GRL4AOI: Graph-based Reinforement Learning for Service-aware AOI Segmentation

Haomin Wen[4], Youfang Lin[1,3], Jinji Fu[1], Xiaowei Mao[1,2,3], Yan Lin [1,3], Yuting Qiang[2], Lixia Wu[2], Huaiyu Wan[1,3,*]
[1]School of Computer and Information Technology, Beijing Jiaotong University [2]Cainiao Network
[3]Beijing Key Laboratory of Traffic Data Analysis and Mining, Beijing, China
[4]Carnegie Mellon University (*Corresponding author)
wenhaomin.whm@gmail.com;{yflin,fujinji,maoxiaowei,ylincs,hywan}@bjtu.edu.cn;
{xiushui.qyt,wallace.wulx}@cainiao.com;

ABSTRACT

AOI segmentation is a prerequisite task in location-based services such as food delivery. It aims to divide the urban geographical space into several non-overlapping regions, also termed Areas of Interest (AOIs). Previous efforts typically resort to optimization methods to meet specific service-semantic goals (e.g., workload equality). Though promising, optimization-based methods lack: i) the ability to incorporate various spatial and temporal features, and ii) the generalizability to accommodate different service-semantic goals. Targeting the above limitation, we present the first attempt to generalize Graph-based Deep Reinforcement Learning (GRL) for AOI segmentation. Specifically, the entire urban space is modeled as a graph, where each node represents one atom region and the edge denotes spatial connectivity between atom regions. We begin by highlighting that the AOI segmentation problem can be naturally formulated as a sequential decision problem on the graph, which adjusts one node (i.e., atom region) along AOI's border at each decision step. Based on the above understanding, we leverage a Markov Decision Process (MDP) to model the sequential decision process, leading to a novel graph-based deep reinforcement framework called GRL4AOI. It effectively ingests various region-related features (such as region size, order number, trajectories) with deep learning capabilities. Furthermore, it models various service-semantic objectives in a flexible manner by treating them as rewards to guide the learning process. Based on the framework, we implement a model equipped with Double-DQN for AOI segmentation in the logistics service, with two service-semantic goals: i) trajectory modularity and ii) predictability. Extensive offline experiments and the online deployment demonstrate the effectiveness of the proposed framework.

1 INTRODUCTION

The rapid development of location-based services (LBS), such as food delivery [29], logistics [23], ride-sharing [25], and spatial crowdsourcing [27, 28], has significantly enhanced people's daily lives. In such LBS platforms, a crucial task is AOI segmentation, which aims to partition the given urban geographical space into multiple non-overlapping regions, known as Areas of Interest (AOIs). AOI segmentation serves as a foundation for the platform as numerous subsequent tasks rely on the segmentation results as a necessary input. To illustrate, leading ride-sharing companies such as DiDi and Uber divide the city into multiple AOIs to efficiently allocate available drivers to areas with high demand [8, 14]. Similarly, logistics platforms like Cainiao and JD.COM adopt AOI segmentation to assign orders to couriers [15, 20, 24] in the last-mile delivery. Given these instances, there is a growing demand for effective methods to generate a well-organized set of AOIs.

Table 1: Comparison between our model and related ones.

Method	Geo-aware	Service-aware	Abundant feature	Generalizability
Fixed-shape	✘	✘	✘	✘
Road-network	✔	✘	✘	✘
Optimization-based	✔	✔	✘	✘
GRL4AOI (ours)	✔	✔	✔	✔

A direct way is to divide the given urban space into several fixed-shape grids or hexagons [8, 11, 13, 26]. In contrast, road-network methods [12, 19] utilize road networks as AOI boundaries, which can naturally capture the geo-semantics (i.e., identifying geographical entities such as communities and schools). Recently, we have witnessed a bloom of optimization-based methods being proposed, aiming to achieve better AOI segmentation defined by certain service-semantic goals, i.e., operational requirements generated by downstream tasks in LBS platforms. As an example, E-partition [10] focuses on achieving workload equity by segmenting AOIs, while RegionGen [3] aims to enhance demand prediction accuracy through optimizing AOI segmentation. Though promising, these optimization-based algorithms face the following two limitations that restrict their performance in real-world applications (the comparison of those methods and ours is shown in Table 1):

1) Lack of ability to effectively incorporate adequate spatial and temporal features into the optimization process. Urban regions are usually associated with various features, such as their size, type, orders located in that region, and trajectories in the region. Technically speaking, a well comprehension of those information is beneficial for a more optimized AOI segmentation. However, previous methods only take one or limited features into the optimization process, lacking the investigation on modeling different types of features. 2) Lack of generalizability to accommodate different

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-1-4503-9385-0/22/08
https://doi.org/10.1145/3534678.3539084

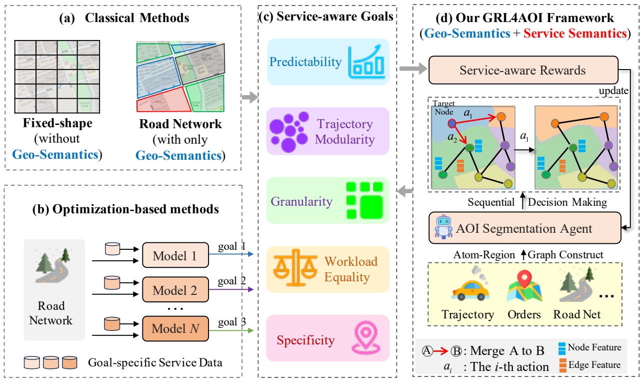

Figure 1: Illustration of different frameworks. Compared with previous works, we present the first attempt to leverage the power of DRL for AOI segmentation.

service-semantic goals. The majority of optimization-based methods are designed to meet only one service goal, as it requires great effort to construct the optimization problem and the constraints with respect to the service goal. However, in reality, there can be many different services (e.g., order dispatching, route prediction) rely on the AOI results. Therefore, there is a high demand for an AOI segmentation framework which could accommodate different downstream tasks.

Targeting the above limitation, we generalize deep reinforcement learning for AOI segmentation, leading to a novel framework called GRL4AOI. Specifically, as shown in Figure 1, we first utilize road segmentation methods [3] to extract geographically semantically meaningful atomic regions. Based on this, a graph is constructed to better represent the geographical space, where nodes are atomic regions and edges represent spatial proximity between them. Then, we show that the AOI segmentation problem can be naturally seen as a sequential decision making problem on the graph, where each decision step adjusts one node (i.e., atom-region) along AOI's border. In this case, we are able to formulate the above process as a Markov Decision Process (MDP) and bring in deep reinforcement learning approaches to solve the problem. Doing so introduces the following two metrics: i) the DRL model can take full advantage of various region-related features (such as region size, order number, workload) with deep learning capabilities (targeting limitation 1); ii) Different service-semantic goals can be introduced as rewards in a flexible manner (targeting limitation 2).

