Decision Intelligence for Two-sided Marketplaces

Tony Qin
foreva.ai

Chengchun Shi
LSE

Hongtu Zhu
UNC Chapel Hill

AAAI-2025 Tutorial
February 26, 2025

Agenda

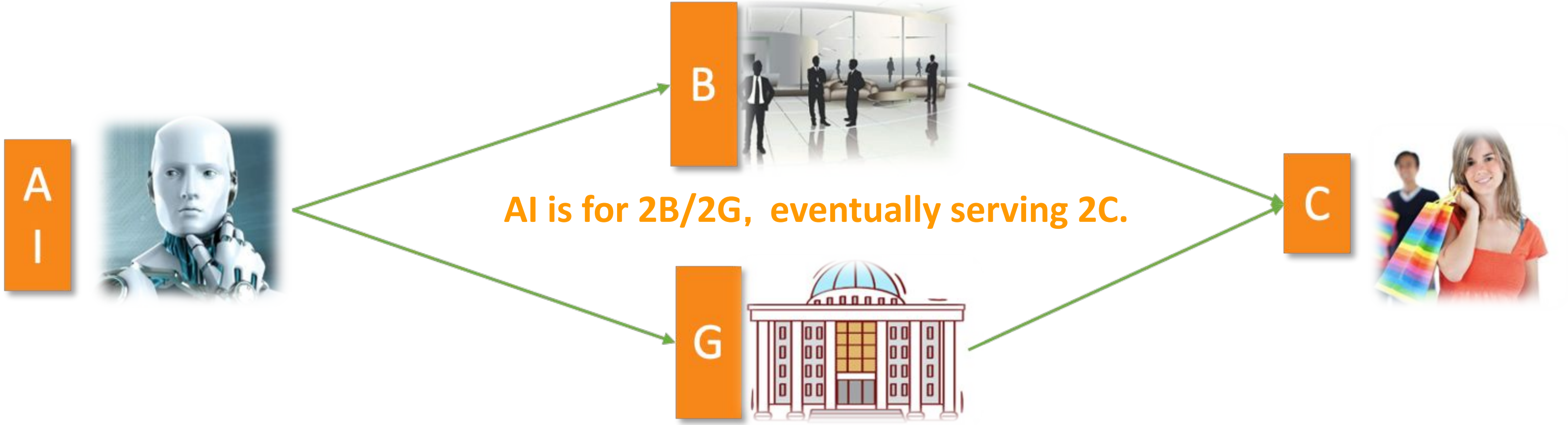
- ❏ Foundations of Two-Sided Marketplaces (Hongtu, 35 min)
- ❏ Optimizing Policies for Strategic Decision-Making (Tony, 75 min)
- ❏ A/B Testing: Policy Evaluation and Experimental Design (Chengchun, 75 min)
- ❏ Leveraging LLMs and Digital Twins for Marketplaces (Tony, 20 min)

Foundations of Two-Sided Marketplaces

Hongtu Zhu

UNC Chapel Hill

Consumer-Centric AI



AI is for 2B/2G, eventually serving 2C.

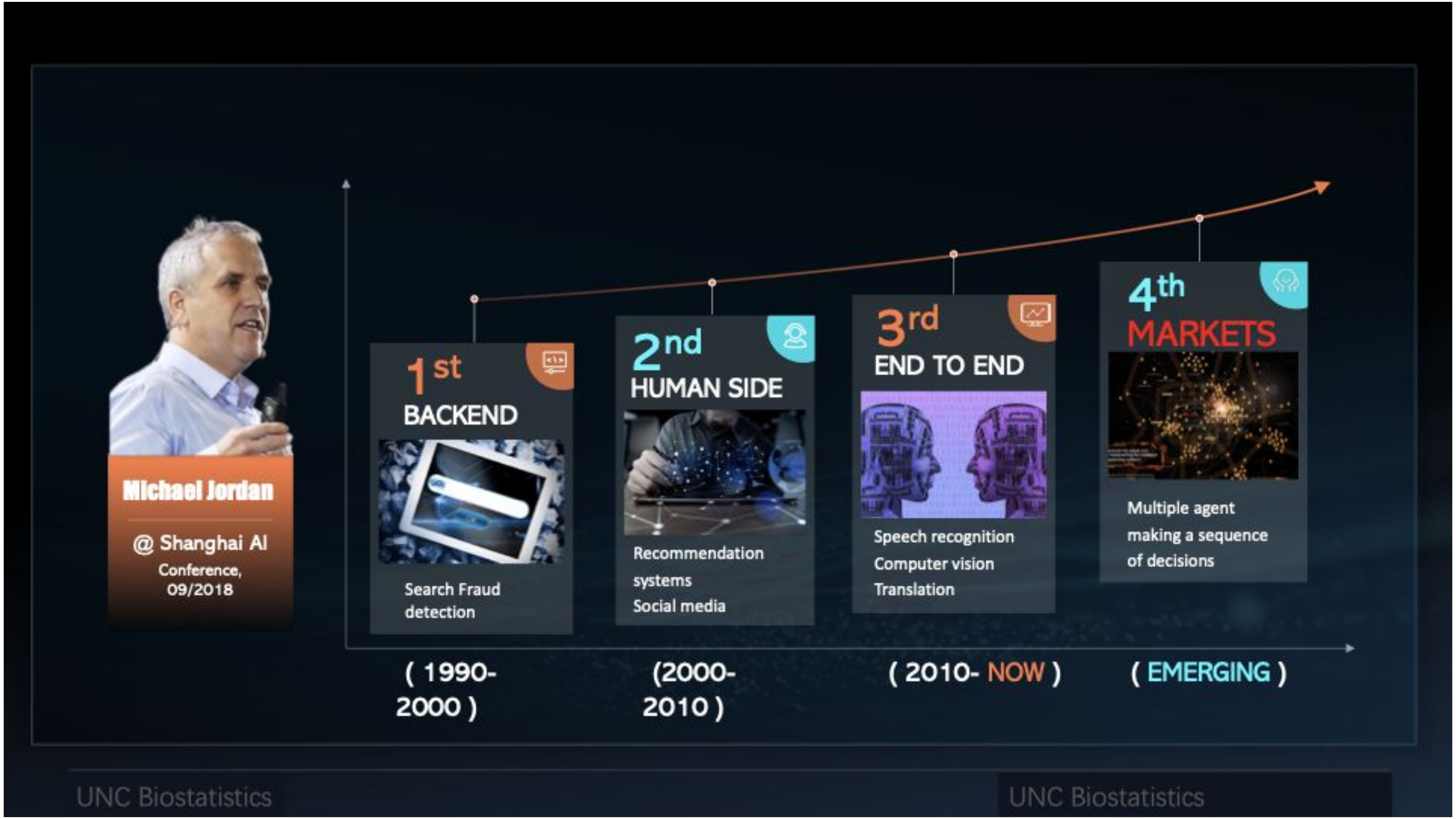
Society

Economy

Business

Manufacturing

ML Successes



Michael Jordan

@ Shanghai AI Conference, 09/2018

1st BACKEND

Search Fraud detection

(1990- 2000)

2nd HUMAN SIDE

Recommendation systems

Social media

(2000- 2010)

3rd END TO END

Speech recognition

Computer vision

Translation

(2010- NOW)

4th MARKETS

Multiple agent making a sequence of decisions

(EMERGING)

What is a Two-sided Marketplace?

A **two-sided marketplace** is a market where one or more platforms facilitate interactions between two (or more) distinct user groups. The platform's goal is to bring both sides “on board” by appropriately structuring incentives, such as pricing and accessibility (Rochet –Tirole, 2006).

Key Characteristics

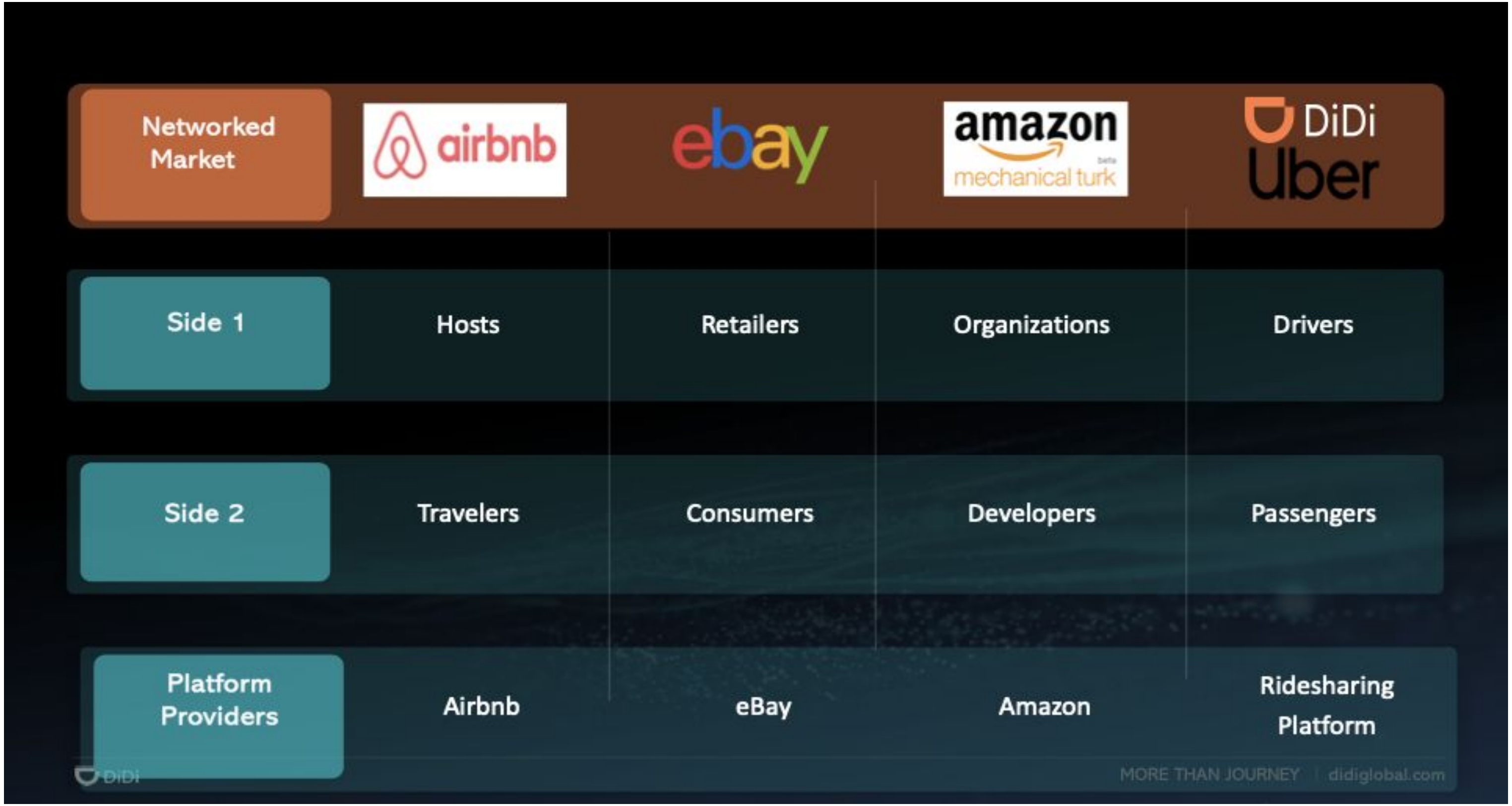
- **Interdependence:** The value for one side depends on the presence and engagement of the other.
- **Platform as an Intermediary:** The marketplace acts as a bridge, reducing transaction costs and enhancing trust.
- **Pricing Strategies:** Platforms use pricing and subsidies to attract and balance supply and demand.

