O}}_s(a)$, which is the mean state observation from the demonstration

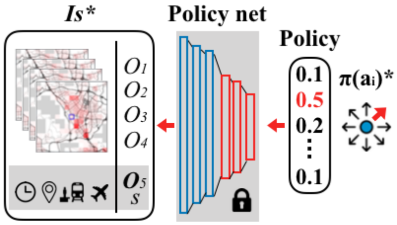


Figure 5: SpatialAM

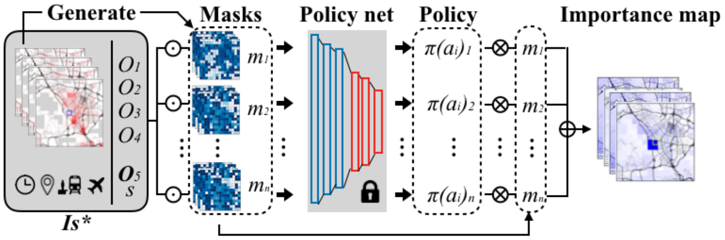


Figure 6: SpatialRISE

data, and captures what a real state observation looks like. And the objective function of SpatialAM is

$$O_s^*(a) = \arg\max_{O_s}\{\pi(a|O_s) + \lambda \cdot Realness(O_s)\}, \tag{7}$$

where λ is the weight of the regularization term. The introduced spatial regularization term guides the activation maximization problem to find a state observation that maximizes the probability of a, and minimizes the difference to the mean state observation $\bar{O}_s(a)$. Fig. 4(d) shows $O_s^*(a)$ obtained by SpatialAM, is clearly closer to the real state observation (and with a higher probability of 0.41 for the target action a).

5.2 SpatialRISE: Local Explanation Method for GAIL

Post-hoc local interpretation approaches help us learn the local explanations of neural networks. We aim to answer the question "which areas of the input layer play important roles in producing the policies in the learned policy and reward nets". One of the basic ideas behind the local explanation is to generate an *importance map*, which can show how important each entry of the input is to the prediction of the model.

5.2.1 Spatial randomized input sampling explanation. Randomized Input Sampling Explanation (RISE) [20], as a local interpretation approach, can discover the importance map of the input by probing the model with randomly masked versions of the input image and obtaining the corresponding outputs. The masks are then aggregated into the importance map according to the corresponding outputs. However, when being applied to the GAIL model that learns spatial features, RISE has two limitations. First, it does not segment the input observation map based on intrinsic geographic characteristics. As a result, the high importance areas identified by RISE are large and do not align meaningfully with the functional region of the city. Second, RISE employs a bi-linear interpolation method to generate the mask values, which ignores the spatial auto-correlation of the features. This may lead to drastically different importance values being assigned to highly similar locations in the same functional region of the city. To deal with these challenges, we propose a novel spatial importance discovery model named **SpatialRISE** to discover the importance of geographic regions with respect to a specified output in the learned GAIL model. It consists of three steps: map segmentation, mask generation, and importance map generation.

Algorithm 1 Observation map segmentation

```
Input: observation map $O$, threshold $Gi^t$ of $|Gi^*|$;
Output: Clusters $C$ for all grids;
$C = \{\}, k = 0$;
Calculate $Gi^*$ for each grid $g_i$ based on $O$;
for Each grid $g_i$ in the observation map do
    if $|Gi^*| > Gi^t$ then
        if $g_i$ is neighboring to any grid in an existing cluster $c_j (0 < j \le k)$ and have the same sign of $Gi^*$ then
            Add $g_i$ to the cluster $c_j$;
        else
            $k = k + 1$;
            Create a new cluster with $g_i$ as the first grid in the cluster $c_k$;
        end if
    else
        $k = k + 1$;
        Consider $g_i$ itself as an independent cluster $c_k$;
    end if
end for
Return the clusters $C = \{c_1, ..., c_k\}$ for all grids.
```

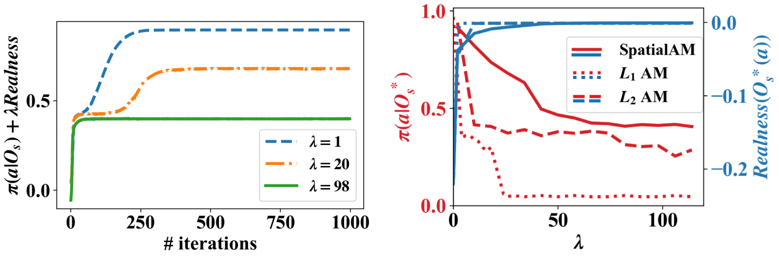
5.2.2 Map segmentation. As LIME [21] tries to discover the importance of the meaningful super-pixels in the image, we want to discover the importance of the functional regions of the city. To do this, we first segment the map into functional regions. Within each functional region, the observation values are expected to have strong spatial auto-correlation.

