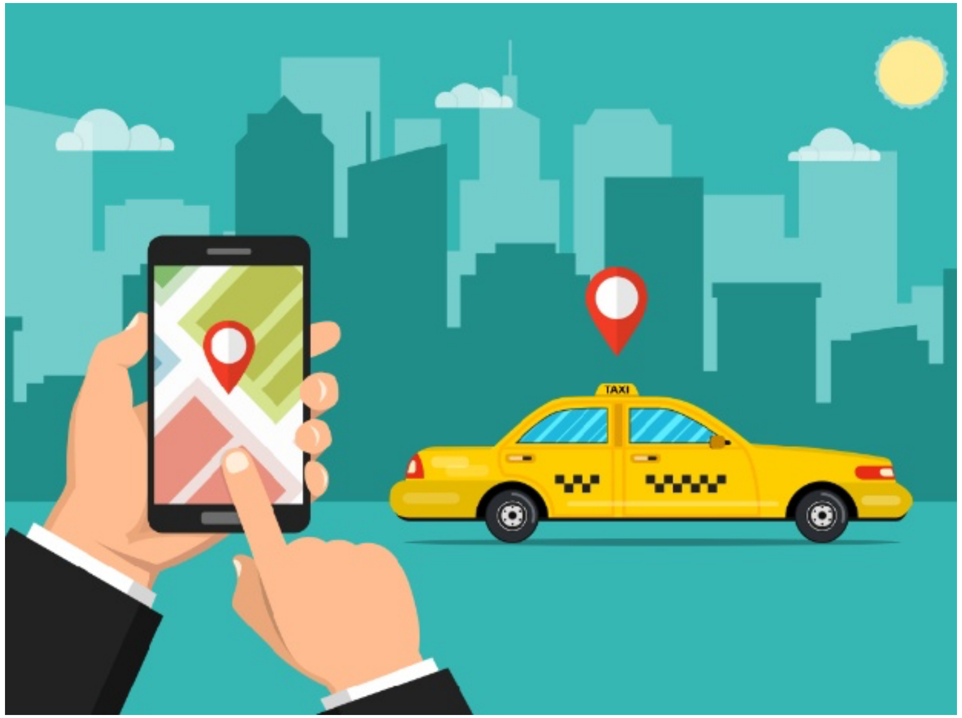
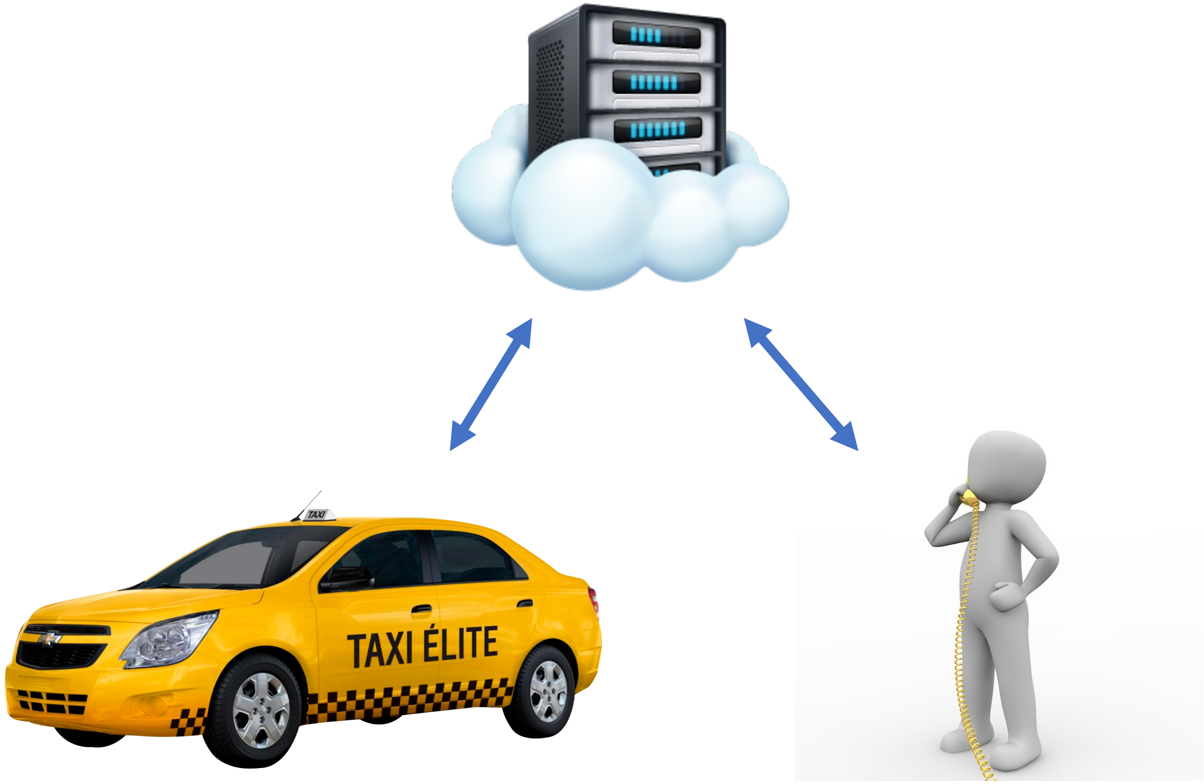
Multi-Agent Reinforcement Learning for Order-Dispatching via Order-Vehicle Distribution Matching