Beyond Pricing

Alvin E. Roth. Nobel Memorial Prize in Economic, @ SIGKDD 2018, 08/2018
“— In many markets, you care who you are dealing with, and prices don’t do all the work

— (In some matching markets (e.g., organ donations), we don’t even let prices do any of the work…) ”

Examples of Two-sided Marketplace

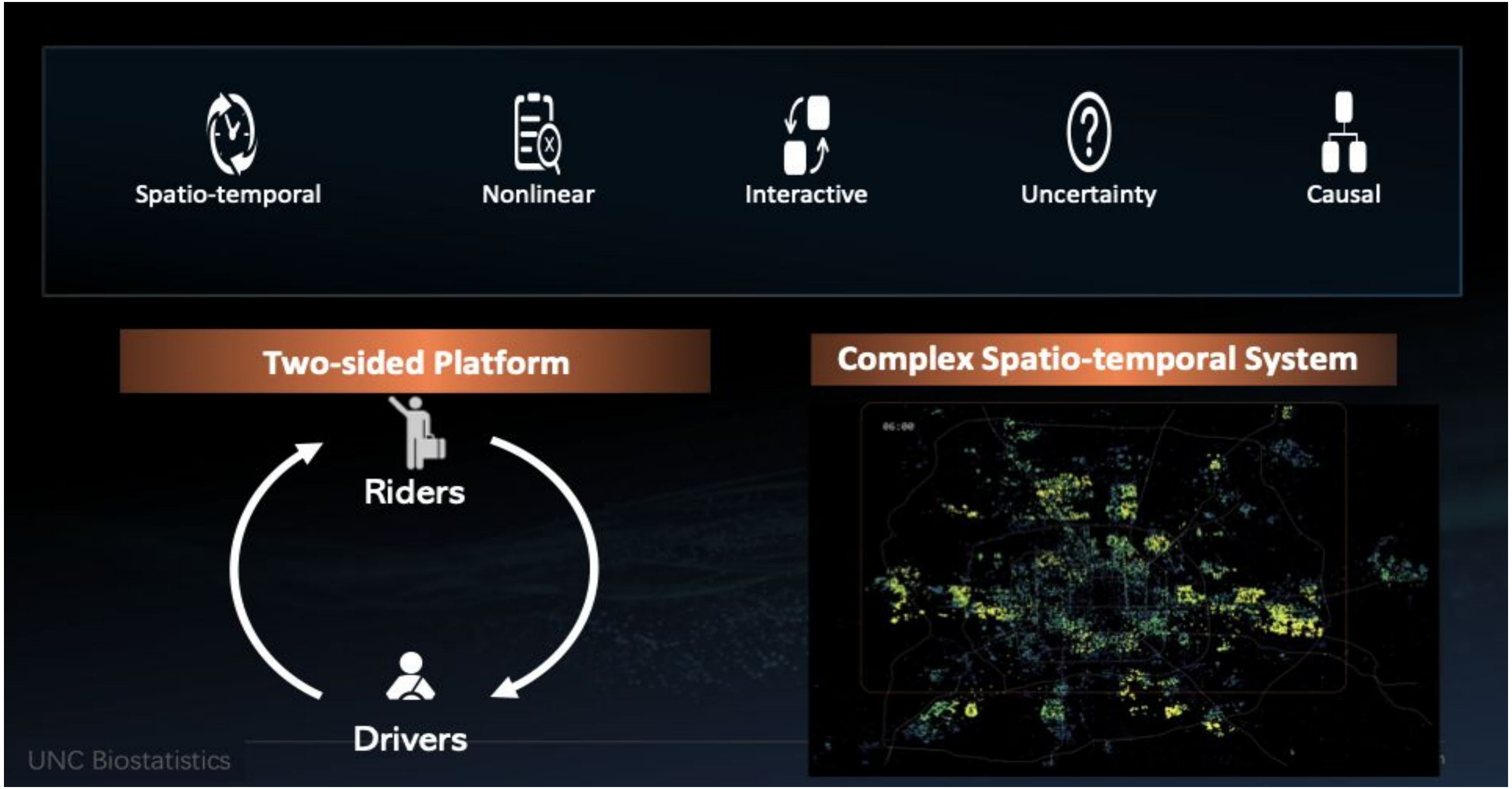


Networked Market	airbnb	ebay	amazon mechanical turk beta	DiDi Uber
Side 1	Hosts	Retailers	Organizations	Drivers
Side 2	Travelers	Consumers	Developers	Passengers
Platform Providers	Airbnb	eBay	Amazon	Ridesharing Platform

DiDi

MORE THAN JOURNEY | didiglobal.com

Ride-sharing is a Complex System



Spatio-temporal

Nonlinear

Interactive

Uncertainty

Causal

Two-sided Platform

Riders

Drivers

Complex Spatio-temporal System

UNC Biostatistics

Digital Twins for Marketplaces

Digital Twins for Marketplaces refer to virtual replicas of two-sided marketplaces that simulate and analyze real-world interactions between buyers and sellers. These AI-driven models integrate real-time data, historical trends, and behavioral analytics to optimize decision-making, improve market efficiency, and test policies in a risk-free environment.

Key Modules:

1️⃣**Supply-Demand Diagnosis** – Identify inefficiencies and bottlenecks.
2️⃣**Supply-Demand Prediction** – Forecast market trends and user behavior.
3️⃣**Policy Optimization** – Enhance pricing, matching, and incentives.
4️⃣**Policy Evaluation** – Measure impact before real-world deployment.
5️⃣**Lifetime Value** – Maximize long-term user engagement and revenue.

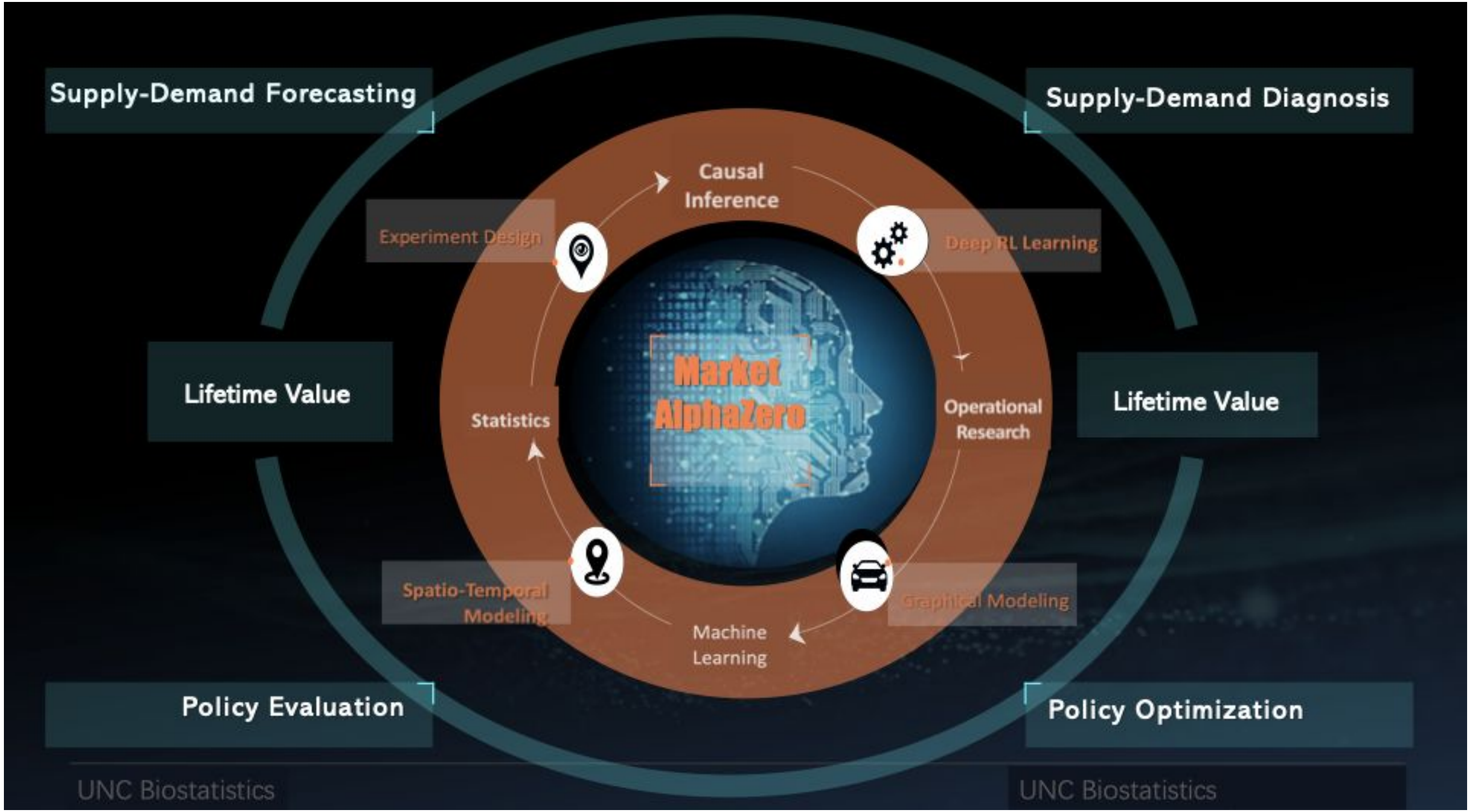
Why It Matters?

✅ Data-driven decision-making
✅ Improved efficiency and user experience

Why It Matters?

✅ Risk-free policy experimentation
✅ Long-term marketplace sustainability

Digital Twins for Marketplaces



Spatio-temporal Causal Digital Twin

Key Components:

- ❑ **Macroscopic Business Indicators** – The overall supply-demand balance is influenced by external factors and platform policies.
- ❑ **Supply-Demand Matching Degree** – Measures how effectively supply meets demand at a given time.
- ❑ **Supply Side** – Availability of providers responding to demand fluctuations.
- ❑ **Demand Side** – Volume of user requests at different times and locations.
- ❑ **Platform Policies (θ_1: Dispatch & Scheduling)**
- ❑ **Platform Policies (θ_2: Pricing & Subsidies)**
- ❑ **External Confounders** – Weather, holidays, workdays, infrastructure, and government policies.

Mathematical Framework:

- Future macro indicators depend on supply, demand, and matching degree.
- Matching degree is influenced by real-time supply, demand, dispatch policies, and environmental factors.
- Supply levels depend on past demand, matching efficiency, pricing, and subsidies.
- Demand levels are affected by past matching, pricing, subsidies, and environmental conditions.

Supply-Demand Diagnosis

Fan, Z., Luo, S., Qie, X., Ye, J., and *Zhu HT.* Graph-based equilibrium metrics for dynamic supply-demand systems with applications to ride-sourcing platforms。*Journal of American Statistical Association,* 2021，116, 1688-1699.

Chin, Alex, and Zhiwei Qin. A unified representation framework for rideshare marketplace equilibrium and efficiency. *Proceedings of the 31st ACM International Conference on Advances in Geographic Information Systems*. 2023.

What is Supply-Demand Diagnosis?

Supply-Demand Diagnosis is the process of analyzing and identifying mismatches between supply and demand in a given system. It aims to uncover **inefficiencies, imbalances, and factors** affecting the equilibrium between what is available (supply) and what is needed (demand).

Key Components of Supply-Demand Diagnosis:

1. **Data Collection & Analysis** – Gathering historical and real-time data on supply and demand trends.
2. **Hotspot Detection**– Identifying shortages (demand exceeds supply) or surpluses (supply exceeds demand).
3. **Causal Analysis** – Investigating underlying factors such as pricing, external market conditions, policies, or operational constraints.
4. **Impact Assessment** – Evaluating how imbalances affect business performance, customer satisfaction, and operational efficiency.
5. **Corrective Strategies** – Recommending policies, pricing adjustments, or operational changes to restore balance.

Why is Supply-Demand Diagnosis important?

- **Prevents inefficiencies** – Helps avoid overproduction, stockouts, or resource wastage.
- **Improves decision-making** – Enables data-driven adjustments in pricing, inventory, workforce, or infrastructure.
- **Enhances market stability** – Reduces volatility by ensuring better alignment between supply and demand.
- **Boosts profitability** – Helps optimize costs and maximize revenues by addressing imbalances effectively.

Example Applications:

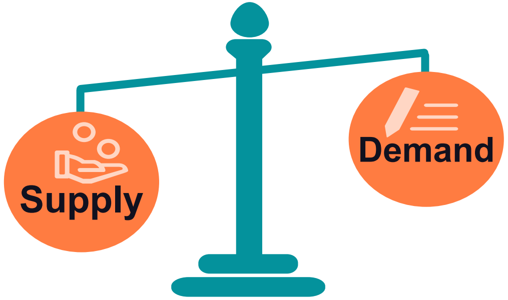
- **Ride-sharing platforms:** Diagnosing supply-demand imbalances in different locations and times to adjust dynamic pricing.
- **Retail & E-commerce:** Identifying stock shortages or excess inventory to optimize restocking strategies.
- **Healthcare systems:** Assessing hospital bed availability against patient needs to optimize resource allocation.
- **Energy markets:** Balancing electricity production with consumption to prevent blackouts or inefficiencies.