We measure the strength of spatial auto-correlation at each location using a Local Getis-Ord Gi^* statistic[7]. The local Gi^* statistic can be calculated via eq.(8),

$$Gi^* = \frac{\sum_{j=1}^{n} w_{i,j} x_j - \bar{X} \sum_{j=1}^{n} w_{i,j}}{S\sqrt{\frac{n\sum_{j=1}^{n} w_{i,j}^2 - (\sum_{j=1}^{n} w_{i,j})^2}{n-1}}}, \tag{8}$$

where x_j is the observation value at location j, $w_{i,j}$ is the spatial neighborhood indicator between location i and j, and n is the total number of locations, $\bar{X} = \sum_{j=1}^{n} x_j / n$, and $S = \sqrt{\sum_{j=1}^{n} x_j^2 / n - (\bar{X})^2}$. In this work, $w_{i,j} = 1$ if i and j are geographically neighboring to each other, otherwise $w_{i,j} = 0$. A large positive local Gi^* score indicates a hotspot (high observation values clustered), and a small negative local Gi^* score indicates a coldspot (low observation values are clustered). Thus, we can segment the map into clusters according to the local Gi^* scores of the grids given cut-off threshold. The cluster algorithm is shown in Algorithm 1.

5.2.3 Mask generation. With the segmented observation maps, we are able to take the spatial auto-correlation into consideration to



(a) Learning curves with different λ's

(b) Impact of λ in SpatialAM

Figure 7: SpatialAM evaluation results

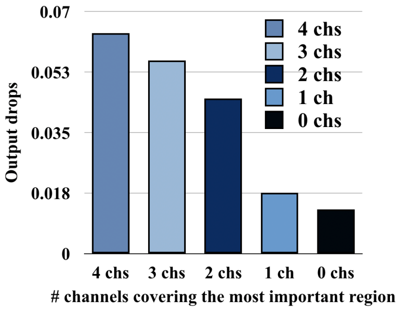


Figure 8: Importance map evaluation

generate masks. The mask values within each cluster have strong spatial auto-correlation. To generate mask values for the grids in each cluster, we first randomly sample an overall trend $tr \in \{1, 0\}$ for each cluster, i.e., preserving or covering the original observation values, with the covering probability p. Then, inside each cluster, we assign mask value, $m(i)$, to grid i according to eq.(9),

$$m(i) = \begin{cases} 1 - \alpha * random(0,1), \text{if } tr = 1; \\ \alpha * random(0,1), \text{if } tr = 0, \end{cases} \tag{9}$$

where $\alpha \in (0, 1)$ is the weight of the randomness. The mask generation method can adapt to all kinds of random distributions. In this paper, we employ uniform distribution to generate random values, which makes sure that the mask values of the grids in the same cluster are within an expected range.

5.2.4 Importance map generation. Once we generate a set of masks, we can estimate the importance map for each observation map. Since we introduce randomness in each cluster, the importance map produced by our proposed SpatialRISE can tell the pixel-wise importance.

The framework of SpatialRISE is shown in Fig. 6. The input of the policy net is I_s, and $\pi(a|I_s)$ is the output of the policy net regarding action a. Let $m : \Lambda \longrightarrow [0, 1]$ be a random mask, and M be the population of all possible masks following distribution D. $I_s \odot m$ is the masked input. Then the importance map $Ipt_{I_s,\pi,a}$ of I_s regarding the output of action a in policy net π can be calculated by the weighted sum of the masks with the model output $\pi(a|I_s \odot m)$ as the weight for each mask m:

$$Ipt_{I_s,\pi,a} = \frac{1}{\mathbb{E}(M)} \sum_{m \in M} \pi(a|I_s \odot m) \cdot m \cdot P[M = m]. \tag{10}$$

The intuition is that $\pi(a|I_s \odot m)$ is high if entries of I_s preserved by mask m are important. Empirically, we can estimate $Ipt_{I_s,\pi,a}$ by sampling a set of N masks M' according to D:

$$Ipt_{I_s,\pi,a} \approx \frac{1}{\mathbb{E}(M') \cdot N} \sum_{m \in M'} \pi(a|I_s \odot m) \cdot m. \tag{11}$$

6 EVALUATION: A CASE STUDY ON PASSENGER-SEEKING STRATEGIES

In this section, we evaluate our xGAIL framework on a pre-trained GAIL model to interpret what knowledge the policy net (learned from GAIL) has learned. We have released our code and data[2] to support the reproducibility.