Ming Zhou, Jiarui Jin, Weinan Zhang, Zhiwei Qin, Yan Jiao,
Chenxi Wang, Guobin Wu, Yong Yu, Jieping Ye

Shanghai Jiao Tong University, Didi AI Labs

Ride-Hailing & Order-Vehicle Dispatching



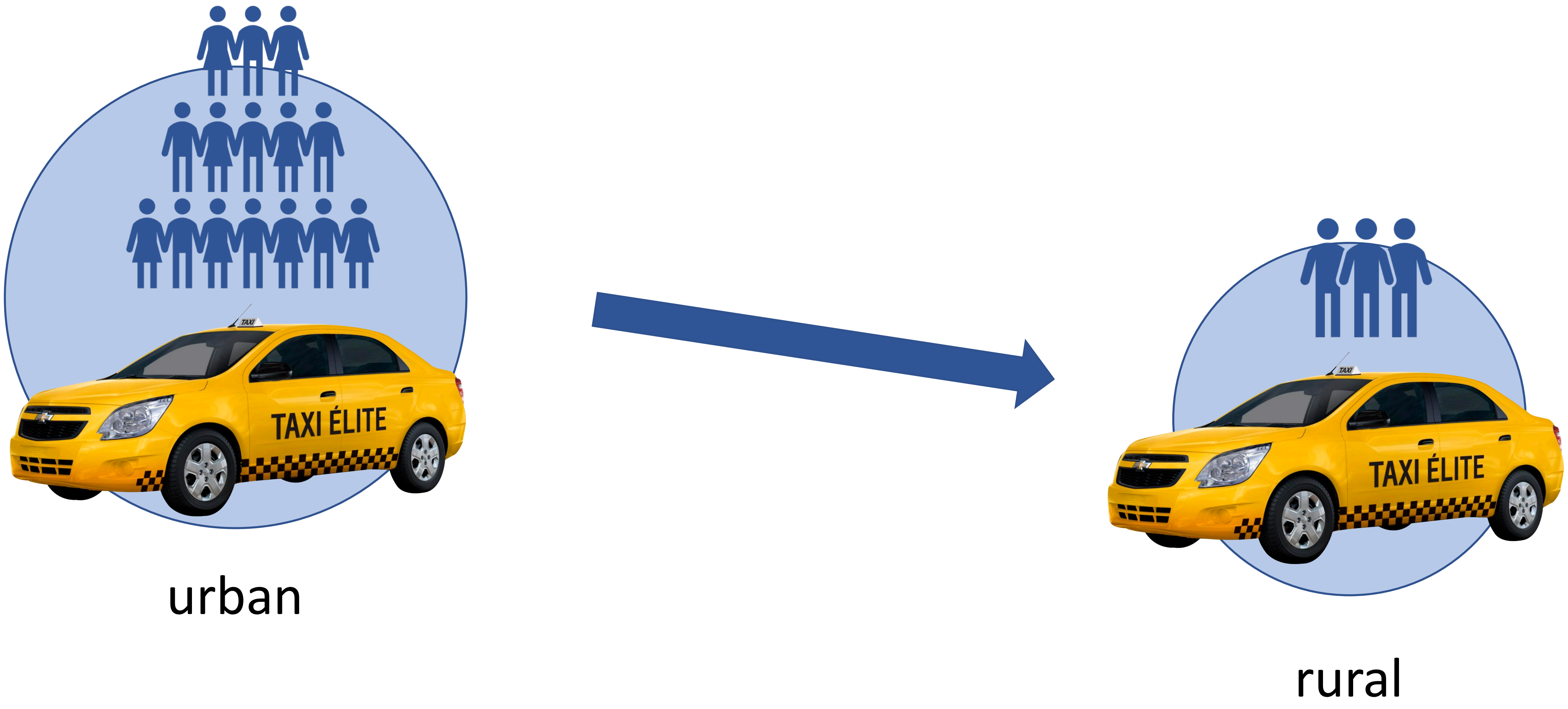


Metrics for order dispatching

- ADI – Accumulative driver income
- ORR – Order response rate

Generally speaking, higher ORR means higher ADI

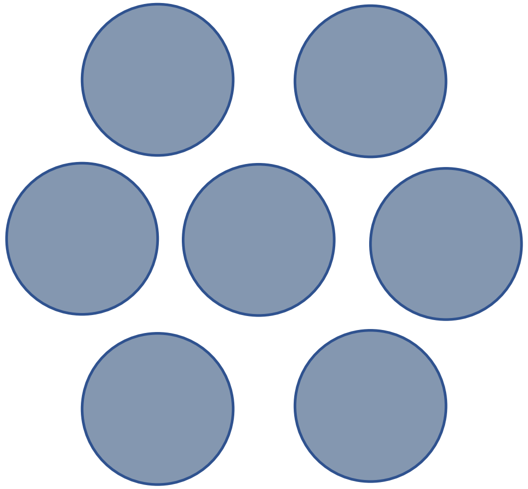
Long-term vs. Long-distance Serving



Depart City into Many Dispatching Regions

Dongcheng district, Beijing

It reduces the waiting time

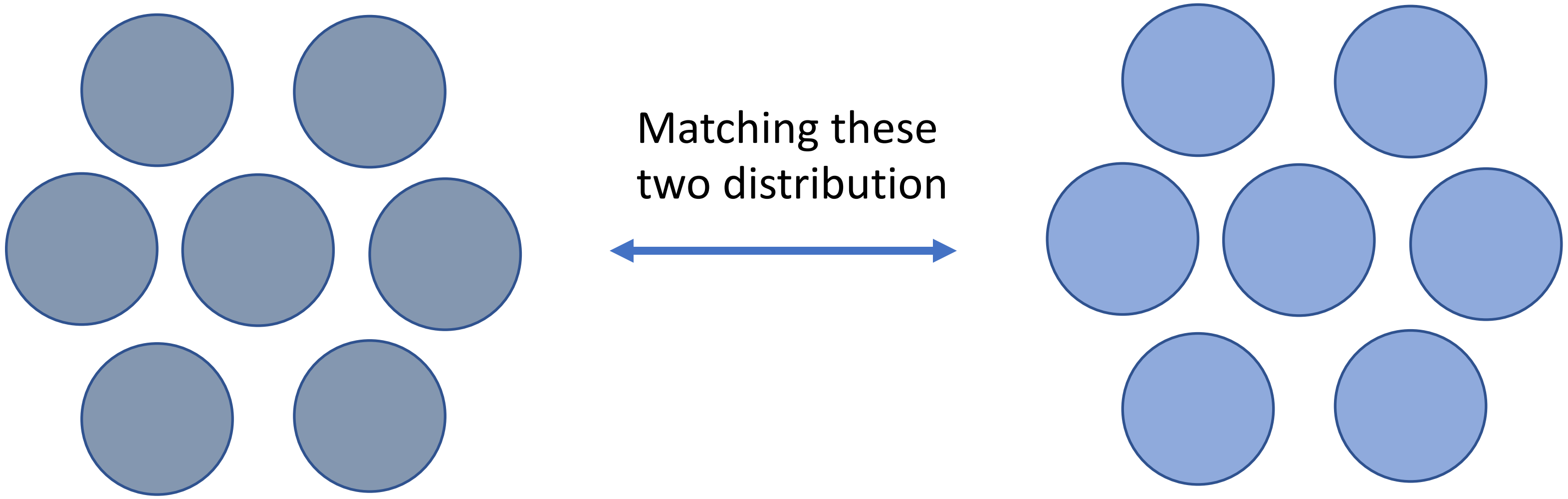


Dispatching regions

Our targets

- Reduce the waiting time.
- Higher order response rate.
- Higher accumulative driver income.

Distribution Matching



Order distribution at t

Vehicle distribution at t

Time Step	0	…	t	…	T
Order Distribution	$\mathcal{D}_0^o$	…	$\mathcal{D}_t^o$	…	$\mathcal{D}_T^o$
Vehicle Distribution	$\mathcal{D}_0^v$	…	$\mathcal{D}_t^v$	…	$\mathcal{D}_T^v$