Graph-based Equilibrium Metric

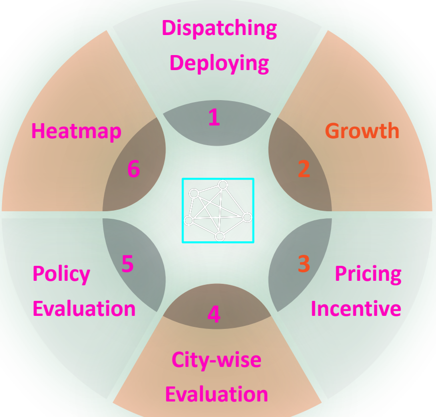
Motivation

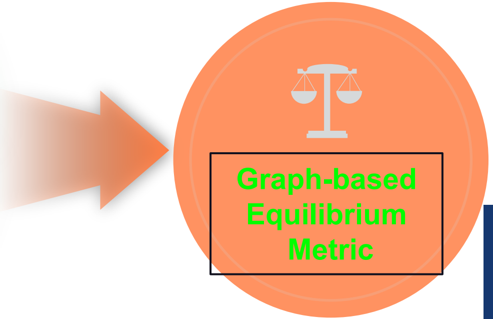
- A metric for measuring supply and demand equilibrium
- Objective function for improving strategies of dispatching, pricing, and incentive optimizations

Graph-Based Equilibrium Metrics are a set of measures used to evaluate and quantify the balance between supply and demand in complex systems that can be represented as graphs (networks). These metrics help assess how well supply nodes (e.g., service providers, resources) are matched with demand nodes (e.g., customers, tasks) while considering connectivity, constraints, and network effects.



Supply-Demand Imbalance

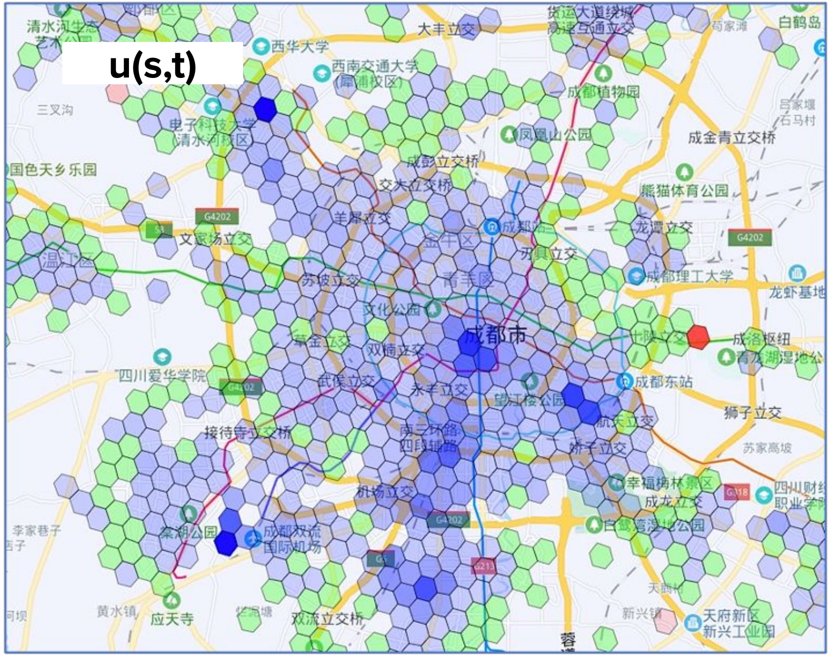


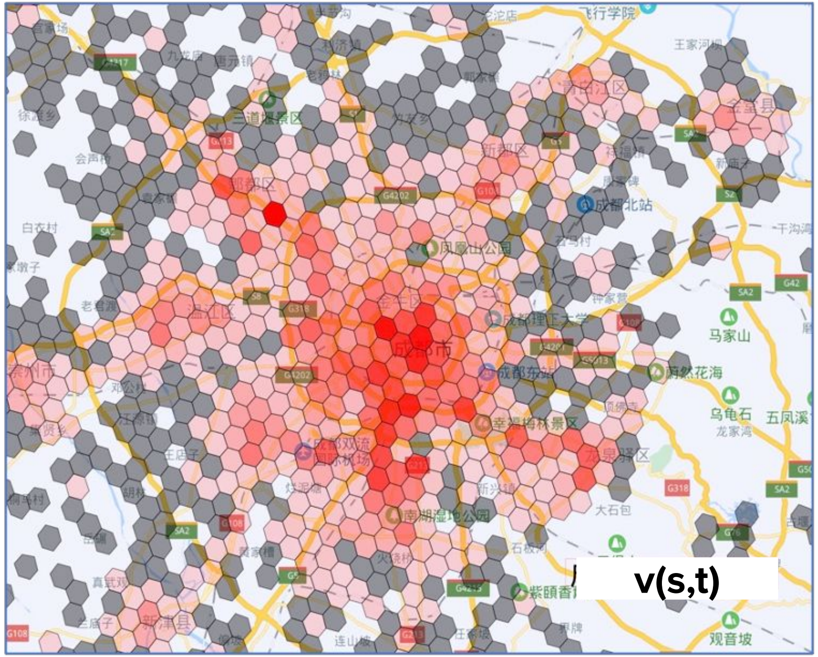


Graph-based Equilibrium Metric

1. **Graph Representation:**
   - The system is represented as a **bipartite graph** (e.g., drivers ↔ riders in ride-sharing) or a **general graph** (e.g., supply chains, logistics, or energy distribution).
   - Nodes represent supply and demand entities at a certain level, while edges represent potential interactions or assignments.
2. **Matching Efficiency:**
   - Measures how effectively supply nodes are connected to demand nodes.
   - Examples: **Maximum matching ratio, average matching distance, or latency in matching.**
3. **Flow Equilibrium:**
   - Examines the distribution of supply relative to demand across the network.
   - Example: **Wasserstein distance** (comparing the distributions of supply and demand over the graph).
4. **Hotspot Identification:**
   - Identifies overloaded nodes or edges where demand exceeds capacity.
   - Example: **Edge congestion score** (quantifies imbalance in flow through different pathways).
5. **Dynamical Adjustments & Policy Evaluation:**
   - Assesses how small interventions (e.g., price changes, routing optimizations) shift equilibrium states.
   - Example: **Sensitivity analysis of equilibrium shifts with small perturbations.**

Graph Representation

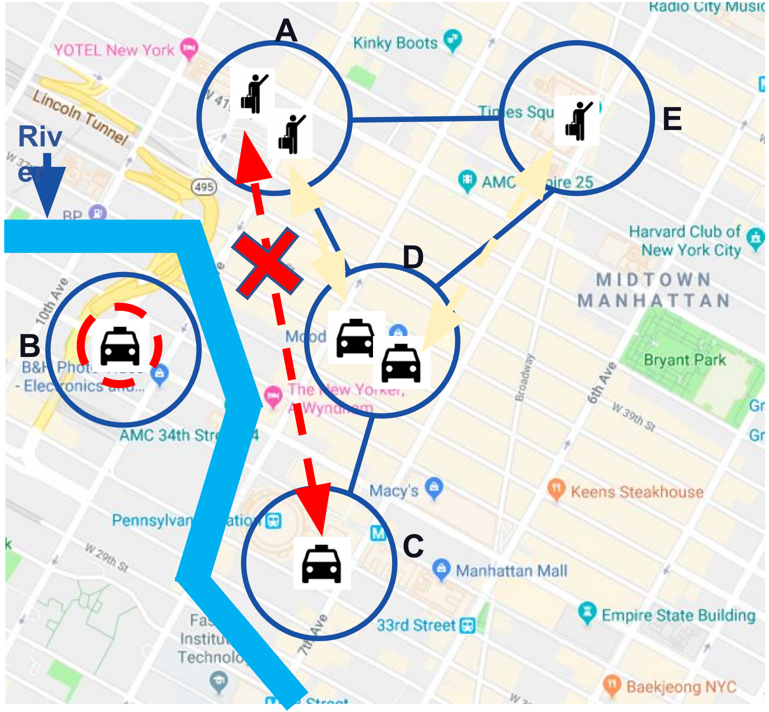




Dynamic Supply Map u(s, t)

Dynamic Demand Map v(s, t)

Matching Efficiency



Notations

(1) Undirected (or directed) graph $G = (V, E)$; weight matrix $W = (w_{ij})$. If $(v_i, v_j) \notin E, w_{ij} = \infty$.

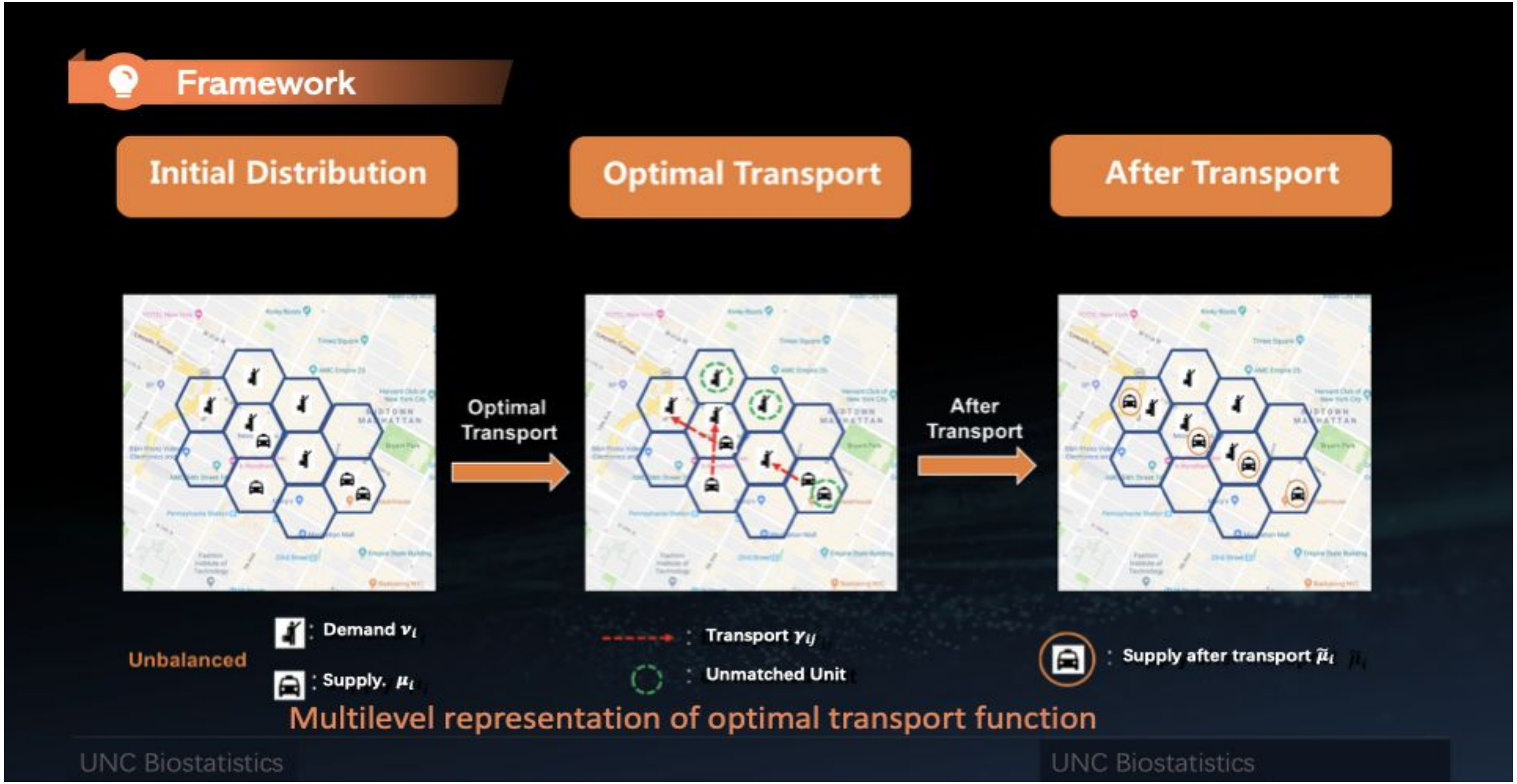
(2) The transport cost on graph G from v_i to v_j is defined as

$$c_{ij} = \min_{K\geq 0,(i_k)_{k=0}^{K}: v_i \to v_j} \left\{\sum_k w_{i_k, i_{k+1}} : \forall k \in [\![0, K-1]\!], (v_{i_k}, v_{i_{k+1}}) \in E\right\}$$

We can introduce a transport cost matrix on (G, W), denoted as $C = (c_{ij}) \in R^{N \times N}$, which is asymmetric when the graph is directed.

(3) Two discrete Lebesgue measures $\mu, \nu \in M_+(V)$ with locally finite mass. We define $\mu_j = \mu(v_j)$ and $\nu_j = \nu(v_j)$ as the point masses at vertex v_j for the two. $\boldsymbol{\mu} = \sum_{i=1}^{N} \mu_i$ and $\boldsymbol{\nu} = \sum_{i=1}^{N} \nu_i$ may be unequal to each other.