Furthermore, we implement a model (named GRL4AOI-L) based on the framework to solve AOI segmentation for last-mile delivery service in logistics. It introduces a Double Deep Q-learning Network (DDQN) to gradually optimize the AOI generation with two service-semantic goals: i) trajectory modularity, i.e., maximize tightness of the trajectory connections within an AOI and the sparsity of connections between AOIs; ii) predictability [3], i.e., maximize the accuracy of demand prediction in generated AOIs. Overall, the contribution of this work is summarized as follows:

- By formulating the AOI segmentation as a sequential decision problem in the graph, we propose the first-ever graph-based DRL framework GRL4AOI. It can achieve both geo-semantic and service-semantic goals in a flexible way while taking full advantage of various features.
- Based on the framework, we propose GRL4AOI-L for AOI segmentation in logistics service. It adopts a value-based model Double-DQN to improve the trajectory modularity and predictability for service in logistics.
- We conduct extensive offline experiments as well as the online deployment, the results demonstrate the superiority of our method over other solutions.

2 PRELIMINARIES

Without loss of generality, we provide a service-agnostic formulation for service-aware AOI segmentation, since different services can have various inputs and service-semantic goals.

Definition 1: Geographic Data $\mathcal{G}$, contains graphical information of the target urban space, e.g., the road network or satellite images. Geographic data is the basic requirement for achieving segmentation with geographic semantics.

Definition 2: Service Data $\mathcal{S}$, defined as data generated during the operation process, such as orders in online food delivery systems, or courier trajectory data in logistics platforms.

Definition 3: Service-semantic Goals, which refers to operation requirements/objectives generated in the service platform, such as predictability and workload equity. We use $O = \{o_1, \ldots, o_k\}$ to denote the service-semantic goals, each o_i is a predefined goal.

Definition 4: Service-aware AOI Segmentation. Given the geographic data $\mathcal{G}$ and service data $\mathcal{S}$, service-aware AOI segmentation learns a mapping function $\mathcal{F}$ that divides the target urban space into several non-overlapping AOIs, which aims to achieve the service-semantic goals O, formulated as:

$$\mathcal{F}_O(\mathcal{G}, \mathcal{S}) \to \mathcal{A} := \{a_1, \ldots, a_n\}, \tag{1}$$

where $\mathcal{A}$ is an segmentation result with n AOIs, and a_i means the i-th AOI. Each AOI is a region in the geographical space.

3 PROPOSED FRAMEWORK: GRL4AOI

As shown in Figure 2, GRL4AOI consists of two steps: Atom-Region Graph (AR-Graph) construction based on geographical data and service data; 2) RL-based AOI segmentation, which gradually adjusts the node in the atom-region graph by the agent under the guidance of the rewards designed by service-semantic goals. Specifically,

1) *Atom-Region Graph Construction* first utilizes the fine-grained road network to divide the urban space into non-overlapping regions, called atom-region. Then, a graph $G = (\mathcal{V}, \mathcal{E}, \mathbf{X}, \mathbf{E})$ is constructed, with each node corresponding to an atom-region. $\mathcal{E}_t = \{(i,j) \mid v_i, v_j \in \mathcal{V}_t\}$ is the set of edges. $\mathbf{X}_t \in \mathbb{R}^{n \times d_v}$ and $\mathbf{E}_t \in \mathbb{R}^{n \times n \times d_e}$ are the node and edge features respectively, where d_v and d_e are the node feature dimension and edge feature dimension, respectively. Various service data can be incorporated into the graph as node/edge features. For example, using the number of orders generated in the atom-region as the node feature.

2) *RL-based AOI segmentation* introduces a novel perspective to show that AOI segmentation can be regarded as a sequential decision-making problem. As illustrated in Figure 3, the core idea is deciding which neighbor AOI it belongs to for a node (called node merging decision or AOI selection) located at AOI's border at each

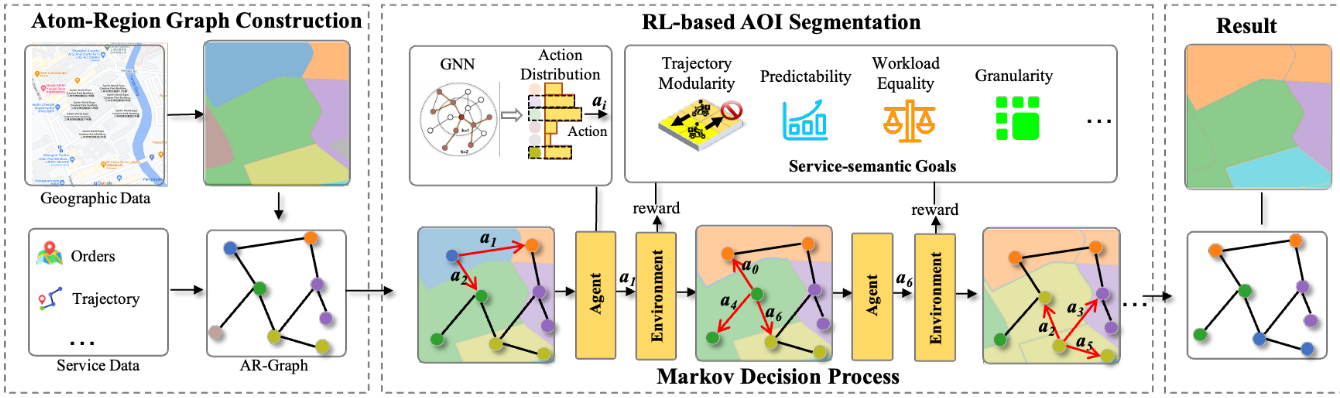

Figure 2: GRL4AOI Architecture, mainly includes two steps: 1) atom-region graph construction, based on geographical data and service data. 2) RL-based AOI segmentation, which gradually adjusts the node's AOI in the atom-region graph by the agent under the guidance of the rewards designed by service-semantic goals.

decision step, via a trained AOI segmentation agent. After each action is performed, the environment will provide reward feedback that is aligned with service-semantic goals. Guided by the rewards, the agent updates its parameters and gradually adjusts the node in the AR-Graph to achieve better segmentation results.