Distribution Matching

Matching Strategy = $\arg\min D_{KL}(\mathcal{D}_{t+1}^{o} \parallel \mathcal{D}_{t+1}^{v})$

KL divergence from vehicle distribution to order distribution

Distribution Transition Formulation:

$$\boxed{\mathcal{D}_t^v \times \pi \times \mathcal{D}_t^o \rightarrow \mathcal{D}_{t+1}^v}$$

Policy for order dispatching

It means, the vehicle distribution at t+1 is determined by policy at time t

Multi-Agent Reinforcement Learning

- **State**: $\langle G, N, M, \mathcal{D}_{dest} \rangle$, represent the grid index, the number of idle vehicles, the number of valid orders and the distribution of orders' destinations respectively.
- **Action**: $\langle G_{source}, G_{dest}, T, C \rangle$, the order features, represent the source grid index, target grid index, time duration and order price respectively.
- **Reward**: propositional to the order price.

Perspective of Each Agent

Each agent aims to maximize the cumulative reward:

$$\max_{\theta} \mathcal{J} = \sum_{t=1}^{T} p(s_0) \sum_{s_{t+1}} p(s_{t+1} \mid s_t) \sum_{a_t} \pi(a_t \mid s_t) \gamma^{t-1} R(s_t, a_t \mid a_{\bar{t}})$$

$$s.t.\, D_{KL} \leq \beta$$

$$\max_{\theta} \mathcal{J} - \lambda(D_{KL} - \beta)$$

Action Selection Q-Learning

- Using the tuple <state, action> as input to handle the dynamic action space problem.
- Update rule: TD(0).

$$Q(s_t, a_t) = \alpha Q(s_t, a_t) + (1 - \alpha)\left[r_t + \gamma \mathbb{E}_{a_{t+1} \sim \pi(s_{t+1})}[Q(s_{t+1}, a_{t+1})]\right]$$

- Boltzmann-style policy for balancing exploration and exploitation:

$$\pi\left(a_t^j \mid s_t\right) = \frac{e^{Q\left(s_t, a_t^j\right)/\tau}}{\sum e^{Q\left(s_t, a_t^j\right)/\tau}}$$

temperature

KL Divergence Optimization

Objective function: TD error + KL-divergence constraint

$$\min_{\theta} \mathcal{L} = \| Q_{\theta}(s,a) - Q^{\star} \|_2 + \lambda D_{KL}$$

Gradient of D_{KL} to policy's parameters:

Rate of order

Vehicle number of grid i

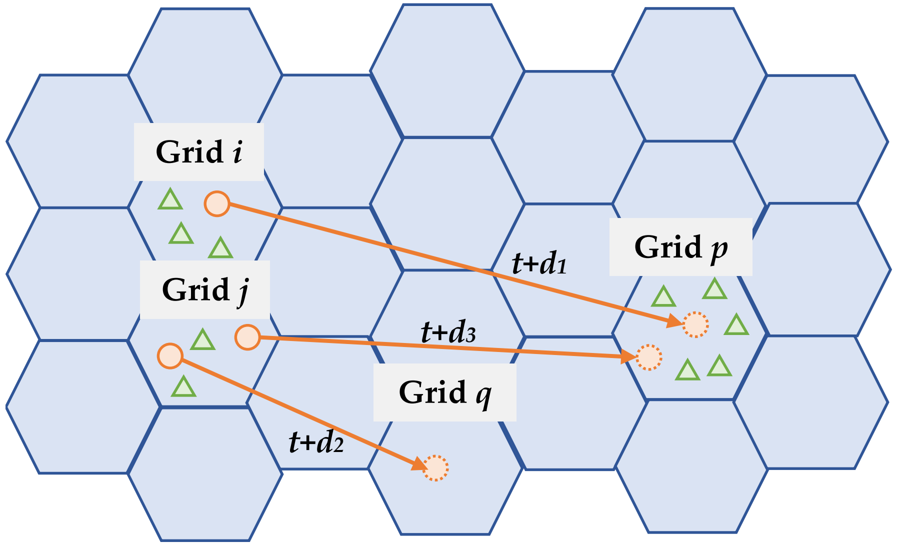
$$\nabla_{\theta_j} D_{KL} = \nabla_{\pi_j} D_{KL} \cdot \nabla_{\theta_j} \pi_j = \boxed{c_t^j} \sum_{i=1}^{N} \boxed{p_{t+1}^i} \left[\frac{1}{\boxed{N_{vehicle}}} - \frac{1}{\boxed{n_{t+1}^i}} \right] \cdot \nabla_{\theta_j} \pi_j$$

Idle vehicle number

Total number of vehicles

Compare to Coordination

- Communication requires intensive interaction.
- In order-dispatching scenario, agents in the same grid interact with others at most one time.



It shows that the order-dispatching process of each grid at time t , and different order has different duration of d , so the vehicles will arrive at the destination grids at different time, and vehicles serve different orders will be assigned to different grids, then it is hard to form continuous interactions and communication between vehicles.

Experiments

Baselines

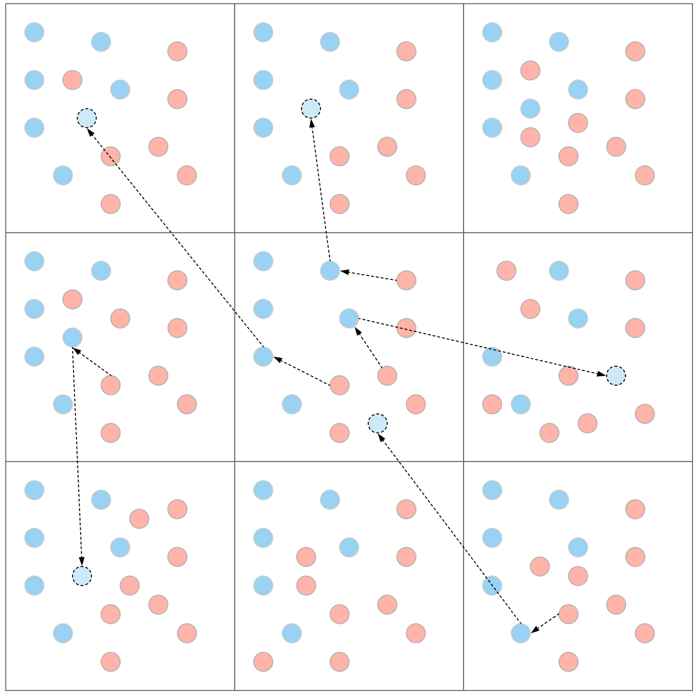
- **NOD** - Nearest-distance Order Dispatching (NOD) algorithm
- **IL** - Independent Q-Learning
- **MDP** - a planning and learning method based on decentralized multi-agent deep reinforcement learning and centralized combinatorial optimization.