Flow Equilibrium



Framework

Initial Distribution

Optimal Transport

After Transport

Unbalanced

: Demand ν_i

: Supply. μ_i

: Transport γ_{ij}

: Unmatched Unit

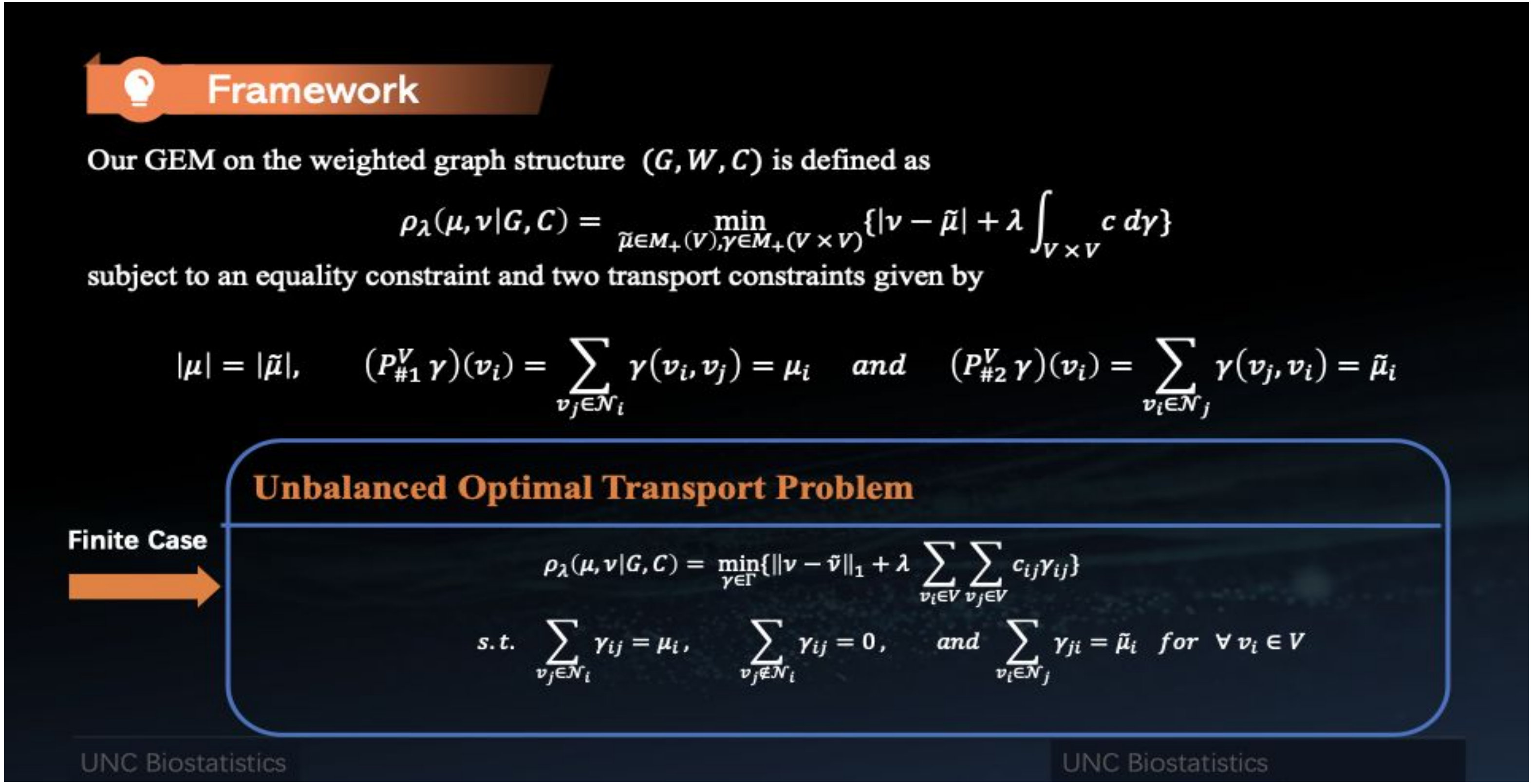
: Supply after transport $\tilde{\mu}_i$

Multilevel representation of optimal transport function

UNC Biostatistics

UNC Biostatistics

Graph-based Equilibrium Metrics (GEMs)



Framework

Our GEM on the weighted graph structure (G, W, C) is defined as

$$\rho_\lambda(\mu, \nu | G, C) = \min_{\tilde{\mu} \in M_+(V), \gamma \in M_+(V \times V)} \{|\nu - \tilde{\mu}| + \lambda \int_{V \times V} c \, d\gamma\}$$

subject to an equality constraint and two transport constraints given by

$$|\mu| = |\tilde{\mu}|, \qquad (P^V_{\#1}\, \gamma)(v_i) = \sum_{v_j \in \mathcal{N}_i} \gamma(v_i, v_j) = \mu_i \quad and \quad (P^V_{\#2}\, \gamma)(v_i) = \sum_{v_i \in \mathcal{N}_j} \gamma(v_j, v_i) = \tilde{\mu}_i$$

Finite Case

Unbalanced Optimal Transport Problem

$$\rho_\lambda(\mu, \nu | G, C) = \min_{\gamma \in \Gamma} \{\|\nu - \tilde{\nu}\|_1 + \lambda \sum_{v_i \in V} \sum_{v_j \in V} c_{ij} \gamma_{ij}\}$$

$$s.t. \sum_{v_j \in \mathcal{N}_i} \gamma_{ij} = \mu_i, \quad \sum_{v_j \notin \mathcal{N}_i} \gamma_{ij} = 0, \quad and \sum_{v_i \in \mathcal{N}_j} \gamma_{ji} = \tilde{\mu}_i \quad for \ \forall\, v_i \in V$$

UNC Biostatistics

UNC Biostatistics

Two-sided View of Marketplace

- Demand-centric view

$$A_d = \frac{\sum_{i,t} m_{i,t} v_{i,t}}{\sum_{i,t} v_{i,t}} = \mathbb{E}_{(i,t)\sim v}[m_{i,t}]$$

Average rider perception of market balance

- Supply-centric view

$$A_s := \frac{\sum_{i,t} m_{i,t} \tilde{\mu}_{i,t}}{\sum_{i,t} \tilde{\mu}_{i,t}} = \mathbb{E}_{(i,t)\sim\tilde{\mu}}[m_{i,t}]$$

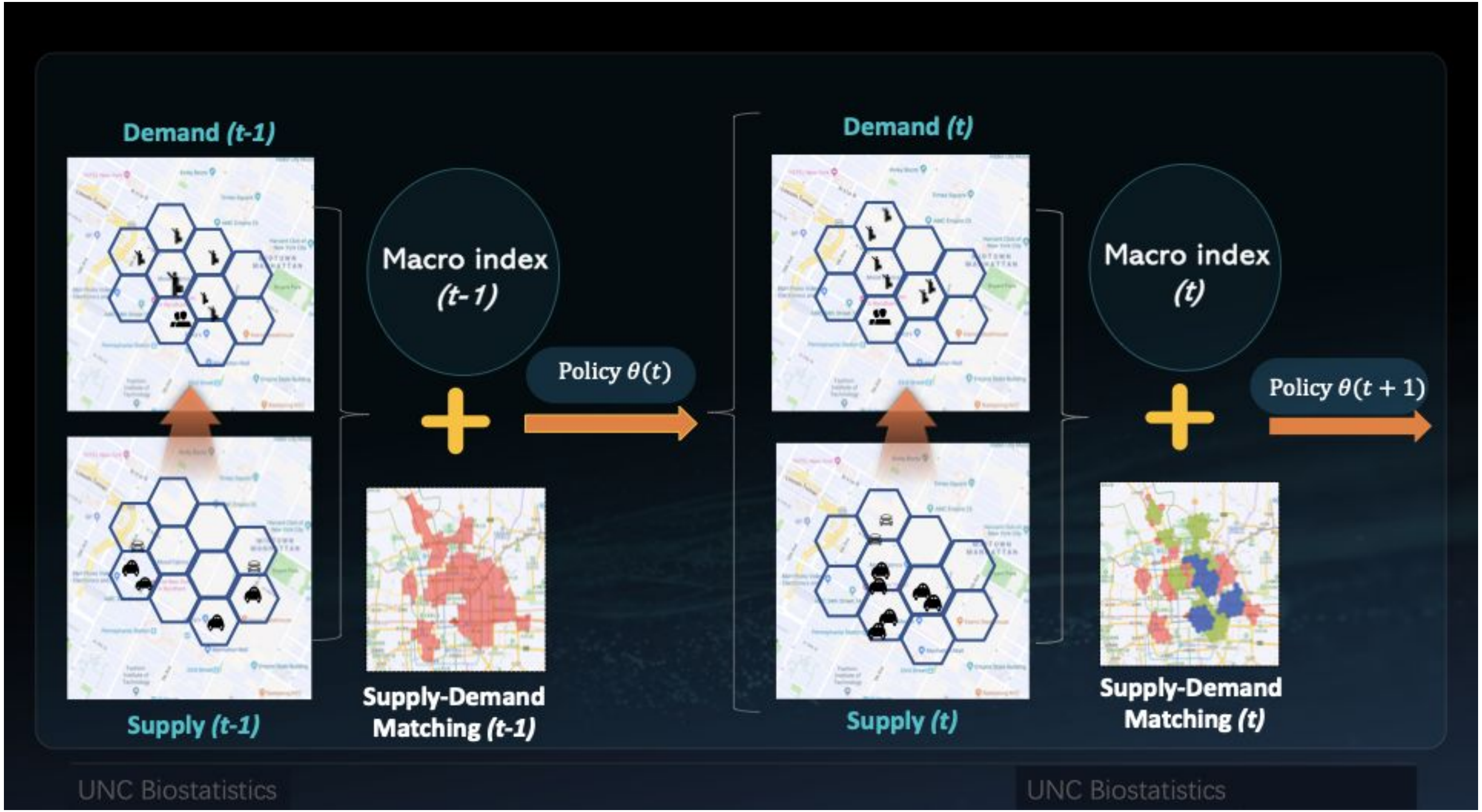
Average driver perception of market balance

$$A_s = \sum_{i,t\in\mathcal{T}} \bar{\mu}_{i,t} \left(\log(\frac{\bar{\mu}_{i,t}}{\bar{v}_{i,t}}) + \log(\frac{M}{N}) \right) \qquad \bar{\mu}_{i,t} = \frac{\tilde{\mu}_{i,t}}{M} \qquad \bar{v}_{i,t} = \frac{\tilde{v}_{i,t}}{N}$$

If M=N

$$A_s = D_{KL}(\bar{\mu}||\bar{v}) \qquad A_d = -D_{KL}(\bar{v}||\bar{\mu})$$

Spatio-temporal GEMs



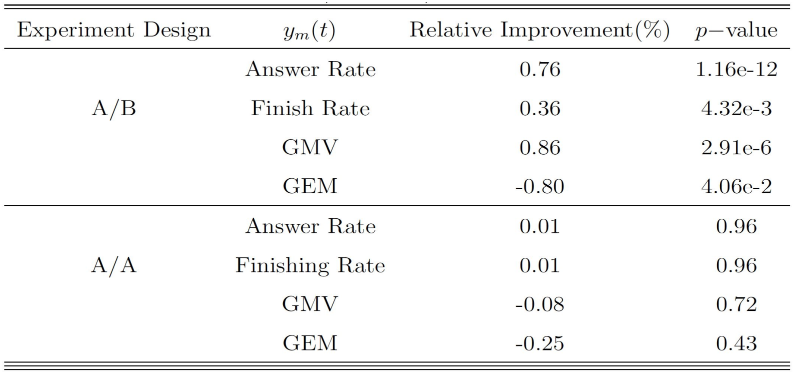
Dynamical Adjustments & Policy Evaluation

Dynamical Adjustments

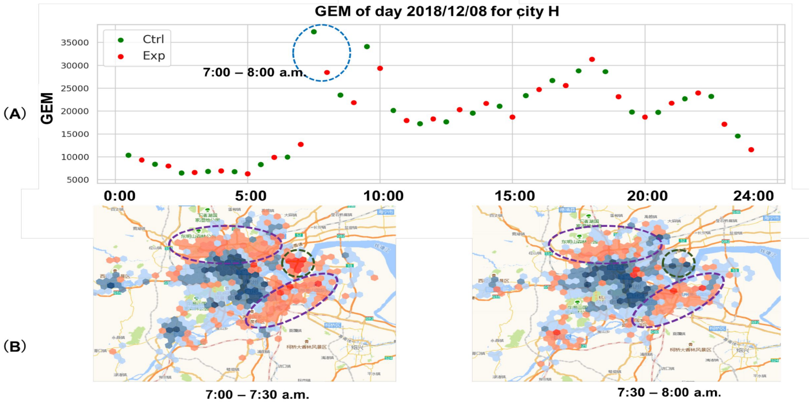
- **Real-Time Incentive Adjustments:**
- **Reallocation of Resources:**
- **Hotspot Mitigation:**

Policy Evaluation

- **Impact Analysis of Pricing Policies:**
- **Fairness & Accessibility Assessment:**
- **Longitudinal Performance Tracking:**
  - Tracks policy effectiveness over time using **dynamic equilibrium graphs**.
  - Provides adaptive policy recommendations based on equilibrium shifts.