Technically, the process of making sequential decisions can be represented by a finite-horizon discounted Markov Decision Process (MDP) [21]. In this process, an AOI segmentation agent interacts with the environment over T discrete time steps by AOI selection. Formally, a MDP can be formulated as $M = (S, A, P, R, \gamma)$, where S is the set of states, A is the set of actions, $P : S \times A \times S \rightarrow \mathbb{R}_+$ represents the transition probability, and $R : S \times A \rightarrow \mathbb{R}_+$ represents the reward function. The initial state distribution is $s_0 : S \rightarrow \mathbb{R}_+$, and $\gamma \in [0, 1]$ is a discount factor to control the trade-offs between the importance of immediate and future rewards. When provided with a state s_t at a given step t, the AOI segmentation agent utilizes the current policy π_θ (a deep neural network parameterized by θ) to generate an action, i.e., selecting a nearby AOI for the node. The agent then receives a reward r_t defined by the service-semantic goals from the environment. During the training process, the objective is learning the best parameter θ^* for the agent to maximize the expected cumulative reward, formulated as:

$$\theta^* = \arg\max_\theta \mathbb{E}_{\pi_\theta}\left[\sum_{t=1}^{T} \gamma^t r_t\right], \tag{2}$$

where T is the total time step related to the node number in the AR-Graph. In the proposed framework, the agent, state, action, reward, and state transition probability are defined as follows:

AOI Segmentation Agent: It is responsible for learning the target function $\mathcal{F}_O$. It selects an AOI for each node with the learned policy (shown in Figure 3). The agent follows an encoder-decoder architecture, where the encoder embeds features from the current state s into hidden representations. The decoder produces an action a at each time step, based on embeddings from the encoder. Abstractly,

$$a_t = \text{Decoder}(\text{Encoder}(s_t)), \tag{3}$$

State: A state s_t is composed of the information of current AOI segmentation, the whole AR-Graph, and the target node. All possible segmentations make up the entire state space.

Action: As shown in Figure 3, an action a_t indicates which neighbor node should be merged to for target AOI. We also add an action named "stay", which means that there is no suitable neighbor to merge, the target node remains its original AOI. Therefore, the action space at each step is made up of all neighbors of the target node and the target node itself. Moreover, it is worth mentioning that by setting the action as adjusting the node's neighborhoods, the spatial continuity constraint that each AOI should have interconnected nodes can be naturally satisfied. Moreover, the maximum area constraint is set to avoid oversized regions, which means if a region exceeds the area constraint after a merge action, then this action will be changed to "stay".

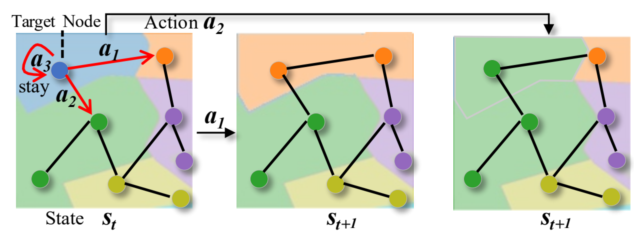

Figure 3: Illustration of Actions.

State Transition Probability: $P(s_{t+1}|s_t, a_t)$ represents the transition probability from state s_t to s_{t+1} if action a_t is taken at s_t, which means that the state transfer from one segmentation from another one. In the AOI segmentation problem, the environment is deterministic with no uncertainty, i.e., the state s_{t+1} transited from state s_t after taking action a_t is deterministic.

Service-semantic Reward: $r_t \in \mathbb{R}$: A well-designed reward can have a significant impact on the learning performance of the agent and ultimately determine the quality of the model. In AOI segmentation, certain service-specific semantic goals need to be achieved after the segmentation. Those goals can be easily converted into rewards in the RL framework, which improves the model's flexibility to accommodate various goals in different services. In Section 4, we will introduce how to design the rewards by taking logistics service as an example.

4 PROPOSED MODEL FOR LOGISTICS SERVICE: GRL4AOI-L

Based on the above framework, we further propose a model GRL4AOI-L for AOI segmentation in logistic platforms (e.g., Cainiao Network[1]), where the order dispatching and logistics management heavily rely on AOIs in its last-mile package pick-up/delivery service. Meanwhile, massive service data has been collected during the operation process, including package information and courier's trajectories. Given the logistic-related service data and geographic data, the task of AOI segmentation in logistics is generating a set of AOIs that can meet the two service-semantic goals: i) trajectory modularity and ii) predictability (detailed in Section 4.2).

4.1 Atom-Region Graph Construction

Figure 4 shows the process of AR-Graph construction by leveraging the road network, trajectory, and parcel data. We first utilize the image-based segmentation method [3] to identify atom-regions by considering fine-grained road networks as region boundaries, with two operations: connected component labeling (CCL), and thinning. CCL finds the connected component separated by roads, and thinning eliminates roads between regions. Each atom-region is a node in the graph. In the adjacency matrix $\mathbf{A}$ (equivalent to the edge set), $\mathbf{A}_{i,j} = 1$ if region i and region j are neighbors in geographical space. A detailed process is given in Appendix A.1.

Node & Edge Features: The AR-Graph is associated with abundant spatial-temporal features, where each node (i.e., atom-region) feature $\boldsymbol{x}_i$ includes: i) number of parcels in the atom-region; ii) Initial AOI assignment by coarse-grained roads-network segmentation; iii) the size of the atom-region. For the edge feature $\boldsymbol{e}_i$, we calculate the number of trajectory transitions between two adjacent atom-regions, allowing the model better to capture the trajectory transfer patterns between different regions.

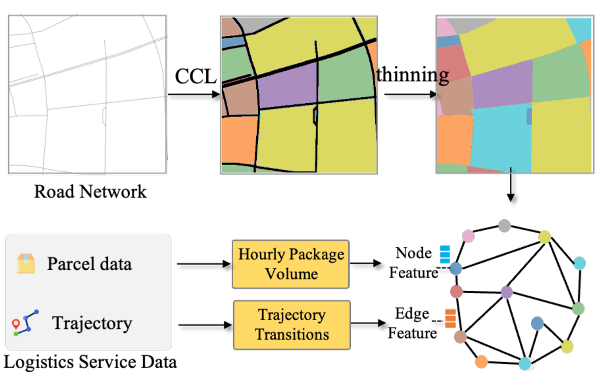

Figure 4: Atom-region Graph Construction.

4.2 Double-DQN for AR-Graph Segmentation

Following the proposed framework, we further implement a reinforcement learning model with Double-DQN to solve the problem.
State. The state records the target node and its neighbors, the current AOI segmentation, and the AR-Graph. Specifically,

- The target node v_t^i is the node i to be modified at decision step t, which is represented by its features $\boldsymbol{x}^i \in \mathbb{R}^{d_v}$.
- Neighbors of the target node $\mathcal{N}_t^i = \{\boldsymbol{x}_j | \mathrm{A}_{i,j} = 1\}$. Understanding the neighbors of a node is crucial for its decision on AOI selection.
- Current AOI segmentation $\mathcal{A}_t = \{a_1, \ldots, a_{n_t}\}$, where a_i means the i-th AOI, and n_t is the number of AOI at current step. $\mathcal{A}_t$ is basic information for the agent to understand the whole distribution of current AOI segmentation.
- AR-Graph $G = (\mathcal{V}, \mathcal{E}, \mathbf{X}, \mathbf{E})$ as introduced above, which contains abundant features for describing each node and the correlation between nodes.