6.1 Model Performance Evaluation

We conduct experiments to evaluate the effectiveness and efficiency of our proposed SpatialAM and SpatialRISE.

6.1.1 SpatialAM model evaluation. We evaluate SpatialAM algorithm by comparing it with baselines and examining the impact of the choice of regularization weight λ.

Comparison with Baselines. We first compare our SpatialAM model with two baseline models, including Activation Maximization with L_1-norm and L_2-norm regularization terms [2]. Fig. 7 shows the comparison results about the *learned policy* and *the realness of state observation* (measured by the realness regularization term $Realness(O_s^*(a))$) with different regularization weight λ ranging from 1 to 100. It clearly indicates that when increasing λ, the policy probability $\pi(a|O_s^*)$ decreases, and the realness of state observation increases for all methods, which makes sense because λ controls how much to maximize the policy vs. maximize realness of the solution state observation. However, note that when λ is sufficiently large (i.e., $\lambda \geq 98$), all three approaches tend to the similar high realness, but SpatialAM can always find state observations with higher policy probability than the baselines. The comparisons show that SpatialAM can generate like real state observations with higher policy probabilities, thus, it provides a better view of the global explanation of what an ideal state observation looks like for the human agent to choose a target action a.

Impact and Choice of λ. Fig. 7a shows an example of the learning curve of SpatialAM with $\lambda = 1, 20$, and 98, respectively, the y-axis is the output of the objective function defined in eq.(7), the x-axis is the number of iterations. The results illustrate that the learning process of our proposed SpatialAM converges to a state observation $\boldsymbol{O}_s^*$ with monotonically increased objective function by gradient ascent.

λ is designed to balance the trade-off between maintaining the spatial auto-correlation in the generated state observation, and obtaining maximum output policy. Fig. 7b shows the optimal output with different settings of λ, which illustrates that, with the increase of λ, the maximum policy $\pi(a|\boldsymbol{O}_s^*)$ obtained by SpatialAM decreases, and the realness regularization term $Realness(\boldsymbol{O}_s^*(a))$ increases. Fig. 7b shows that when λ is sufficiently large, i.e., $\lambda \geq 98$, the spatial regularization term converges to a large realness regularization term $Realness(\boldsymbol{O}_s^*(a)) \geq -5e^{-3}$, i.e., small distance from true state observation. On the other hand, the policy $\pi(a|\boldsymbol{O}_s^*)$ converges to 0.41. As a result, we select $\lambda \geq 98$.

[2] https://github.com/paperpublicsource/xgail

Table 1: Gap between the maximum policy from real states and the policy from SpatialAM

Mean gap	Max gap	Min gap
0.2882	0.3885	0.1113

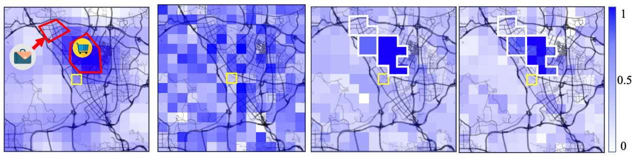
To better illustrate the ability of SpatialAM in generating like-real observations and maximizing the policy probability of a target action, Table 1 summarize the statistical results, by comparing the maximum policy obtained by SpatialAM vs that from the dataset. Overall different target actions, SpatialAM can generate observations with on average 0.2882 (i.e., the mean gap) more policy probabilities than that from the dataset.

6.1.2 SpatialRISE model evaluation. We first quantitatively evaluate the importance map generated by SpatialRISE, then we compare spatialRISE with RISE, PixelRISE, and SolidRISE, respectively. **Importance map evaluation.** The question we aim to answer in this evaluation is, "**Is the important region found by SpatialRISE really important to the maximum policy?**" The more important a region is, the greater its impact on the output policy should be. We propose a measurement that the impact of a region on the output policy can be quantified by the drop amount of the output policy when covering the region in a channel, i.e., a state observation obtained by SpatialAM. Therefore, we make comparisons in the output policy drops between covering the most important regions found by SpatialRISE and covering other regions. Fig. 8 shows the results of the output drops when covering different regions. The x-axis is the number of channels, i.e., state observations, where the most important regions found by SpatialRISE are covered. For example, "3 chs" means that in 3 out of 4 channels the most important regions found by SpatialRISE are covered, while in the remaining channel a region other than the most important one is covered. Since there are multiple regions other than the most important one, we just show the maximum output drops as the y-axis in Fig. 8. The results prove that covering the most important regions in all 4 channels leads to the most significant output policy drop. In other words, the impact of the SpatialRISE detected regions is much greater than the impacts of other regions. Thus, the important region found by SpatialRISE is the key to the maximum policy.