Experiment Design	$y_m(t)$	Relative Improvement(%)	$p-$value
A/B	Answer Rate	0.76	1.16e-12
	Finish Rate	0.36	4.32e-3
	GMV	0.86	2.91e-6
	GEM	-0.80	4.06e-2
A/A	Answer Rate	0.01	0.96
	Finishing Rate	0.01	0.96
	GMV	-0.08	0.72
	GEM	-0.25	0.43



Five Key Components of the SDD System

A. Supply-Demand Estimation

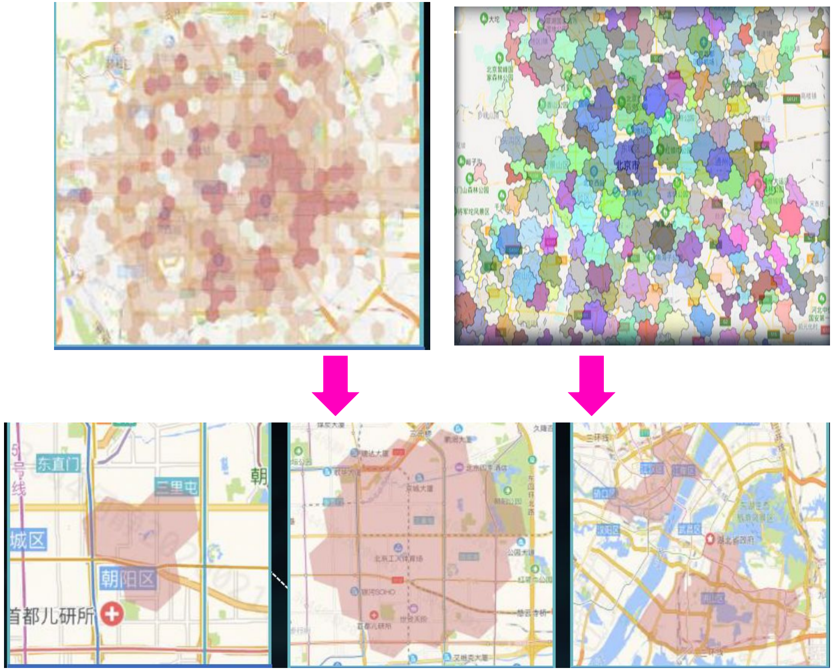
B. Prediction Models for Supply-Demand Forecasting

C. Spatiotemporal Value Calculation

- Graph-Based Equilibrium Metric (GEM) Analysis:
- Reinforcement Learning:

D. Clustering and Demand Hotspot Identification

E. Intelligent Incentive & Pricing Mechanism



(Small-Sized) (Medium-Sized) (Large-Scale)

Hotspots

Supply-Demand Prediction

Geng et al., Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting. *AAAI* 2019.
Yang, R., Dai, R., Tang, X., Zhou, F., and Zhu, H.T. Spatio-temporal prediction of fine-grained origin-destination matrices with applications in ridesharing. *In revision*.
Wang, S., Luo, S.C., and Zhu, H.T. Causal probabilistic spatio-temporal fusion transformers in two-sided ride-hailing markets. *ACM Transactions on Spatial Algorithms and Systems*, 2024, 10, 1 - 18.

Why Is Supply-Demand Forecasting Important?

Supply-Demand Forecasting is crucial across various industries as it helps organizations optimize resources, reduce inefficiencies, and improve decision-making.

1. **Optimized Resource Allocation**
2. **Enhanced Customer Experience**
3. **Cost Reduction and Operational Efficiency**
4. **Data-Driven Decision Making**
5. **Facilitates Dynamic Pricing & Revenue Optimization**
6. **Reduces Risks & Enhances Stability**
7. **Supports Sustainability & Environmental Efficiency**

Industries Where Supply-Demand Forecasting Is Critical

- **E-commerce & Retail:** Managing inventory and customer demand.
- **Transportation & Logistics:** Predicting passenger demand and vehicle availability.
- **Healthcare:** Ensuring medical supply availability and hospital capacity management.
- **Energy Sector:** Balancing power generation with consumption needs.
- **Manufacturing:** Optimizing production schedules and raw material sourcing.

Supply-Demand Prediction in Ride-sharing

The Problem

Predicting the demand-supply distribution

Model

- Multi-modal data fusion
- Complex spatio-temporal patterns

Transfer

- Heterogeneous space among cities
- Heterogeneous feature among tasks

Recognition

- Causal inference
- Model interpretation
- Impact analysis

The Goal

Improve the service quality

Drivers

- Reduce empty driving

Riders

- Intelligent travel guidance
- Less queueing time

Platform

- Fill demand-supply gap
- Recognize the market
- Better dispatching and scheduling

Key Challenges in Supply-Demand Forecasting

Supply-demand forecasting is highly complex due to the interplay of **platform policies, environmental factors, economic conditions, social and policy-driven influences, and infrastructure**. Moreover, **supply is primarily driven by demand**, requiring adaptive forecasting models that consider dynamic and uncertain real-world conditions.

1. Policy & Platform-Driven Challenges

a) Platform Policies

b) Regulation & Compliance Constraints

2. Infrastructure Constraints

a) Transportation & Logistics Bottlenecks

b) Energy & Resource Limitations

c) Digital Infrastructure Gaps

3. Environmental & Random Event Factors

a) Climate Change & Natural Disasters

b) Pandemics & Health Crises.

c) Geopolitical Risks & Global Conflicts

4. Economic & Market Dynamics

a) Inflation & Currency Fluctuations

b) Changing Consumer Behavior & Demand Elasticity

c) Bullwhip Effect in Supply Chains

5. Data & Model Uncertainty in Forecasting

a) Confounding Factors in Predictive Models

b) Data Scarcity & Inaccuracy

c) Feedback Loops & Non-Stationarity

6. Linking Supply & Demand: The Core Challenge

- Infrastructure-Aware Forecasting
- Real-Time Demand Sensing
- Dynamic Supply Adjustments
- Policy-Aware Equilibrium Models

Hierarchical supply-demand forecasting System

A hierarchical supply-demand forecasting system is a structured, multi-level approach that models **macroscopic business indicators** while simultaneously predicting **the joint distribution of supply and demand**. This system integrates **platform policies**, **economic factors, environmental influences, social trends,** and **infrastructure constraints** to provide adaptive, dynamic forecasting in uncertain real-world conditions.

A. Macroscopic Level (Business Indicator Prediction)

- **Objective:** Understand how market equilibrium metric, demand elasticity metrics, and infrastructure utilization index influence key business metrics.

B. Meso Level (Supply-Demand Matching & Policy Integration)

- **Objective:** Understand how policies, infrastructure, and economic conditions affect supply-demand matching.

C. Micro Level (Real-Time Adaptive Forecasting)

- **Objective:** Predict local supply-demand fluctuations in dynamic environments.

Micro Level Supply-Demand Forecasting

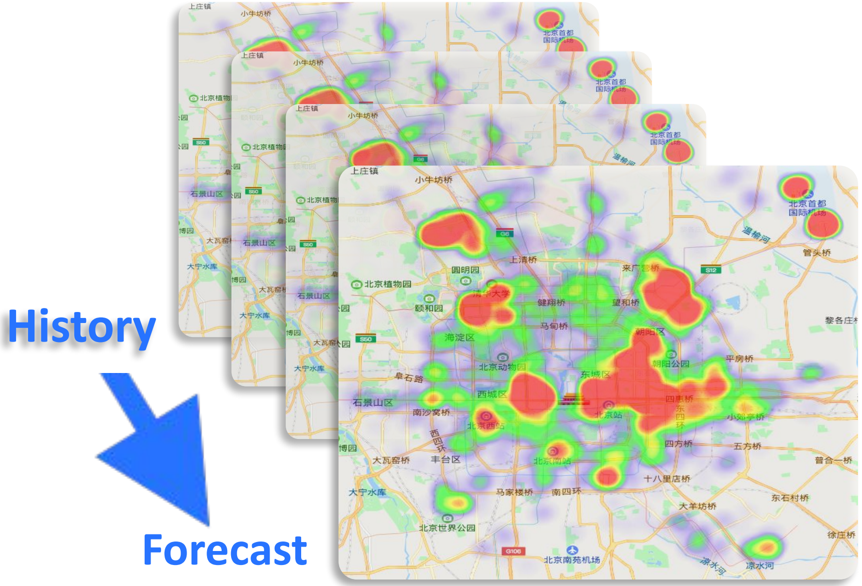
Spatiotemporal Multi-Graph Convolution Network for Ride-Hailing Demand Forecasting

Goal: Predict ride-hailing demand at the region level using historical data
Algorithm 1: Spatiotemporal Multi-Graph Convolution Network **(ST-MGCN)**
Authors: Geng et al.,
Proceeding: AAAI 2019

Spatio-temporal Prediction of Fine-Grained Origin-Destination Matrices with Applications in Ridesharing

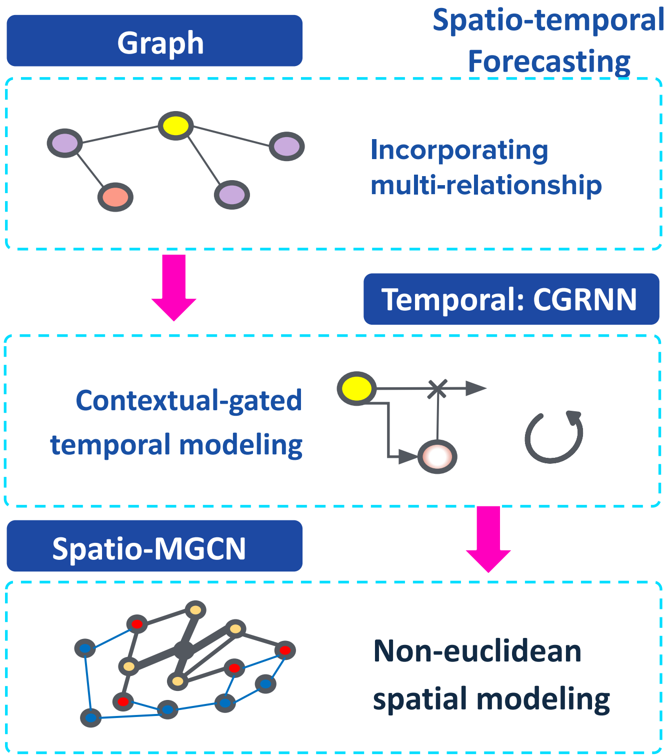
Goal: Accurate OD demand prediction to enhance ridesharing efficiency
Algorithm 2: The Coarseing-Encoder-Decoder network for fine-grained Origin-Destination data **(OD-CED)**
Authors: Yang et al.,
Journal: In revision for Journal of Computational and Graphical Statistics