Service-aware Reward. Our framework can handle different objective functions and optimize the segmentation by converting them as rewards. For logistics services, we set two goals in the reward function, trajectory modularity o_1 and predictability o_2:

Trajectory modularity. In logistics service, couriers usually finish all tasks in one AOI and then head to another one. To this end, we hope that within each AOI, there are dense trajectory connections; while between different regions, trajectory transitions are as weak as possible. Inspired by the concept of modularity in community detection [4], we utilize "trajectory modularity" to name this service goal. It is calculated by the proportion of trajectories within the same AOI in total trajectories:

$$o_1 = \frac{\sum_\tau \sum_a N_{a,\tau}}{\sum_\tau N_\tau}, \tag{4}$$

where $N_{a,\tau}$ is the number of the trajectory τ in AOI a and N_τ is the total number of the trajectories.

Predictability. Predictability refers to whether future events of interest (i.e., the package delivery demand in this paper) can be easily predicted. One important service-semantic goal is to achieve better predictability of the demand time series by appropriate AOI segmentation. Intuitively, predictability should be tested by evaluating a given prediction model (usually deep models) using metrics such as MSE and RMSE. However, this model-dependent way requires retraining the prediction model each time the AOI segmentation changes, which is notoriously time-consuming and practically intractable. To tackle the challenge, we utilize the Auto-Correlation Function (ACF) to represent the predictability of an AOI's time series following RegionGen [3], which shows that a better ACF is usually associated with better predictability. Formally, let the time series of the i-th AOI be $s_{i,1}, s_{i,2}, \ldots, s_{i,M}$, where $s_{i,m}$ means the number of packages generated at the i-th region in m-th hour. The ACF of region i after k slots delay is formulated as:

$$\rho_i^k = \frac{M \cdot \sum_{m=k+1}^{M} (s_{i,m} - \bar{s}_i)(s_{i,m-k} - \bar{s}_i)}{(M-k) \cdot \sum_{m=1}^{M} (s_{i,m} - \bar{s}_i)^2}, \tag{5}$$

where $\bar{s}_i$ is the mean value of the series in AOI i, and k is the delay of the series. Here we evaluate the predictability by setting k to 24 to calculate the auto-correlation of 24 hours (i.e., one day) delay:

$$o_2 = \sum_{i \in A} \rho_i^{24} / |A|. \tag{6}$$

ACF is utilized as a fast proxy measurement for predictability, which does not require re-train the model after the change of segmentation results. In summary, the overall objective function is:

$$R_t = k_1 o_1 + k_2 o_2, \tag{7}$$

[1] https://www.cainiao.com/

where k_1 and k_2 are coefficients to balance the scale of the two objectives. And we use the difference between the objective value of two states as the reward for each step, it can be formulated as:

$$r_t = R_{t+1} - R_t. \tag{8}$$

GNN-based Agent. To fully model the spatial correlation between different atom-regions, we employ a Graph Neural Network (i.e., GCN) to encode the AR-Graph, then utilize the attention mechanism to conduct AOI selection for the target node.

GCN for Capturing the Spatial Correlation. Given the node and edge features, we first project them into a hidden representation by a linear projection, resulting in the node embedding $\boldsymbol{h}_i \in \mathbb{R}^{d_h}$, and the edge embedding $\boldsymbol{z}_i \in \mathbb{R}^{d_z}$, respectively. Then, we perform graph convolution for L times, updating the node and edge embeddings by leveraging their interactions. We use one-hop neighbor messages in the information updating process, which can be formulated as:

$$\begin{aligned} \boldsymbol{h}_i^{l+1} &= f(\boldsymbol{h}_i^l, \mathrm{Agg}\{\boldsymbol{h}_j^l, \boldsymbol{z}_{ij}^l : j \in \boldsymbol{N}_i\}) \\ \boldsymbol{z}_{ij}^{l+1} &= g(\boldsymbol{z}_{ij}^l, \mathrm{Agg}\{\boldsymbol{h}_i^l, \boldsymbol{h}_j^l\}), \end{aligned} \tag{9}$$

where $\mathrm{Agg}(\cdot)$ is the aggregation function, and the updating function f, g is composed by a non-linear transformation, defined as follows:

$$\boldsymbol{h}_i^{l+1} = \boldsymbol{h}_i^l + \sigma((\mathbf{W}_1^l \boldsymbol{h}_i^l + \sum\nolimits_{j \in \mathcal{N}_i} \eta_{ij}^l \odot \mathbf{W}_2^l \boldsymbol{h}_j^l)) \tag{10}$$

$$\boldsymbol{z}_{ij}^{l+1} = \boldsymbol{z}_{ij}^l + \sigma(\mathbf{W}_3^l \boldsymbol{z}_{ij}^l + \mathbf{W}_4^l \boldsymbol{h}_i^l + \mathbf{W}_5^l \boldsymbol{h}_j^l), \tag{11}$$

where $\mathbf{W}_i^l \in \mathbb{R}^{d_h \times d_h}$ $(i = 1, \ldots, 5)$ are trainable parameters, σ is ReLU activation function. And $\eta_{ij}^l = \sigma(\mathbf{W}_6^l \mathbf{z}_{ij}^l) / \sum_{j' \in \mathcal{N}_i} \sigma(\mathbf{W}_6^l \mathbf{z}_{ij'}^l)$, where $\mathbf{W}_6^l \in \mathbb{R}^{d_h \times 1}$ is a trainable parameter. After L layer of graph convolution, we could get the node and edge representation $\mathbf{H} \in \mathbb{R}^{n \times d_h}$, $\mathbf{Z} \in \mathbb{R}^{n \times n \times d_z}$.

Attention mechanism for neighbor selection. Based on the node and edge embeddings, the attention mechanism is utilized to select one node from the target's neighbors, formulated as:

$$\mathbf{H}^{\mathrm{att}} = \mathrm{softmax}(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_h}})\mathbf{V} \tag{12}$$

where $\mathbf{Q}$ is the query from the embedding of target node $\boldsymbol{h}_i$, $\mathbf{K}$, $\mathbf{V}$ is the node embeddings $\mathbf{H}$. Then, a feedforward layer is added to project $\mathbf{H}^{\mathrm{att}}$ into output scores, and we mask non-neighbor nodes by setting their scores to $-inf$. At last, we apply softmax to the score, which serves as the Q value in the RL framework:

$$\mathbf{Q} = \mathrm{softmax}(\mathrm{mask}(\mathbf{W}_7 \mathbf{H}^{\mathrm{att}})), \tag{13}$$

where $\mathbf{W}_7 \in \mathbb{R}^{d_h \times d_h}$ is a trainable parameter.

Overall, at each step, the agent encodes the state with a graph convolutional network. Via convolutional operations, the GCN state encoder extracts an effective representation of the target node's neighborhood information. Then the attention mechanism computes the influence scores for the target node, which indicates the performance score of the action in the learned policy. At last, the AOI-selection policy network selects one of its neighbors based on the attention score.