Comparison experiments. We compare our proposed SpatialRISE with the following baselines:

- **RISE** [20]: masks are generated with bilinear interpolation;
- **RISE with pixel-independent masks(PixelRISE)**: The mask value in each grid is independent with each other;
- **RISE with solid cluster masks(SolidRISE)**: We first partition the underlying spatial region into clusters using Algorithm 1. The masks are generated with respect to the clusters, such that all grids in the same cluster are assigned with the same random number.

Comparison Results. For all baselines, we set the zero mask value probability $p = 0.3$. Taking the observation channel of "the number of pickups" as an example, Fig. 9a shows the importance map generated from the original RISE. Although it provides the importance of grids, it does not take the underlying geographic information into consideration. Thus, the city functional regions, such as Dalang business center and Longhua Market marked by the red boxes in Fig. 9a, cannot be detected. Fig. 9b is the importance map generated by RISE with pixel-independent masks, which scatters noise with no reliable information of importance. The reason is that the pretrained policy net is not sensitive to the change of individual pixels with the pixel-independent masks. Fig. 9c represents the importance map generated via RISE with solid cluster masks. The clusters extracted by Algorithm 1 (the white boxes in Fig. 9c) identify the two nearby functional regions, as highlighted in Fig. 9a. However, since the mask values in the same cluster are the same, the results can only provide cluster-level importance, rather than pixel-wise importance interpretation in finer granularity. Fig. 9d is the importance map using our proposed spatialRISE. It is able to distinguish the geographic functional regions, as well as provide the pixel-wise importance, i.e., the importance score of each pixel integrates both region-level and pixel-level importance information.



(a) RISE **(b) PixelRISE** **(c) SolidRISE** **(d) SpatialRISE**

Figure 9: Importance map on O_1 from different methods

6.2 Explainable knowledge Learned from Passenger-Seeking Strategies

To interpret how the input observations affect the taxi driver's passenger-seeking policy, we generate optimal state observations $O_1^*, O_2^*, O_3^*, O_4^*$ in different locations, which maximize the policy on a target action via SpatialAM, and use SpatialRISE to generate importance maps for state observations. By examining the results using SpatialAM and SpatialRISE for different locations, we observe and present three interesting findings which explain how human taxi drivers make decisions for seeking passengers.

6.2.1 Experimental results of SpatialAM. Fig. 10a-d & Fig.10i-l present the generated observation maps maximizing the policy at location $loc1$ on action a_1 (northeast direction) and $loc2$ on action a_5 (southwest direction) respectively. Taking Fig. 10a as an example, Fig. 10a-d are four observation maps of $O_1^*(a_1)$ (number of pickups), $O_2^*(a_1)$ (traffic volume), $O_3^*(a_1)$ (traffic speed), and $O_4^*(a_1)$ (waiting time), respectively. Except for the unreachable grey area, the color map spanning from white to red corresponds to the small to large observation values. For example, a grid cell in Fig. 10a with white color means that in this grid cell the number of pickups is close to 0, and a grid cell with red color means the number of pickups in it is close the the maximum of the map. These plots show the global observations of the four input features under which the driver's likelihood to go northeast at $loc1$ and southwest at $loc2$ is the highest.

6.2.2 Experimental results of SpatialRISE. The importance maps produced by SpatialRISE for the observations generated by SpatialAM at location $loc1$ on action a_1 and $loc2$ on action a_5 are shown in Fig. 10e-h & Fig.10m-p. Except the unreachable grey area, the color map spanning from white to blue indicates the importance