Algorithm 1: ST-MGCN



History

Forecast



Graph

Spatio-temporal Forecasting

Incorporating multi-relationship

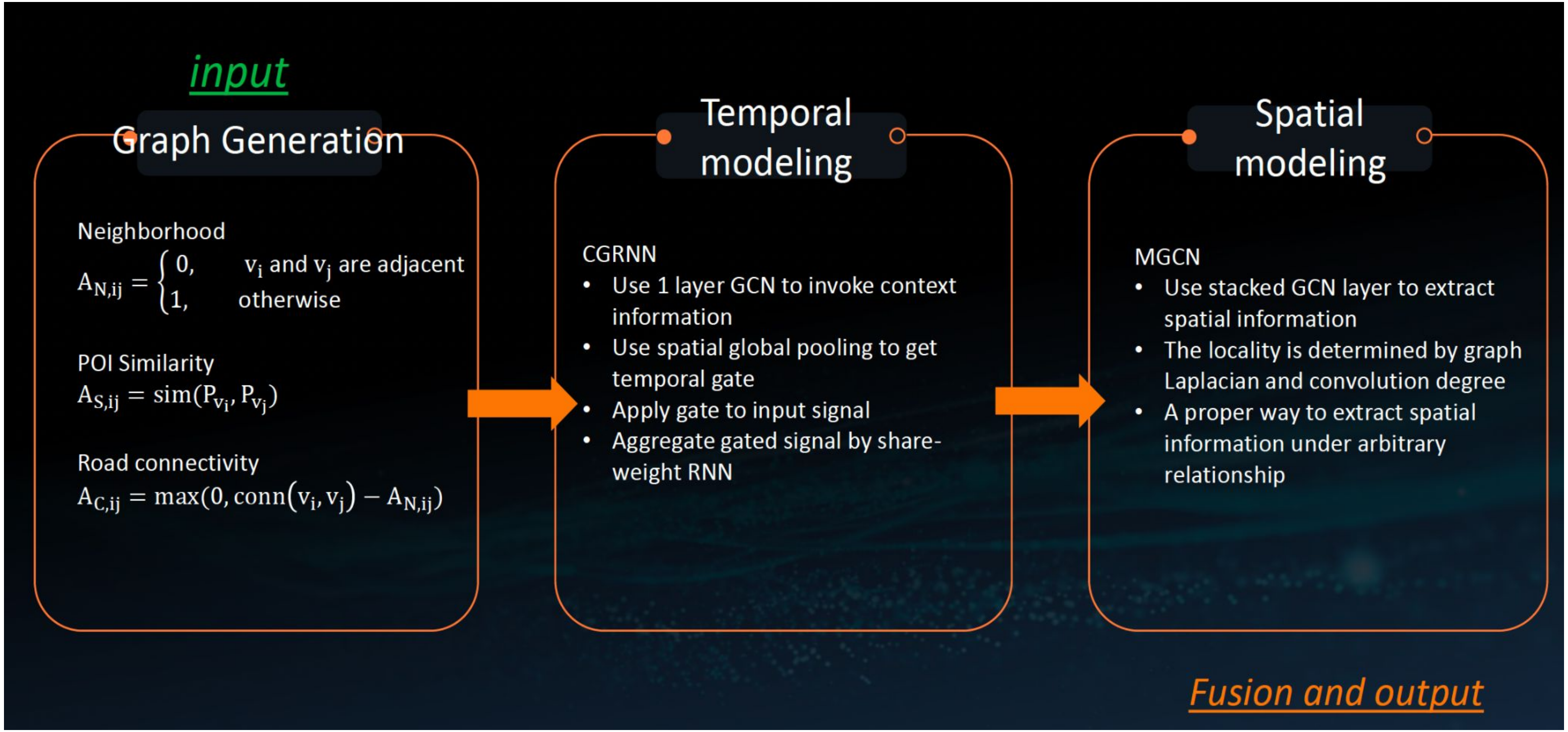
Temporal: CGRNN

Contextual-gated temporal modeling

Spatio-MGCN

Non-euclidean spatial modeling

Algorithm 1: ST-MGCN



input

Graph Generation

Neighborhood

$$A_{N,ij} = \begin{cases} 0, & v_i \text{ and } v_j \text{ are adjacent} \\ 1, & \text{otherwise} \end{cases}$$

POI Similarity

$$A_{S,ij} = \mathrm{sim}(P_{v_i}, P_{v_j})$$

Road connectivity

$$A_{C,ij} = \max(0, \mathrm{conn}(v_i, v_j) - A_{N,ij})$$

Temporal modeling

CGRNN

- Use 1 layer GCN to invoke context information
- Use spatial global pooling to get temporal gate
- Apply gate to input signal
- Aggregate gated signal by share-weight RNN

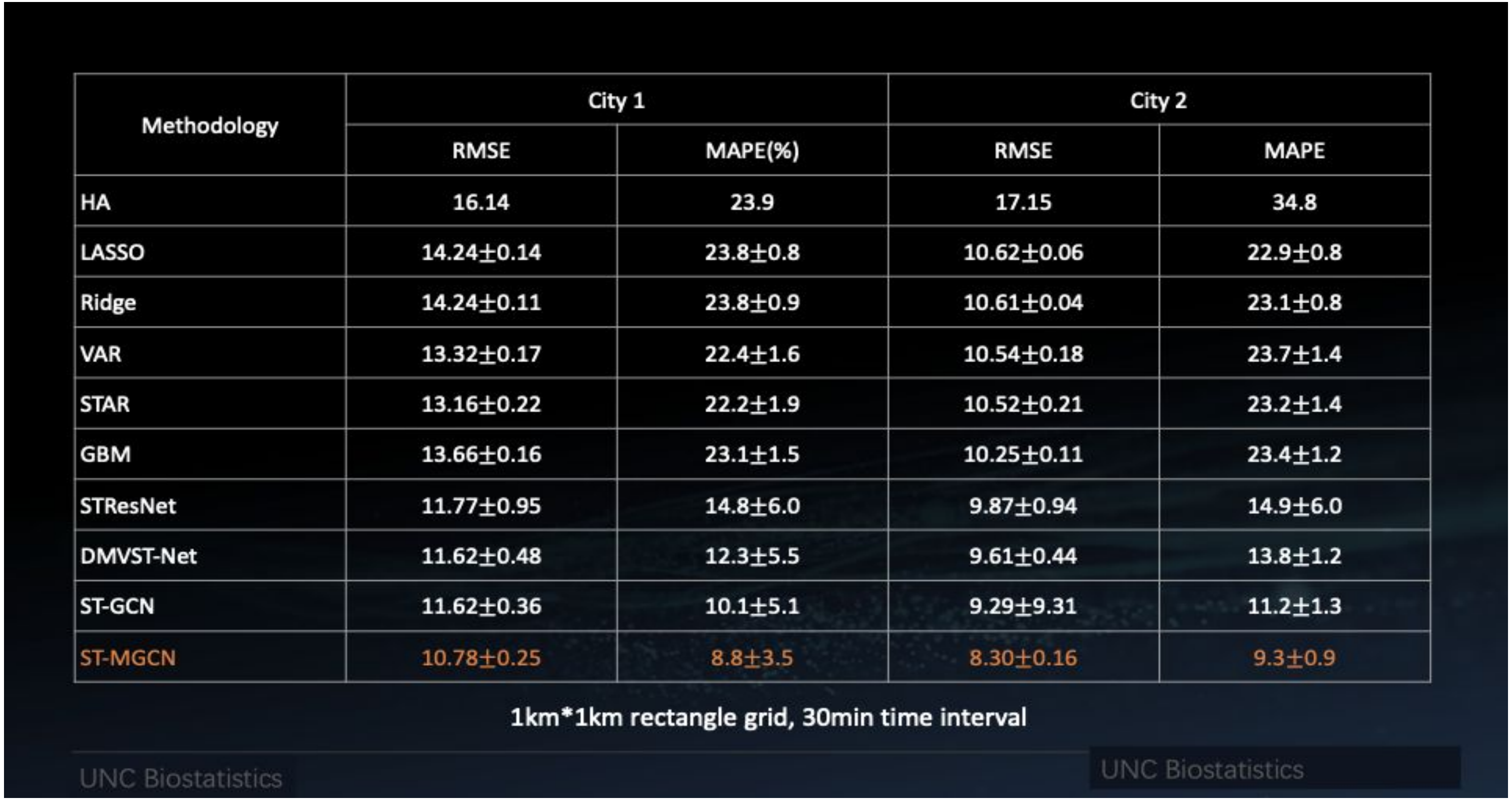
Spatial modeling

MGCN

- Use stacked GCN layer to extract spatial information
- The locality is determined by graph Laplacian and convolution degree
- A proper way to extract spatial information under arbitrary relationship

Fusion and output

Experiment



Methodology	City 1		City 2	
	RMSE	MAPE(%)	RMSE	MAPE
HA	16.14	23.9	17.15	34.8
LASSO	14.24±0.14	23.8±0.8	10.62±0.06	22.9±0.8
Ridge	14.24±0.11	23.8±0.9	10.61±0.04	23.1±0.8
VAR	13.32±0.17	22.4±1.6	10.54±0.18	23.7±1.4
STAR	13.16±0.22	22.2±1.9	10.52±0.21	23.2±1.4
GBM	13.66±0.16	23.1±1.5	10.25±0.11	23.4±1.2
STResNet	11.77±0.95	14.8±6.0	9.87±0.94	14.9±6.0
DMVST-Net	11.62±0.48	12.3±5.5	9.61±0.44	13.8±1.2
ST-GCN	11.62±0.36	10.1±5.1	9.29±9.31	11.2±1.3
ST-MGCN	10.78±0.25	8.8±3.5	8.30±0.16	9.3±0.9

1km*1km rectangle grid, 30min time interval

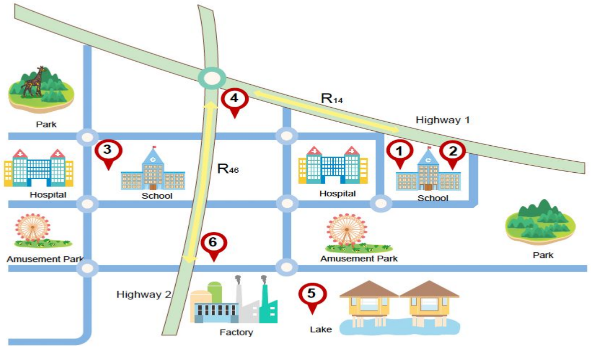
UNC Biostatistics

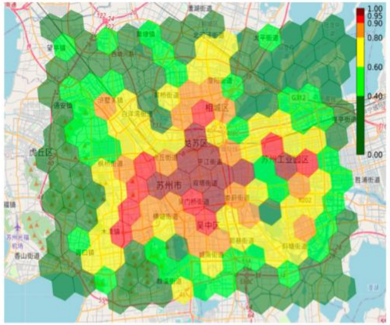
UNC Biostatistics

Algorithm 2: OD-CED

Challenges:

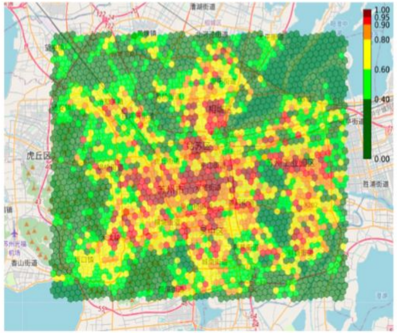
1. **Scalability** – OD matrices grow exponentially with more spatial divisions.
2. **Data Sparsity** – Over 90% of fine-grained OD flows have zero demand.
3. **Semantic & Geographical Dependencies** – Travel demand is influenced by both regional function (e.g., residential vs. commercial) and spatial proximity.





$$\begin{array}{c|ccccccc} & g_1 & g_2 & g_3 & \cdot & g_j & \cdot & g_{183} \\ \hline g_1 & 9 & 0 & 12 & \cdot & \cdot & \cdot & 0 \\ g_2 & 0 & 23 & 0 & \cdot & \cdot & \cdot & 2 \\ g_3 & 1 & 8 & 9 & \cdot & \cdot & \cdot & 4 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 \\ g_i & \cdot & \cdot & \cdot & \cdot & x_{i,j} & \cdot & 6 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 \\ g_{183} & 5 & 3 & 7 & 0 & 0 & 2 & 0 \end{array}$$

(a)



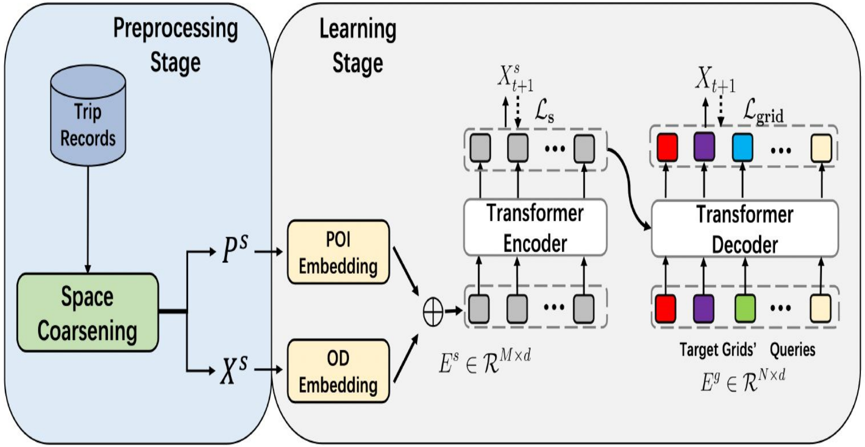
$$\begin{array}{c|ccccccc} & g_1 & g_2 & g_3 & \cdot & g_j & \cdot & g_{2531} \\ \hline g_1 & 1 & 0 & 2 & \cdot & \cdot & \cdot & 0 \\ g_2 & 0 & 0 & 0 & \cdot & \cdot & \cdot & 2 \\ g_3 & 0 & 0 & 0 & \cdot & \cdot & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 \\ g_i & \cdot & \cdot & \cdot & \cdot & x_{i,j} & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 \\ g_{2531} & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{array}$$

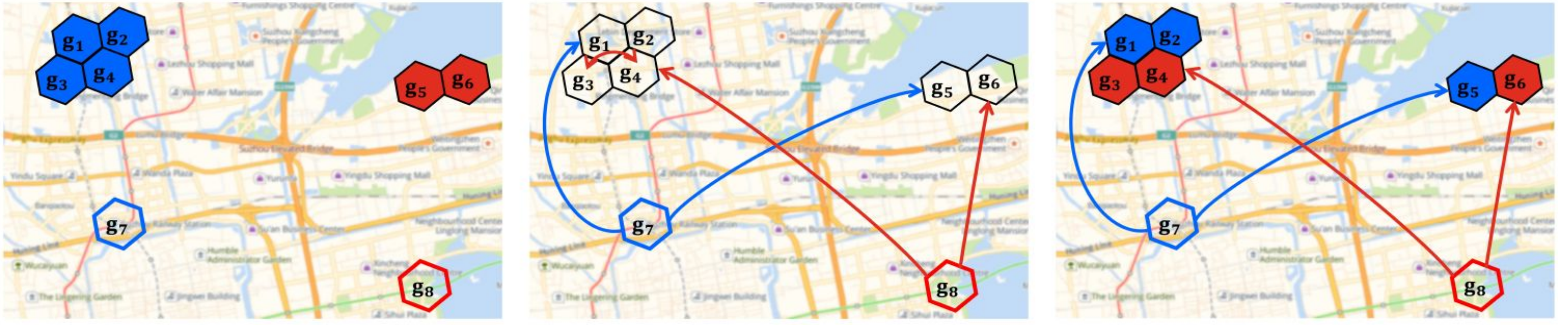
(b)

Algorithm 2: OD-CED

OD-CED Model: A Novel OD Prediction Framework

- **Space Coarsening Module:** Merges fine-grained cells into super-cells to mitigate sparsity.
- **Encoder-Decoder Architecture:** Captures semantic and geographical dependencies effectively.
- **Permutation-Invariant OD Embedding:** Learns robust representations of OD flows.