DDQN-based RL Training. To learn the agent's policy network, we adopt Double Deep Q-Networks (DDQNs) [22] for the reinforcement training procedure because of its strong performance as proved in different environments such as games [9, 16, 18].

Specifically, DDQN is an improvement of DQN [16, 17], whose primary idea is utilizing a neural network to approximate the Q-value function, which reflects how good it is for an AOI selection action at a certain state. The network takes a state s_t as input and produces Q-values for all possible actions. The Q-value is learned iteratively through the Bellman equation:

$$Q(s_t, a_t) = r_t + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}). \tag{14}$$

The optimal Q-value of an action is the sum of the current reward and the maximum Q-values of all actions in the next state. However, DQNs tend to overestimate Q-values because they use the same network to select the best action and evaluate it. This can lead to suboptimal policies and reduced stability in training.

Double DQNs aim to address this overestimation bias. Instead of using a single network for both action selection and evaluation, Double DQNs use two networks: i) An online network to select the best action; ii) A target network to evaluate the Q-value of the selected action. Thus, the updated Q-value in DDQNs is:

$$Q(s_t, a_t) = r_t + \gamma Q_{\mathrm{target}}(s_{t+1}, \mathrm{argmax}_{a_{t+1}} Q_{\mathrm{online}}(s_{t+1}, a_{t+1})). \tag{15}$$

Here Q_{online} is the Q-value estimated by the online network, and Q_{target} is the Q-value estimated by the target network. Compared with DQN, the optimal action value Q is determined by the immediate reward, and the maximum Q-value from the target network in the next state. By decoupling the action selection from its evaluation, DDQNs reduce the overestimation bias, resulting in more stable and efficient learning.

At last, since over-small atom-regions are meaningless in logistics management, we add the minimal area constraints to generate more practical AOIs. Specifically, we merge the AOI (which area is smaller than the minimal area) to its neighbor who can bring the best gain in total reward.

5 EXPERIMENTS

5.1 Experiment settings.

5.1.1 Dataset. The experiments are conducted based on the delivery data provided by one of China's largest logistics companies, which includes the delivery data from Shanghai, China. The dataset comprises both order information and trajectory data. We collected data from Jun. 2019 to Aug. 2019, resulting in nearly 260k orders and 57.04 million trajectory points. The trajectory is sampled at intervals of 2-4 seconds, with each point represented by its latitude, longitude and timestep.

5.1.2 Baselines. We choose several classic algorithms as well as state-of-the-art models as baselines.

- Grid, which segments the target urban space by fixed-shape grids. The area of each grid is $1km^2$.
- RoadNetwork [12], which is a common practice for AOI segmentation that inferences AOIs by road networks.
- DBSCAN [6]. A classical method that makes the AOI segmentation by clustering all packages, with each cluster transformed into an AOI.
- Louvain [2], which converts the AOI segmentation into the community detection problem [2, 7] on the graph, and applies the Louvain to segment AOI.

Table 2: Experiment Results.

Method	Region A			Region B			Region C		
	o_{total}	Traj-Modularity	Predictablity	o_{total}	Traj-Modularity	Predictablity	o_{total}	Traj-Modularity	Predictablity
RoadNetwork	0.29	0.35	0.19	0.30	0.39	0.17	0.25	0.34	0.11
DBSCAN	0.31	0.18	0.50	0.26	0.18	0.38	0.23	0.16	**0.34**
Louvain	0.46	0.56	0.32	0.48	0.66	0.21	0.48	**0.71**	0.14
GCLP	0.49	0.51	0.46	0.40	0.42	0.36	0.42	0.51	0.28
RegionGen	0.64	**0.57**	0.73	0.32	0.22	0.47	0.29	0.37	0.18
GRL4AOI-L (ours)	**0.68**	**0.57**	**0.84**	**0.70**	**0.70**	**0.68**	**0.52**	0.66	0.32

- GCLP [4], which also formulates the problem as a community detection problem and employs the label propagation method for clustering.
- RegionGen [3] models the AOI segmentation as a multi-objective optimization problem, by setting predictability and service specificity as goals.

5.1.3 Metrics. We use the ACF for predictability, trajectory modularity (short as Traj-modularity), and their combined value o_{total} to evaluate the performance of different models.

5.1.4 Implementation. The offline experiments are conducted on a machine with a Hygon C86 7151 CPU and NVIDIA RTX A4000 GPUs. We use the RMSprop optimizer for the parameter update. The initial learning rate is set to 10^{-4} and decays in each episode. The training episode is set to 1000. The random seed is set to 1. The number of the GCN layer is set to 2, and its output dim is 16. The max area constraint is set to $2kmm^2$. The hyper-parameters k_1, k_2 in the total reward function are set to 0.6, 0.4, respectively.

5.2 Results.

To make a comprehensive comparison under different environments, we choose different regions in Shanghai. Table 2 presents the comparison of our model with different baselines, which shows that GRL4AOI-L significantly outperforms other methods. GRL4AOI-L outperforms the most competitive baseline by $6.25\% - 44.52\%$ in terms of total reward.

RoadNetwork only considers coarse-grained road networks for AOI segmentation, without taking service semantics into account. To this end, RoadNetwork achieves a bad performance in terms of the service-semantic goals, especially only achieving 0.19 in ACF for predictability. DBSCAN makes the AOI segmentation by clustering the packages, the resulting AOI can well perceive the package density, thus achieving a promising performance in ACF score (since a large number of packages usually means better predictability). However, AOI with high package density does not necessarily indicate fewer trajectory transitions between different AOIs. For example, there may be obstacles like rivers that separate those parcels. This leads to its interior performance in Traj-modularity compared with RoadNetwork.

Louvain transforms the AOI segmentation into a graph cluster problem, aiming to group tightly connected nodes (defined by trajectory transitions) together. As a result, it gets a competitive performance of 0.56 in trajectory modularity. In contrast, GCLP divides regions with distance restrictions which can reduce the trajectory modularity and avoid generating over-large regions. Both methods cannot take other service-semantic goals into account, resulting in suboptimal performance in ACF. Especially for Louvain, it only gets 0.32 in ACF.

RegionGen utilizes the optimization method that combines value functions to maintain a Pareto-optimal [5] solution set. It achieves the most competitive results compared to our proposed method, especially achieving 0.73 in ACF at Region A. In comparison, GRL4AOI-L improves the trajectory modularity and data predictability by casting them into the reinforcement learning framework, which serves as the reward to guide the model training. In this way, GRL4AOI-L provides a more effective and flexible way to model abundant region features and service-semantic goals compared to RegionGen, thus achieving optimal results considering the trajectory rewards and predictability rewards.

5.3 Component Analysis.

We conduct the ablation study to investigate the impact of different components of GRL4AOI-L. Figure 5 illustrates the results.