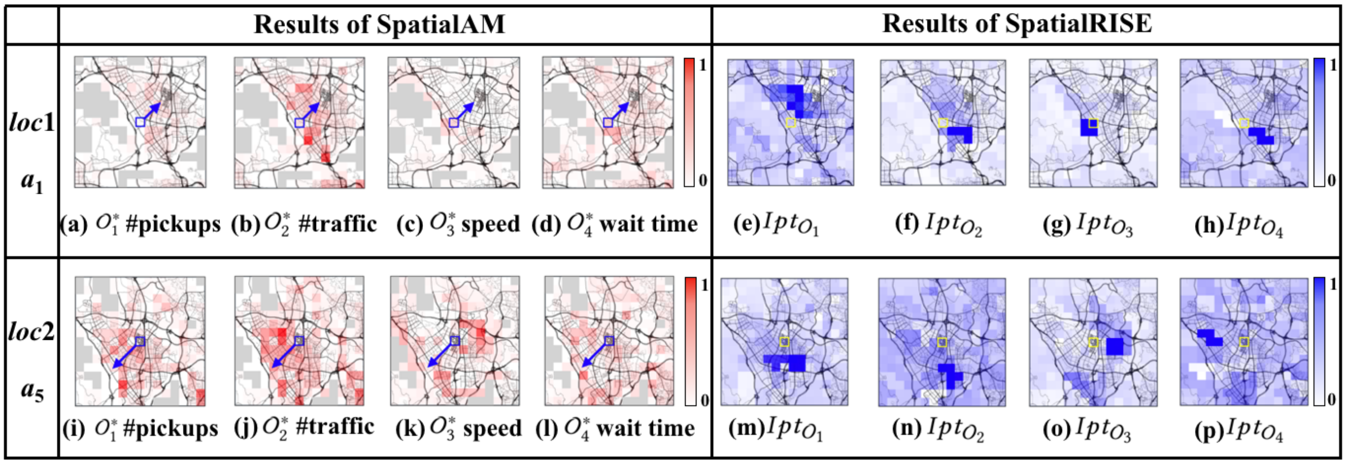


Figure 10: Results of SpatialAM and SpatialRISE

value ranging from 0 to 1. A grid cell in Fig. 10e, for example, with dark blue color (value close to 1) means that the value of $O_1^*(a_1)$ at this grid cell is quite important for obtaining the maximum policy on action a_1, namely, the taxi driver considers the number of pickups in this particular grid cell heavily, when making decisions.

6.2.3 Knowledge learned from SpatialAM and SpatialRISE results. Integrating the results of SpatialAM and SpatialRISE, we observe the following interesting findings which explain what human agents (i.e., taxi drivers) think about when making decisions:

Finding 1: Taxi drivers prefer the regions with large numbers of pickups. Take the case of location $loc1$ and action a_1 as an example, Fig. 10a shows that there are many pickups in the grids in the direction of the northeast. Fig. 10e indicates that the taxi driver pays her attention to the grids in the direction of the northeast where there is Longhua Market. Recall that both Fig. 10a and Fig. 10e are based on the maximized policy of the action towards northeast, and generated time slot for the state observation in Fig. 10a is 6:50 pm- 6:55 pm which is the evening rush hours. Thus, the taxi driver prefers northeast direction primarily because of a large number of pickups near Longhua Market in the evening. A similar observation can be found in the case of $loc2$ and action a_5 from Fig. 10i & Fig. 10m.

Finding 2: Taxi drivers prefer to avoid visiting regions with high traffic volume and long waiting time. For the case of location $loc1$ and action a_1, Fig. 10b and Fig. 10d suggest that the traffic volume and waiting time in the direction of the southeast are high. Fig. 10f and Fig. 10h show that the driver cares much about the grids in the southeast direction, where there is Shenzhen North Railway Station. As a result, the high traffic volume and waiting time near the railway station propel the driver choosing to go another direction (northeast in this case). The possible reason is that the high traffic volume and long waiting time indicate traffic jams near the railway station. The driver wants to avoid approaching these areas. A similar finding can be interpreted in the case of location $loc2$ and action a_5 from Fig. 10j & l and Fig. 10n & p.

Finding 3: Taxi drivers do not prefer regions with high traffic speeds. From the case of location $loc1$ and action a_1, Fig. 10c and Fig. 10g indicate that the high traffic speed in the southwest direction leads the driver to go to another direction, i.e., northeast. This is somehow counter-intuitive because a high traffic speed usually means a good traffic condition, which taxi drivers should prefer. However, in fact a high traffic speed probably also imply that the path is for vehicles only, such as highway and expressway. Therefore, there are few pedestrians. Taxi drivers know that they are unlikely to find passengers.

7 RELATED WORK

In this section, we summarize the literature from two related areas, imitation learning (IL) and explainable artificial intelligence (XAI).

Imitation learning (IL), also known as learning from demonstrations, inverse reinforcement learning (IRL), inversely recovers the agent's policy and reward functions from the collected demonstrations. IL approaches [4, 33, 34] have been proposed based on different principles, including maximum entropy, maximum causal entropy, and relative entropy principles [4, 33, 34]. All the approaches assume that the underlying reward function is a linear function and features have to be manually extracted. Generative adversarial imitation learning (GAIL) [10], and its extension works cGAIL [31] and adversarial IRL [6] 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. These works either rely on manually extracted features or learn policies through black-box models (i.e., DNNs) which make the processes hard to explain the key features human agents are considering. In this work, we make the first attempt to tackle this challenge.