(a) (b) (c)

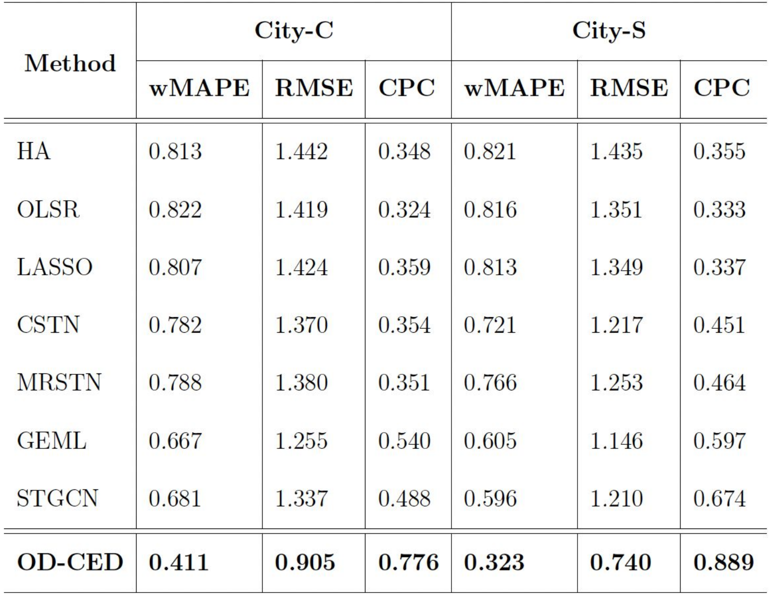
Experiment

Dataset Performance Comparison (City-C & City-S):

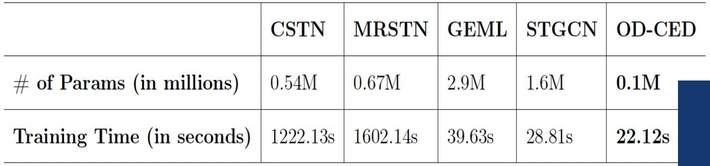
- **City-C:**
  - RMSE reduced from **1.255 (GEML) → 0.905 (OD-CED) (~28% improvement).**
  - wMAPE reduced from **0.667 (GEML) → 0.411 (OD-CED) (~39% improvement).**
- **City-S:**
  - RMSE reduced from **1.146 (GEML) → 0.740 (OD-CED) (~35% improvement).**
  - wMAPE reduced from **0.605 (GEML) → 0.323 (OD-CED) (~47% improvement).**

Training Time Comparison (per epoch on V100 GPU):

- OD-CED: **22.12s**
- STGCN: **28.81s**
- GEML (state-of-the-art): **39.63s**
- CSTN & MRSTN: **1200+ seconds**
- OD-CED is **2x faster than GEML** and **over 50x faster** than CNN-based methods.



Method	City-C			City-S		
	wMAPE	RMSE	CPC	wMAPE	RMSE	CPC
HA	0.813	1.442	0.348	0.821	1.435	0.355
OLSR	0.822	1.419	0.324	0.816	1.351	0.333
LASSO	0.807	1.424	0.359	0.813	1.349	0.337
CSTN	0.782	1.370	0.354	0.721	1.217	0.451
MRSTN	0.788	1.380	0.351	0.766	1.253	0.464
GEML	0.667	1.255	0.540	0.605	1.146	0.597
STGCN	0.681	1.337	0.488	0.596	1.210	0.674
OD-CED	**0.411**	**0.905**	**0.776**	**0.323**	**0.740**	**0.889**



	CSTN	MRSTN	GEML	STGCN	OD-CED
# of Params (in millions)	0.54M	0.67M	2.9M	1.6M	**0.1M**
Training Time (in seconds)	1222.13s	1602.14s	39.63s	28.81s	**22.12s**

Meso Level Supply-Demand Forecasting

Causal Probabilistic Spatio-Temporal Fusion Transformers in Two-Sided Ride-Hailing Markets

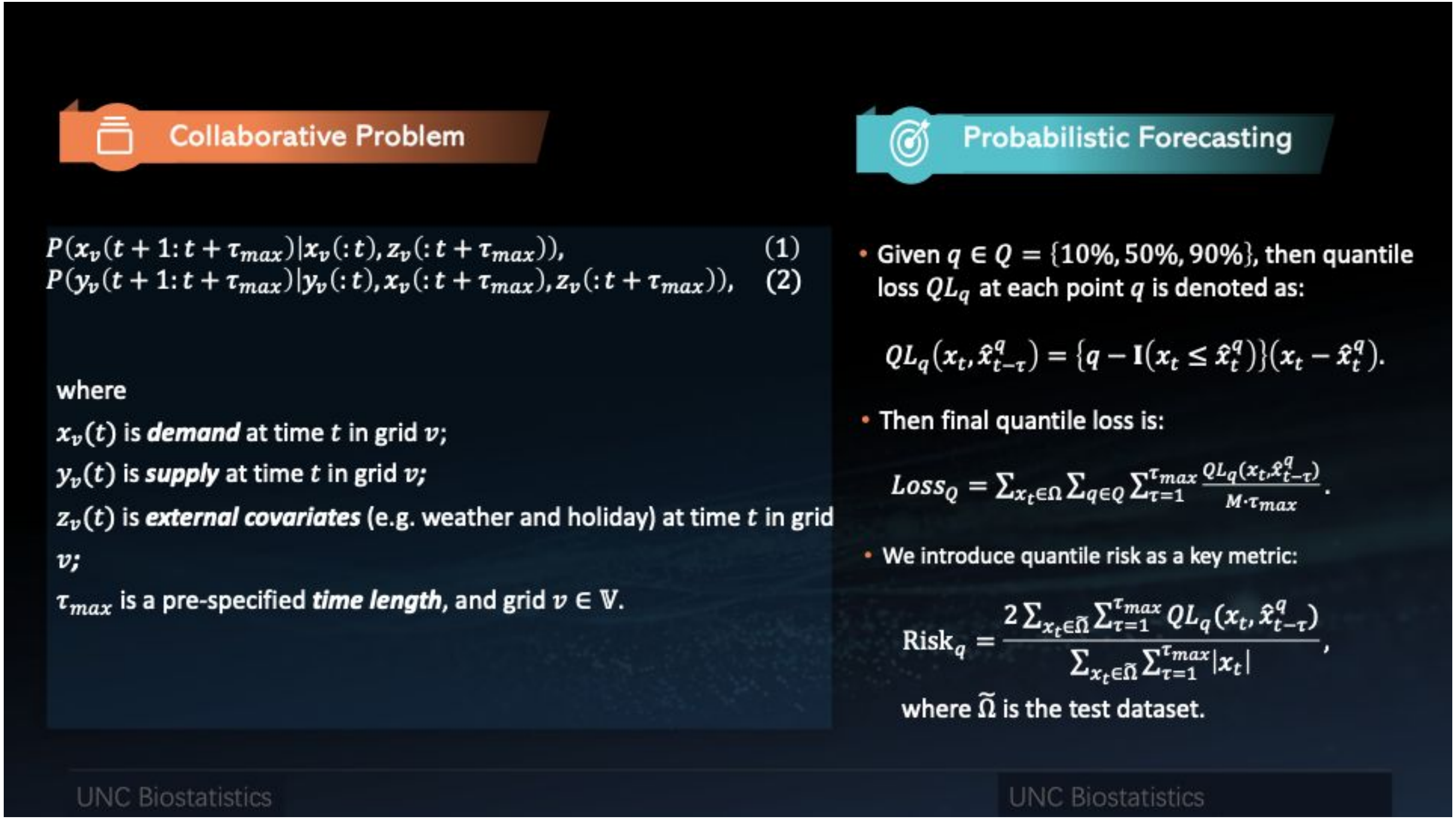
Goal: Predicting supply and demand in ride-hailing platforms using a **causal, interpretable,** and **scalable** forecasting framework

Algorithm 3:, A collaborative causal spatio-temporal fusion transformer **(CausalTrans)**.

Authors: Wang et al.,

Journal: *ACM Transactions on Spatial Algorithms and Systems, 2024*

Algorithm 3: CausalTrans



Collaborative Problem

$$P(x_v(t+1:t+\tau_{max})|x_v(:t), z_v(:t+\tau_{max})), \quad (1)$$
$$P(y_v(t+1:t+\tau_{max})|y_v(:t), x_v(:t+\tau_{max}), z_v(:t+\tau_{max})), \quad (2)$$

where

$x_v(t)$ is ***demand*** at time t in grid v;

$y_v(t)$ is ***supply*** at time t in grid v;

$z_v(t)$ is ***external covariates*** (e.g. weather and holiday) at time t in grid v;

τ_{max} is a pre-specified ***time length***, and grid $v \in \mathbb{V}$.

Probabilistic Forecasting

- Given $q \in Q = \{10\%, 50\%, 90\%\}$, then quantile loss QL_q at each point q is denoted as:

$$QL_q(x_t, \hat{x}^q_{t-\tau}) = \{q - \mathbf{I}(x_t \leq \hat{x}^q_t)\}(x_t - \hat{x}^q_t).$$

- Then final quantile loss is:

$$Loss_Q = \sum_{x_t \in \Omega} \sum_{q \in Q} \sum_{\tau=1}^{\tau_{max}} \frac{QL_q(x_t, \hat{x}^q_{t-\tau})}{M \cdot \tau_{max}}.$$

- We introduce quantile risk as a key metric:

$$\text{Risk}_q = \frac{2\sum_{x_t \in \tilde{\Omega}} \sum_{\tau=1}^{\tau_{max}} QL_q(x_t, \hat{x}^q_{t-\tau})}{\sum_{x_t \in \tilde{\Omega}} \sum_{\tau=1}^{\tau_{max}} |x_t|},$$

where $\tilde{\Omega}$ is the test dataset.

Algorithm 3: CausalTrans

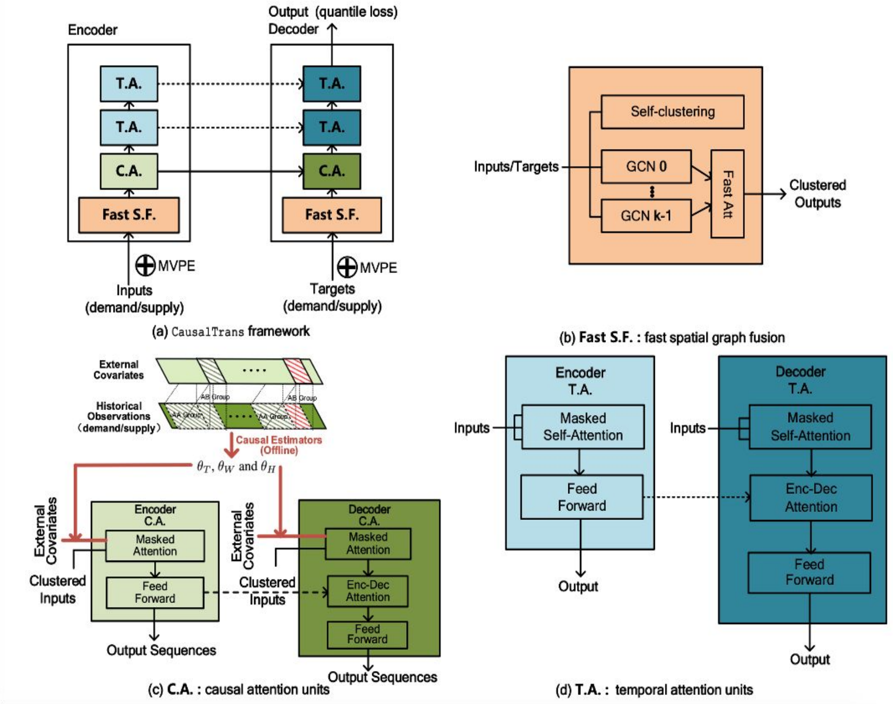
The overview of **CausalTrans** framework:

(a). The framework consists of three essential components: ***Fast S.F.*** (*fast graph spatial fusion*), ***C.A.*** (*causal attention*), and ***T.A.*** (*temporal attention*). Demand and supply are trained separately in sequence.