Firstly, we remove the GCN encoder. Without GCN (w/o GCN), the total reward decreased by 3.66%. This indicates that the GCN effectively integrates information from AR-Graph and plays a role in the model's decision-making process. Secondly, we remove the Attention (w/o Attention) module. As a result, the reward decreases by 2.09%. The Attention module helps select important nodes for the current AOI selection more effectively, thereby enhancing the model's performance. When both GCN and attention are removed, leaving only MLP (w/o GCN & Attention), the model's performance is the poorest.

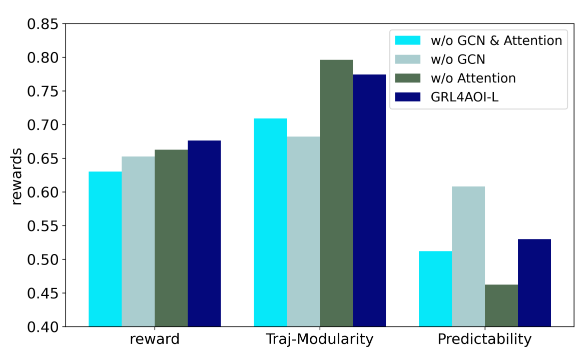

Figure 5: Component Analysis.

5.4 Parameter Effectiveness.

Minimal Area. We investigate the impact of the minimum area constraint in Region A. Figure 6 illustrates the rewards of different area districts. As the minimum area constraint increases, we observe an increase in trajectory modularity, predictability, and total reward. This can be attributed to the fact that smaller areas typically contain less data, and merging them with larger neighboring regions often results in improved rewards.

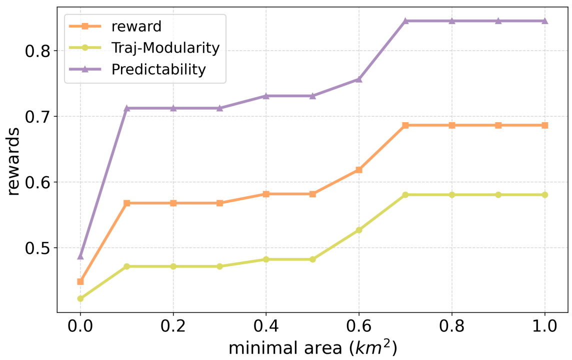

Figure 6: Influence of minimal area constraints.

AR-Graph Granularity. We also investigate the impact of different granularities in atom-region graph construction. Specifically, we obtain 3 graphs (fine-grained & medium-grained & coarse-grained) with different granularity by selecting different levels of the road network(e.g. primary and secondary routes). Table 3 displays the model's performance in different granularity. The coarse-grained model performs worst in all metrics, because the solution space is significantly reduced in the coarse-grained AR-Graph, thus cannot achieve the optimal solution. In contrast, fine-grained data leads to overall small atom-regions which do not have abundant data, thus resulting in poor performance in predictability.

Table 3: Influence of Different AR-Graph Granularities.

Granularity	o_{total}	Traj-Modularity	Predictability
Fine (107 nodes)	0.67	0.63	0.75
Medium (79 nodes)	0.68	0.57	0.84
Coarse (53 nodes)	0.50	0.34	0.75

5.5 Case Study

We visualize the AOI segmentation results in one region of Shanghai to give an intuitive comparison between different methods. Figure 8 shows the results with each color representing a different AOI. Figure 8(a) shows the geographical information of the target region. Figure 8(b) and (c) show the distribution of trajectories and packages respectively, where the red color indicates high density. In Figure 8(d), Grid generates AOI with the fixed shape, resulting in a poor performance in geographical semantics. In Figure 8(e), Road-Network produces AOI segmentation results with good spatial semantic meaning. However, the fine-grained road network will generate overall small regions that are unnecessary for logistics management and lack service semantic meaning. In Figure 8(f), DBSCAN makes AOI segmentation by clustering packages, where atom-regions with high package density are aggregated into one AOI. However, DBSCAN cannot introduce the area size constraints, thus can easily result in oversized AOI. GCLP and Louvaion are illustrated in Figture 8(g) and Figure 8(h), respectively. They both tend to aggregate atom-regions with high trajectory density into one AOI, and thus can perform well in the metrics of trajectory modularity. In Figure 8(i), RegionGen is designed to improve the predictability of AOIs, therefore, a large portion of its generated AOI has a high ACF. However, regions with dense trajectory transitions are separated since it cannot take the trajectory distribution into account. In Figure 8(j), the proposed GRL4AOI-L generates AOI segmentation results considering both package distribution and trajectory distribution, leading to a promising performance in trajectory modularity and predictability.

5.6 System Design

Our GRL4AOI framework operates in the offline mode primarily for the task of AOI segmentation, leveraging the comprehensive road network, realistic historical courier trajectories, as well as dispatched orders. It updates AOIs in T+1 mode, which means that the updates occur on the subsequent day to the current one (T) when the courier data is collected. This systematic update ensures that the AOIs remain relevant and are a true reflection of the evolving logistics landscape.

Upon updating, the AOIs crafted by the offline phase of our GRL4AOI framework are then synchronized with PolarDB, a real-time online database. This integration facilitates the seamless utilization of AOIs in real-time logistic services. In the online environment, various applications harness these AOIs for critical operational tasks such as order dispatch, which involves dynamically assigning orders to couriers; route prediction, which forecasts the most efficient paths; and arrival time estimation, ensuring customers receive accurate predictions of delivery times. As couriers go about their deliveries, their activities are captured, which are fed back into our system. These real-time data points are stored in a MySQL database, providing a wealth of information that continually refines and enhances the AOI segmentation process. The bidirectional flow of information between the offline AOI generation and the online