Explainable artificial intelligence (XAI) as an emerging topic has been extensively studied in recent years [5, 16, 29], which all aim to provide explanations of what DNNs capture. In the category of post-hoc global explanation, Activation Maximization (AM) aims to generate an input that maximizes the activation of a neuron in a network [17, 24, 28]. Karpathy et al. provide an analysis of Long Short-Term Memory (LSTM)'s representations, predictions, and error types through character-level language models [12]. Kádár et al.'s word level interpretation approach estimates the amount of contribution of individual tokens in the input to the final prediction [11]. Augasta et al. introduce a new neural network rule extraction algorithm *RxREN* to overcome the lack of explanation capability of neural network models [3]. The algorithm prunes the insignificant input neurons and constructs the classification rules only with significant input neurons based on reverse engineering technique [3]. In addition, other research focuses on local explanation. Ribeiro et al. identify an interpretable model, *LIME*, over the interpretable representation of a binary vector indicating the presence or the absence that is locally faithful to the classifier [21]. Petsiuk et al. establish *RISE* which can generate the importance maps through tons of pixel masks [20]. Ribeiro et al. put forward high-precision rules representing local "sufficient" conditions for predictions [22]. Differing from these works, we focus on a framework to explain what GAIL model learned from human-generated spatial-temporal data.

8 CONCLUSION

Generative adversarial imitation learning (GAIL) achieves great success in learning human decision-making strategies from demonstrated data using deep neural networks (DNNs). However, such DNN-based models are hard to explain what aggregate knowledge the models learned from data. To bridge this gap, we propose the explainable generative adversarial imitation learning (xGAIL) framework which includes two novel techniques, namely, Spatial Activation Maximization (SpatialAM) and Spatial Randomize Input Sampling Explanation (SpatialRISE). They can learn global and local explainable spatial-temporal features, respectively. In particular, we take taxi drivers' passenger-seeking strategy as an example to validate the effectiveness of the xGAIL framework. Our analysis of a large-scale real-world taxi trajectory data shows interesting results from two perspectives i) global explainable knowledge of what nearby traffic condition impels a taxi driver to choose a particular direction to find the next passenger, and ii) local explainable knowledge of what key (sometimes hidden) factors a taxi driver considers when making a particular decision. All the knowledge we found sheds light on how to promote taxi drivers' well-being and improve the quality of taxi services, e.g., reducing the waiting time, etc. Moreover, our proposed xGAIL framework can be naturally applied to other urban decision-making processes, such as commuter transit mode choice, and personal vehicle route choice.

9 ACKNOWLEDGEMENTS

Menghai Pan, Weixiao Huang and Yanhua Li were supported in part by NSF grants IIS-1942680, CNS-1657350 and CMMI-1831140. Xun Zhou was partially supported by a grant from the SAFER-SIM University Transportation Center.

REFERENCES

[1] OpenStreetMap. http://www.openstreetmap.org/.

[2] H. Anton and C. Rorres. Elementary linear algebra. 10. auflage. hoboken, 2010.

[3] M. G. Augasta and T. Kathirvalavakumar. Reverse engineering the neural networks for rule extraction in classification problems. *Neural processing letters*, 35(2):131–150, 2012.

[4] A. Boularias, J. Kober, and J. Peters. Relative entropy inverse reinforcement learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 182–189, 2011.

[5] M. Du, N. Liu, and X. Hu. Techniques for interpretable machine learning. *Communications of the ACM*, 63(1):68–77, 2019.

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

[7] A. Getis and J. K. Ord. The analysis of spatial association by use of distance statistics. In *Perspectives on Spatial Data Analysis*, pages 127–145. Springer, 2010.

[8] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[9] R. Hecht-Nielsen. Theory of the backpropagation neural network. In *Neural networks for perception*, pages 65–93. Elsevier, 1992.

[10] J. Ho and S. Ermon. Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems*, pages 4565–4573, 2016.

[11] A. Kádár, G. Chrupała, and A. Alishahi. Representation of linguistic form and function in recurrent neural networks. *Computational Linguistics*, 43(4):761–780, 2017.

[12] A. Karpathy, J. Johnson, and L. Fei-Fei. Visualizing and understanding recurrent networks. *arXiv preprint arXiv:1506.02078*, 2015.

[13] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel. Handwritten digit recognition with a back-propagation network. In *NeurIPS*, pages 396–404, 1990.

[14] P. Li, S. Bhulai, and J. van Essen. 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*, pages 47–52. IARIA, 2017.