Experiment: Particle Migration

- Including 10x10 grids
- Blue particle – Agent.
- Red particle – Order.
- Red particles generated by given distribution

  Number: $\sim \mathcal{N}(\mu_t, \sigma_t)$, Destination: $\sim U(0, N)$

- Agents can only select orders in the same grid.
- Reward: $R(s,a) = 0.1 \times \| \#source - \#target \|_2$
- Agents will migrate to other grids by serving orders.



A part of the Particle Environment

Experiment: Particle Migration

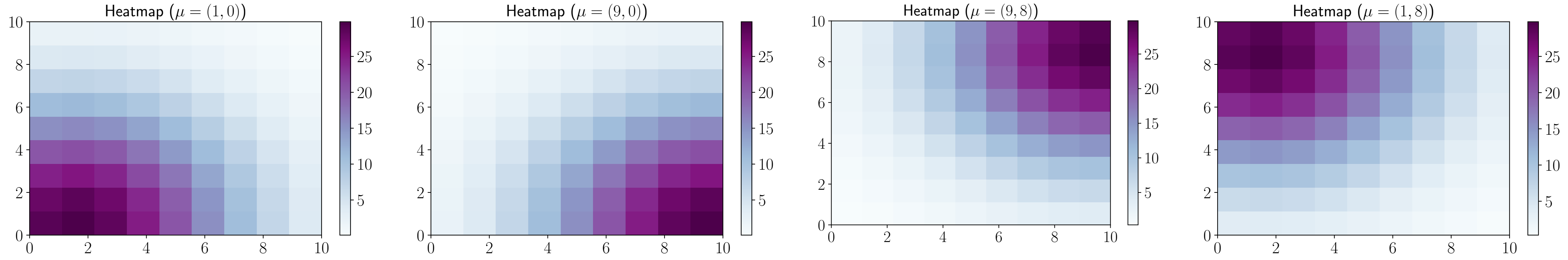


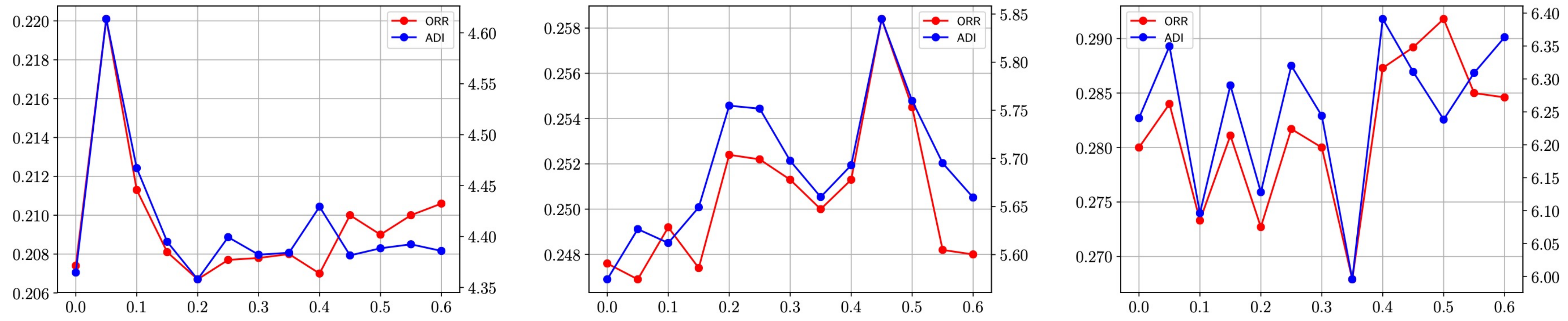
Figure 1. Order distribution, $\| \mu_t - \mu_{t+1} \|_2 = 8$

Experiment: Particle Migration

Order Distribution Divergence	Low		Medium		High	
Metrics	ADI	ORR	ADI	ORR	ADI	ORR
IL	+12.5%	+6.94%	+11.5%	+6.3%	+6.68%	+2.32%
MDP	+14.5%	+8.94%	+13.3%	+6.69%	+7.28%	+3.42%
KL-Based	**+25.12%**	**+13.40%**	**+20.94%**	**+7.89%**	**+13.47%**	**+4.61%**

Table 1. Performance comparison in terms of ADI and ORR with respect to NOD

Experiment: Particle Migration



(a) $\| \mu_t - \mu_{t+1} \|_2 = 1$

(b) $\| \mu_t - \mu_{t+1} \|_2 = 2$

(c) $\| \mu_t - \mu_{t+1} \|_2 = 4$

Figure 2. ORR and ADI performance under different λ settings.