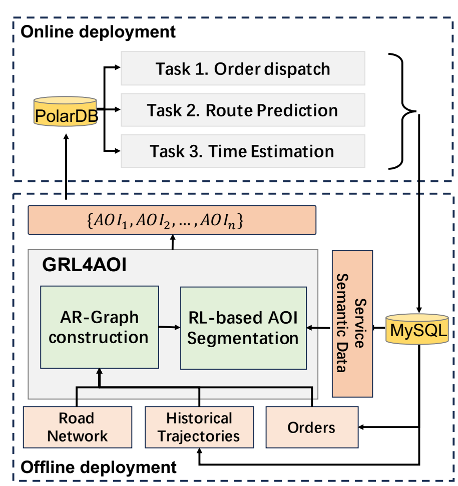

Figure 7: System deployment for GRL4AOI system

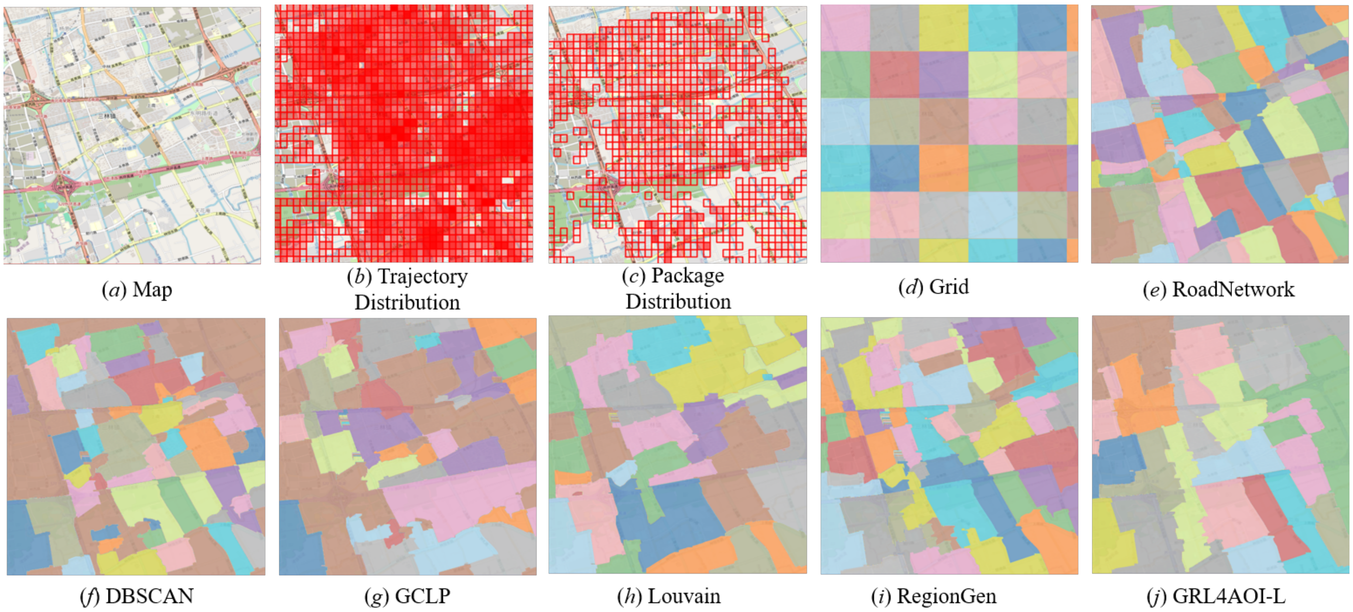

Figure 8: Case Study, which shows the AOI segmentation results of different methods.

application of these AOIs creates a dynamic, self-improving system. Figure 7 illustrates the system design.

6 RELATED WORK

Current AOI segmentation methods can be broadly classified into three classes: 1) fixed-shape, 2) road-network based and 3) optimization-based methods.

Fixed-shape & Road-network methods. Fixed-shape methods divide the urban space into several fixed-shape regions, such as grids or hexagons [8, 11, 13, 26]. Those methods cannot capture the geographic semantic meaning, which may significantly trim down the performance of other AOI-based tasks in the platform. Road-network based methods [12, 19], which utilize road-networks as boundary for AOIs, can well capture the geographic semantics of the urban space. Though effective, a fundamental limitation is the lack of service semantic information [3, 10], which is usually a bottleneck that restricts the overall service performance.

Optimization-based methods. To address the above limitation, optimization-based methods introduce service-specific goals or constraints (i.e., service semantics) to guide the AOI generation based on optimization-based methods by leveraging various kinds of data generated in the service. RegionGen [3] approaches the problem as a multi-objective optimization task. The city is initially divided into smaller spatial units using road networks. These units are then grouped together into distinct AOIs by maximizing different objectives, including the average predictability and service specificity of the clusters. [1] introduces POI data and Web data, then utilizes an alpha-shape partitioning method to infer the boundaries of imprecise AOIs. E-partition [10] cluster AOIs with the goal of equitable workload assignment. It first predicts the service time of a worker given a specific AOI, which is fed into the optimization algorithm by setting the workload balance as the optimization goal. C-AOI [30] introduces the order location, satellite image data and utilizes an image-based model that converts the problem to an instance segmentation task in the image field.

Overall, fixed-grid and road-networks solely rely on the fixed grid or road networks as the AOI boundary, which can perceive the geographical meaning of the AOIs, but cannot meet the service-related goal. Optimization-based methods utilize optimization method for improving the service-semantic goals. They typically require much effort to construct a specific optimization model for one or two service goals. However, they lack the ability to effectively incorporate adequate spatial-temporal features, and the generalizability to accommodate different service-semantic goals.

7 CONCLUSION

In this paper, we introduce a novel approach to generalize Graph-based Deep Reinforcement Learning (GRL) for AOI segmentation. In this approach, we conceptualize the entire urban space as a graph, where each node represents an atom-region and the edges represent spatial connectivity between atom regions. We address the AOI segmentation problem by formulating it as a sequential decision problem on the graph, which adjusts one node (i.e., atom region) along AOI's border at each decision step. To model this sequential decision process, we employ a Markov Decision Process (MDP), resulting in a new framework called GRL4AOI. This framework effectively incorporates various region-related features, such as region size, order number, and trajectories, using deep learning capabilities. Additionally, it allows for the flexible integration of service-semantic objectives by treating them as rewards to guide the learning process. To validate the framework, we implement a model equipped with Double-DQN for AOI segmentation in the logistics service domain. The model focuses on two service-semantic goals: trajectory modularity and predictability. Through extensive offline experiments and online deployment, we demonstrate the effectiveness of our proposed framework.

REFERENCES

[1] Avi Arampatzis, Marc Van Kreveld, Iris Reinbacher, Christopher B Jones, Subodh Vaid, Paul Clough, Hideo Joho, and Mark Sanderson. 2006. Web-based delineation of imprecise regions. *Computers, Environment and Urban Systems* 30, 4 (2006), 436–459.

[2] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment* 2008, 10 (2008), P10008.

[3] Liyue Chen, Jiangyi Fang, Zhe Yu, Yongxin Tong, Shaosheng Cao, and Leye Wang. 2023. A Data-driven Region Generation Framework for Spatiotemporal Transportation Service Management. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM. https://doi.org/10.1145/3580305.3599760

[4] Longbiao Chen, Daqing Zhang, Leye Wang, Dingqi Yang, Xiaojuan Ma, Shijian Li, Zhaohui Wu, Gang Pan, Thi-Mai-Trang Nguyen, and Jérémie Jakubowicz. 2016. Dynamic cluster-based over-demand prediction in bike sharing systems. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. 841–852.

[5] Dipti Dubey and Aparna Mehra. 2016. Pareto-optimal solutions for multi-objective flexible linear programming. *J. Intell. Fuzzy Syst.* 30, 1 (2016), 535–546. https://doi.org/10.3233/IFS-151778

[6] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, Vol. 96. 226–231.