[15] L. Liu, C. Andris, A. Biderman, and C. Ratti. Revealing taxi driver's mobility intelligence through his trace. In *Movement-Aware Applications for Sustainable Mobility: Technologies and Approaches*, pages 105–120. IGI Global, 2010.

[16] N. Liu, M. Du, and X. Hu. Representation interpretation with spatial encoding and multimodal analytics. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pages 60–68, 2019.

[17] A. Nguyen, A. Dosovitskiy, J. Yosinski, T. Brox, and J. Clune. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. In *Advances in Neural Information Processing Systems*, pages 3387–3395, 2016.

[18] M. Pan, W. Huang, Y. Li, X. Zhou, Z. Liu, R. Song, H. Lu, Z. Tian, and J. Luo. Dhpa: Dynamic human preference analytics framework: A case study on taxi driversâĂŹ learning curve analysis. *ACM Trans. Intell. Syst. Technol.*, 11(1), Jan. 2020.

[19] M. Pan, Y. Li, X. Zhou, Z. Liu, R. Song, and J. Luo. Dissecting the learning curve of taxi drivers: A data-driven approach. In *Proceedings of the 2019 SIAM International Conference on Data Mining*. SIAM, 2019.

[20] V. Petsiuk, A. Das, and K. Saenko. Rise: Randomized input sampling for explanation of black-box models. *arXiv preprint arXiv:1806.07421*, 2018.

[21] M. T. Ribeiro, S. Singh, and C. Guestrin. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144. ACM, 2016.

[22] M. T. Ribeiro, S. Singh, and C. Guestrin. Anchors: High-precision model-agnostic explanations. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[23] 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 *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 2329–2334. ACM, 2016.

[24] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[25] R. S. Sutton, A. G. Barto, et al. *Introduction to reinforcement learning*, volume 135. MIT press Cambridge, 1998.

[26] G. Vandewiele, O. Janssens, F. Ongenae, F. De Turck, and S. Van Hoecke. Genesim: genetic extraction of a single interpretable model. *arXiv preprint arXiv:1611.05722*, 2016.

[27] G. Wu, Y. Li, J. Bao, Y. Zheng, J. Ye, and J. Luo. Human-centric urban transit evaluation and planning. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 547–556. IEEE, 2018.

[28] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson. Understanding neural networks through deep visualization. *arXiv preprint arXiv:1506.06579*, 2015.

[29] H. Yuan, Y. Chen, X. Hu, and S. Ji. Interpreting deep models for text analysis via optimization and regularization methods. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5717–5724, 2019.

[30] C. Zeng and N. Oren. Dynamic taxi pricing. *Frontiers in Artificial Intelligence and Applications*, 263:1135–1136, 01 2014.

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

[32] X. Zhou, H. Rong, C. Yang, Q. Zhang, A. V. Khezerlou, H. Zheng, M. Z. Shafiq, and A. X. Liu. Optimizing taxi driver profit efficiency: A spatial network-based markov decision process approach. *IEEE Transactions on Big Data*, 2018.

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

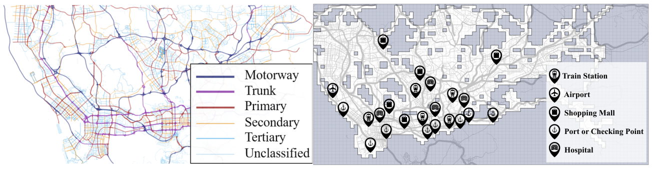
[34] B. D. Ziebart, A. L. Maas, J. A. Bagnell, and A. K. Dey. Maximum entropy inverse reinforcement learning. In *AAAI*, volume 8, pages 1433–1438. Chicago, IL, USA, 2008.

A APPENDIX FOR REPRODUCIBILITY

We have released our code and data[3] to support the reproducibility. Our experiments are running on Red Hat Enterprise Linux 7.2 with a GPU of K40 and CPU of E5-2680. The code released is in Python 3.7.3. The implementation of neural networks is based on PyTorch 1.0.1[4]. We also employ Numpy 1.16.4 and Scipy 1.3.0 in the implementation.

A.1 Details of Data Preprocessing

The road map data includes $21,000$ road segments with six levels as shown in Fig. 11a. We set a bounding box of Shenzhen city from 22.44° to 22.87° in latitude and 113.75° to 114.63° in longitude, and divide the city into $1km \times 1km$ grids. After filtering out those grids that taxis cannot reach, there are 1934 valid grids as shown in Fig. 11b.