Experiment: Real-World Testing

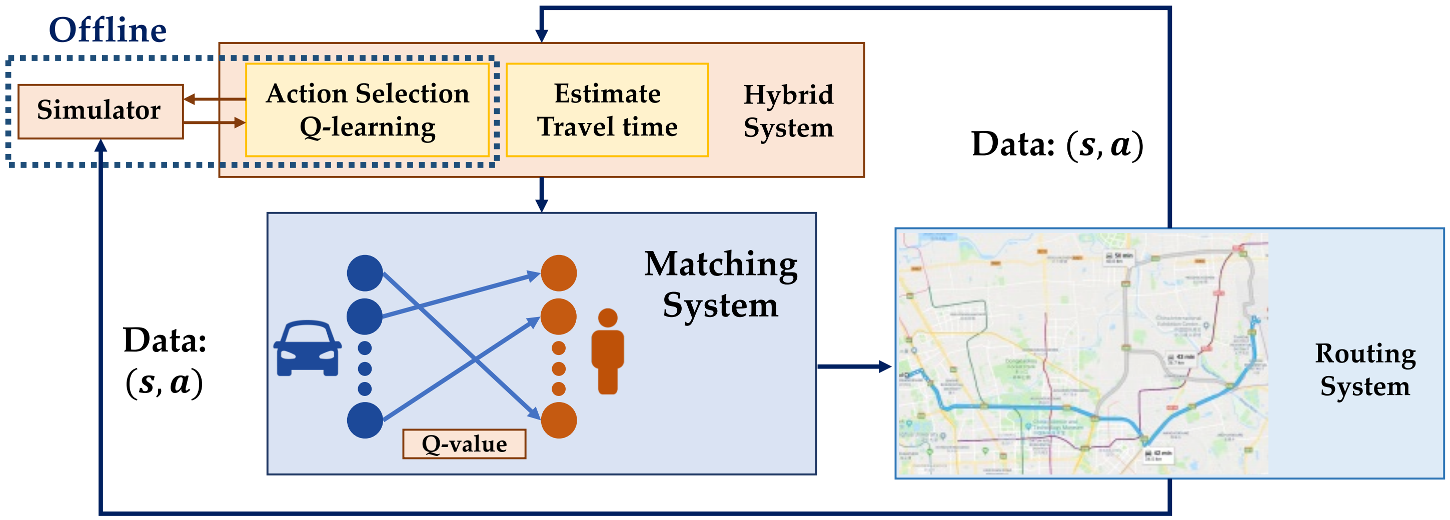
- Conduct experiments on an open source grid-based environment simulator provided by Didi Chuxing.
- Divide city into N hexagonal grids.
- The travel distance between neighboring regions is approximately 2.2km and the time interval is 10min.
- 3 cities in one month.

Experiment: Real-World Testing

City	City A		City B		City C	
Metrics	ADI	ORR	ADI	ORR	ADI	ORR
IL	+4.69%	+1.68%	+2.96%	+1.11%	+4.72%	+2.05%
MDP	+5.80%	+1.89%	+3.69%	+2.63%	+5.98%	+2.14%
KL-Based	+**6.46**%	+**3.07**%	+**4.94**%	+**3.30**%	+**6.12**%	+**3.01**%

Table 2. Performance comparison in terms of ADI and ORR with respect to NOD

Deployment



A hybrid system which incorporates our method with routing planning and arrival time estimation.

Summary

- Dividing city into multiple dispatching regions to ensure reasonable waiting time.
- A decentralized and no explicit coordination MARL method for order dispatching and ride-hailing.
- Constraint distribution matching method can handle different traffic cases.

Thanks