[7] Santo Fortunato. 2010. Community detection in graphs. *Physics reports* 486, 3-5 (2010), 75–174.

[8] Xu Geng, Yaguang Li, Leye Wang, Lingyu Zhang, Qiang Yang, Jieping Ye, and Yan Liu. 2019. Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 3656–3663.

[9] Adrian Goldwaser and Michael Thielscher. 2020. Deep reinforcement learning for general game playing. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 34. 1701–1708.

[10] Baoshen Guo, Shuai Wang, Haotian Wang, Yunhuai Liu, Fanshuo Kong, Desheng Zhang, and Tian He. 2023. Towards Equitable Assignment: Data-Driven Delivery Zone Partition at Last-mile Logistics. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 4078–4088.

[11] Jintao Ke, Hongyu Zheng, Hai Yang, and Xiqun Michael Chen. 2017. Short-term forecasting of passenger demand under on-demand ride services: A spatio-temporal deep learning approach. *Transportation research part C: Emerging technologies* 85 (2017), 591–608.

[12] Hongtao Lei, Gilbert Laporte, and Bo Guo. 2012. Districting for routing with stochastic customers. *EURO Journal on Transportation and Logistics* 1, 1-2 (2012), 67–85.

[13] Shuai Ling, Zhe Yu, Shaosheng Cao, Haipeng Zhang, and Simon Hu. 2023. STHAN: Transportation demand forecasting with compound spatio-temporal relationships. *ACM Transactions on Knowledge Discovery from Data* 17, 4 (2023), 1–23.

[14] Yang Liu, Fanyou Wu, Cheng Lyu, Shen Li, Jieping Ye, and Xiaobo Qu. 2022. Deep dispatching: A deep reinforcement learning approach for vehicle dispatching on online ride-hailing platform. *Transportation Research Part E: Logistics and Transportation Review* 161 (2022), 102694.

[15] Xiaowei Mao, Haomin Wen, Hengrui Zhang, Huaiyu Wan, Lixia Wu, Jianbin Zheng, Haoyuan Hu, and Youfang Lin. 2023. DRL4Route: A Deep Reinforcement Learning Framework for Pick-up and Delivery Route Prediction. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining.*

[16] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. 2013. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602* (2013).

[17] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. 2015. Human-level control through deep reinforcement learning. *nature* 518, 7540 (2015), 529–533.

[18] Seyed Sajad Mousavi, Michael Schukat, and Enda Howley. 2018. Deep reinforcement learning: an overview. In *Proceedings of SAI Intelligent Systems Conference (IntelliSys) 2016: Volume 2*. Springer, 426–440.

[19] Luc Muyldermans, Dirk Cattrysse, Dirk Van Oudheusden, and Tsippy Lotan. 2002. Districting for salt spreading operations. *European Journal of Operational Research* 139, 3 (2002), 521–532.

[20] Sijie Ruan, Cheng Long, Xiaodu Yang, Tianfu He, Ruiyuan Li, Jie Bao, Yiheng Chen, Shengnan Wu, Jiangtao Cui, and Yu Zheng. 2022. Discovering Actual Delivery Locations from Mis-Annotated Couriers' Trajectories. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*. IEEE, 3241–3253.

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

[22] Hado Van Hasselt, Arthur Guez, and David Silver. 2016. Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 30.

[23] Haomin Wen, Youfang Lin, Fan Wu, Huaiyu Wan, Shengnan Guo, Lixia Wu, Chao Song, and Yinghui Xu. 2021. Package Pick-up Route Prediction via Modeling Couriers' Spatial-Temporal Behaviors. In *ICDE*. IEEE, 2141–2146.

[24] Lixia Wu, Haomin Wen, Haoyuan Hu, Xiaowei Mao, Yutong Xia, Ergang Shan, Jianbin Zhen, Junhong Lou, Yuxuan Liang, Liuqing Yang, et al. 2023. LaDe: The First Comprehensive Last-mile Delivery Dataset from Industry. *arXiv preprint arXiv:2306.10675* (2023).

[25] Teng Ye, Wei Ai, Lingyu Zhang, Ning Luo, Lulu Zhang, Jieping Ye, and Qiaozhu Mei. 2020. Predicting Individual Treatment Effects of Large-scale Team Competitions in a Ride-sharing Economy. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020.* ACM, 2368–2377.

[26] Junbo Zhang, Yu Zheng, and Dekang Qi. 2017. Deep spatio-temporal residual networks for citywide crowd flows prediction. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 31.

[27] Yan Zhao, Kai Zheng, Yue Cui, Han Su, Feida Zhu, and Xiaofang Zhou. 2020. Predictive task assignment in spatial crowdsourcing: a data-driven approach. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)*. IEEE, 13–24.

[28] Yan Zhao, Kai Zheng, Yang Li, Han Su, Jiajun Liu, and Xiaofang Zhou. 2019. Destination-aware task assignment in spatial crowdsourcing: A worker decomposition approach. *IEEE Transactions on Knowledge and Data Engineering* 32, 12 (2019), 2336–2350.

[29] Yida Zhu, Liying Chen, Daping Xiong, Shuiping Chen, Fangxiao Du, Jinghua Hao, Renqing He, and Zhizhao Sun. 2023. C-AOI: Contour-based Instance Segmentation for High-Quality Areas-of-Interest in Online Food Delivery Platform. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2023, Long Beach, CA, USA, August 6-10, 2023.* ACM, 5750–5759.

[30] Yida Zhu, Liying Chen, Daping Xiong, Shuiping Chen, Fangxiao Du, Jinghua Hao, Renqing He, and Zhizhao Sun. 2023. C-AOI: Contour-based Instance Segmentation for High-Quality Areas-of-Interest in Online Food Delivery Platform. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 5750–5759.

A IMPLEMENTATION DETAILS

A.1 Atom-Region Graph Construction

It is based on road networks in two steps. The first step filters data and exports the image. The second step executes the image partitioning algorithm.

Step 1: Filter the road network data and export it as an image. We obtain road network data from OpenStreetMap[2] and filter it based on a specified area. Then we exclude certain roads that are unlikely to be useful, such as those with tags like 'building' or 'subway'. We select a fine-grained road network (e.g., highways and rivers) to partition the atomic regions, and a coarse-grained road network (e.g. only some kinds of highway like 'highway: primary') as the basis for the initial road network partitioning. Because identifying atomic regions requires a higher level of granularity. We then export the road network as a binary image.

Step 2: Execute the road network image partitioning and recognition algorithm [3] to obtain the partitions. We first dilate the image to eliminate closely parallel roads. Then find the connected components and expand them to remove roads, so that we could obtain a label matrix (called AOI label map) containing only AOIs. Each AOI corresponds to an atom region in the image. Based on the AOI label map, we can easily determine the adjacency relationships between each AOI and construct the edges of the atom-region graph.

[2] https://www.openstreetmap.org/about