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

Figure 11: Shenzhen map data

We collect the information of 23 places of interests (POIs) in Shenzhen as shown in Fig. 11b, including 5 train stations, 1 airport, 5 popular shopping malls, 8 ports and checking points, and 4 major hospitals. Based on these POIs, we calculate the Euclidean distance from the location of each state to the POIs as observation O_5 for each state.

A.2 Detailed Settings of Strategy Learning with GAIL

Settings of the generator network

The generator net consists of 3 convolutional layers and 3 fully-connected layers, and between each 2 consecutive convolutional layers, there is a max pooling layer with filter size of 2×1. We use a kernel size of 3×3 for the convolutional layers with padding of size 1. The output dimensions of the 3 fully-connected layers are 120, 84, 10.

Settings of the discriminator network

In the discriminator network, the input is an input state os size $5 \times 15 \times 15$ and a policy of size 10, before the covolutional layers, we use a fully-connected layer to map the input to the dimension of $5 \times 15 \times 15$. Then following the same 3 convolutional layers, 2 max pooling layers as in the generator network. The output dimensions of the 3 fully-connected layers are 36, 18, 1.

Parameters for training GAIL

During the training process, we apply batch gradient descent approach to update the generator network and discriminator network, with a predefined 200 epochs. We employ ADAM with a learning rate of $2e^{-6}$ to update the parameter of both the generator and the discriminator networks.

A.3 Detailed Settings of Strategy Explanation with xGAIL

Implementation of SpatialAM

When implementing SpatialAM, given the location loc of a state s and a target action a, first, we calculate the distance to POIs based on the location, and put the coordinates of the location and distance to POIs in the first channel of the $5 \times 15 \times 15$ state observation $\boldsymbol{O_s}$. Then, we initialize the rest entries of $\boldsymbol{O_s}$ to 0. For the regularization term, we extract all of the state observations at loc and calculate the mean state observation $\bar{\boldsymbol{O}}_{\boldsymbol{s}}(a)$. In the experiments, we employ $\lambda = 98$ if not specified.

Baselines of SpatialAM

- L_1 **AM**: The objective function of L_1 AM is shown in Eq.(12). In the experiment, we employ $\lambda = 0.05$.

$$\boldsymbol{O}_s^*(a) = \arg\max_{\boldsymbol{O}_s}\{\pi(a|\boldsymbol{O}_s) - \lambda \cdot ||O_s||_1\}, \tag{12}$$

- L_2 **AM**: The objective function of L_2 AM is shown in Eq.(13). In the experiment, we employ $\lambda = 0.05$.

$$\boldsymbol{O}_s^*(a) = \arg\max_{\boldsymbol{O}_s}\{\pi(a|\boldsymbol{O}_s) - \lambda \cdot ||O_s||_2\}, \tag{13}$$

- **AM**: The objective function of AM is shown in Eq.(14), which is simply finding the $\boldsymbol{O}_s^*(a)$ to maximize $\pi(a|\boldsymbol{O}_s)$.

$$\boldsymbol{O}_s^*(a) = \arg\max_{\boldsymbol{O}_s}\{\pi(a|\boldsymbol{O}_s), \tag{14}$$

The SpatialAM and the baselines are trained using Adam optimizer with an initial learning rate of $5e^{-}3$.

Implementation of SpatialRISE

When implementing SpatialRISE, given an input state observation, first, we obtain the map segmentation by using Algorithm 1 with the threshold $G_i^t = 0.8, 0.75, 0.45, 0.5$ for $O_1, O_2, O_3, andO_4$ respectively. For mask generation, we employ the weight of the randomness $\alpha = 0.3$.

Baselines of SpatialRISE:

- **RISE**: The original RISE employs bilinear interpolation to generate masks, there are 2 parameters, i.e., H ($H \geq 15$), the side length of the map after interpolation, and h ($h \leq H$), the side length of the seed mask. RISE first randomly assign 0 or 1 to each entry of the seed map, then, bilinear interpolation is used to extend the seed mask to the mask with a side length of H, finally, the mask with a side length of 15 is cropped from the mask with a side length of H with random indents. Here, we employ $H = 17$ and $h = 7$ in the experiments.
- **PixelRISE**: In each entry of the mask, the value of 0 or 1 is assigned randomly and independently.
- **SolidRISE**: First, SolidRISE employs the same map segmentation algorithm to obtain the clusters, then, it randomly assign 0's or 1's to all entries inside each cluster. The settings of the parameters for map segmentation is the same as that in SpatialRISE.

In SpatialRISE and the baselines, we generate $N = 3000$ masks, and calculate the weighted average of the masks to obtain the importance map according to eq.(11).

[3] https://github.com/paperpublicsource/xgail
[4] https://pytorch.org/get-started/previous-versions/