(b). The ***Fast S.F.*** consists of self-clustering with GAT and fast attention.

(c). The ***C.A.*** applies offline trained causal weights θ to online treatments evaluations.

(d). The ***T.A.*** aims to keep ordering self-attentions.



(a) CausalTrans framework

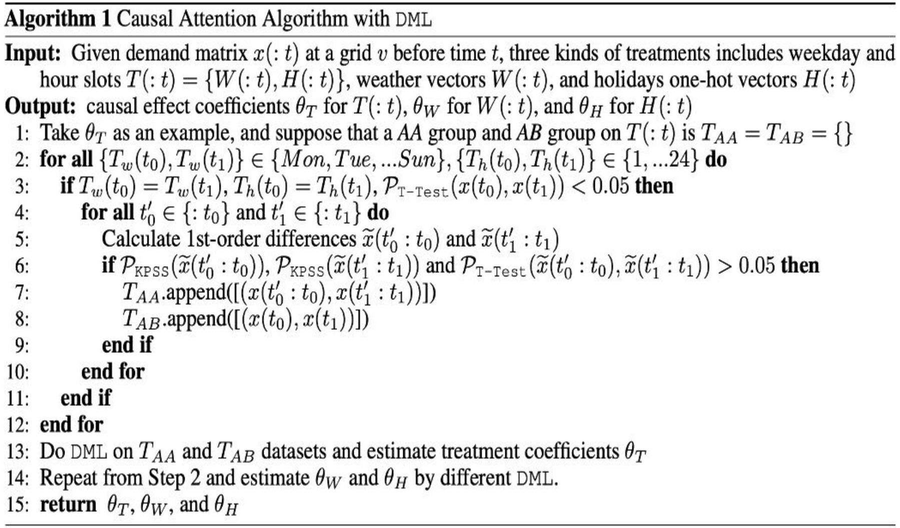
(b) **Fast S.F.** : fast spatial graph fusion

(c) **C.A.** : causal attention units

(d) **T.A.** : temporal attention units

Causal Attention Mechanism

We transfer the weights of external covariates to causal weights by **HTE methods** (e.g. double machine learning).

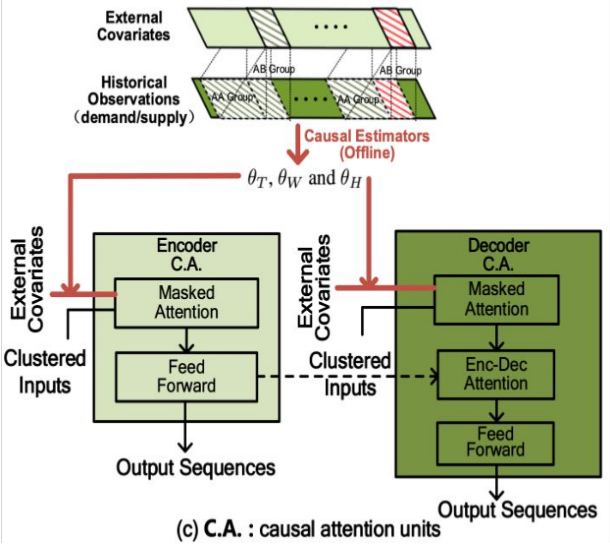


Algorithm 1 Causal Attention Algorithm with DML

Input: Given demand matrix $x(:t)$ at a grid v before time t, three kinds of treatments includes weekday and hour slots $T(:t) = \{W(:t), H(:t)\}$, weather vectors $W(:t)$, and holidays one-hot vectors $H(:t)$

Output: causal effect coefficients θ_T for $T(:t)$, θ_W for $W(:t)$, and θ_H for $H(:t)$

```
1:  Take $\theta_T$ as an example, and suppose that a AA group and AB group on $T(:t)$ is $T_{AA} = T_{AB} = \{\}$
2:  for all $\{T_w(t_0), T_w(t_1)\} \in \{Mon, Tue, ...Sun\}$, $\{T_h(t_0), T_h(t_1)\} \in \{1, ...24\}$ do
3:    if $T_w(t_0) = T_w(t_1)$, $T_h(t_0) = T_h(t_1)$, $\mathcal{P}_{\texttt{T-Test}}(x(t_0), x(t_1)) < 0.05$ then
4:      for all $t_0' \in \{:t_0\}$ and $t_1' \in \{:t_1\}$ do
5:        Calculate 1st-order differences $\widetilde{x}(t_0' : t_0)$ and $\widetilde{x}(t_1' : t_1)$
6:        if $\mathcal{P}_{\texttt{KPSS}}(\widetilde{x}(t_0' : t_0))$, $\mathcal{P}_{\texttt{KPSS}}(\widetilde{x}(t_1' : t_1))$ and $\mathcal{P}_{\texttt{T-Test}}(\widetilde{x}(t_0' : t_0), \widetilde{x}(t_1' : t_1)) > 0.05$ then
7:          $T_{AA}$.append([$(x(t_0' : t_0), x(t_1' : t_1))$])
8:          $T_{AB}$.append([$(x(t_0), x(t_1))$])
9:        end if
10:     end for
11:   end if
12: end for
13: Do DML on $T_{AA}$ and $T_{AB}$ datasets and estimate treatment coefficients $\theta_T$
14: Repeat from Step 2 and estimate $\theta_W$ and $\theta_H$ by different DML.
15: return $\theta_T, \theta_W$, and $\theta_H$
```



(a) causal attention algorithm

step 1: external covariates: weather, holidays and subsidy;

step 2: build various of control groups and treat groups;

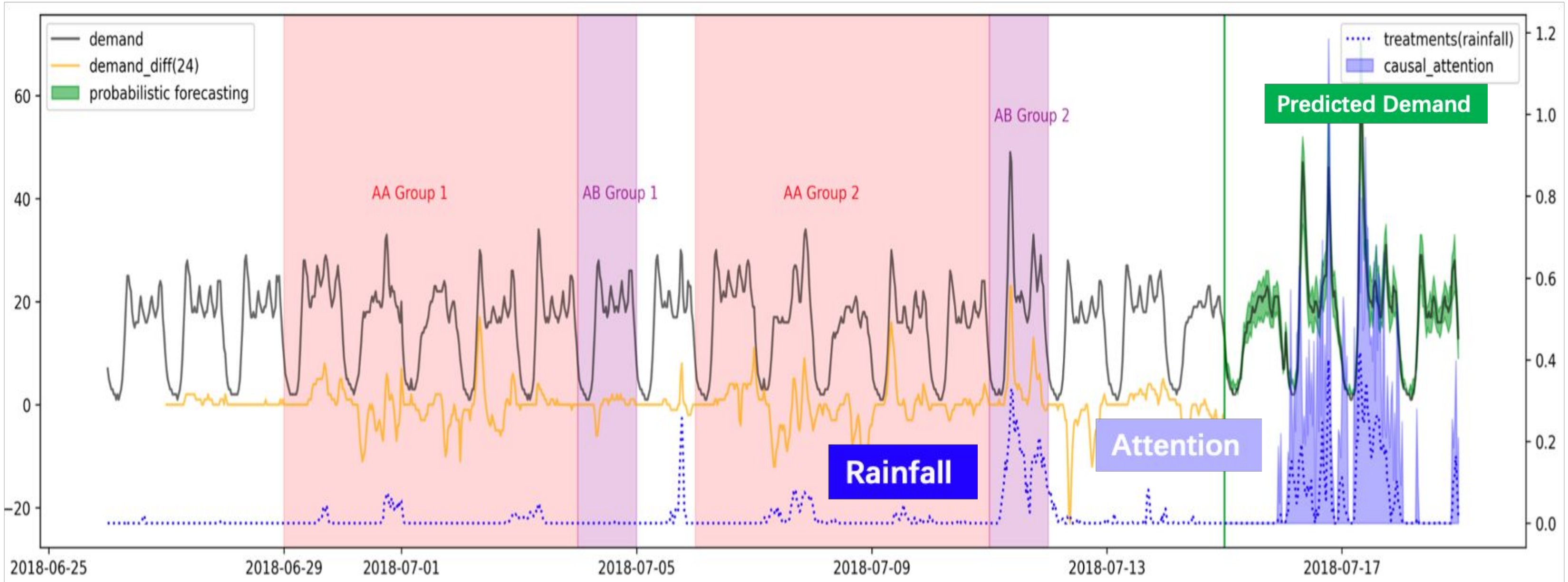
step 3: do *DML* and get causal attention or weights.

(b) how to work in ConvTrans

step 1: offline training causal attention;

step 2: add above weights in multi-head attention

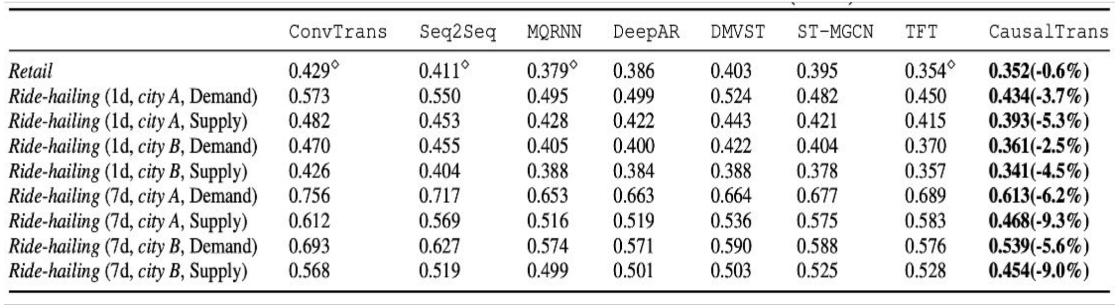
Causal Attention Visualization



- "AA group 1" and "AA group 2" are regarded as comparable contexts;
- "AB group 1" and "AB group 2" is control group and treatment group;
- Do ***DML*** and get causal attention weights.

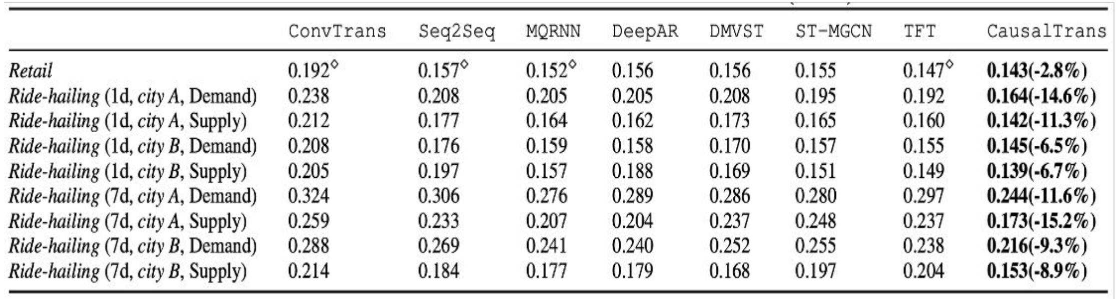
Experiment

(a) Risk_(50%) losses on the retail and ride-hailing datasets.



	ConvTrans	Seq2Seq	MQRNN	DeepAR	DMVST	ST-MGCN	TFT	CausalTrans
Retail	0.429$^{\diamond}$	0.411$^{\diamond}$	0.379$^{\diamond}$	0.386	0.403	0.395	0.354$^{\diamond}$	**0.352(-0.6%)**
Ride-hailing (1d, *city A*, Demand)	0.573	0.550	0.495	0.499	0.524	0.482	0.450	**0.434(-3.7%)**
Ride-hailing (1d, *city A*, Supply)	0.482	0.453	0.428	0.422	0.443	0.421	0.415	**0.393(-5.3%)**
Ride-hailing (1d, *city B*, Demand)	0.470	0.455	0.405	0.400	0.422	0.404	0.370	**0.361(-2.5%)**
Ride-hailing (1d, *city B*, Supply)	0.426	0.404	0.388	0.384	0.388	0.378	0.357	**0.341(-4.5%)**
Ride-hailing (7d, *city A*, Demand)	0.756	0.717	0.653	0.663	0.664	0.677	0.689	**0.613(-6.2%)**
Ride-hailing (7d, *city A*, Supply)	0.612	0.569	0.516	0.519	0.536	0.575	0.583	**0.468(-9.3%)**
Ride-hailing (7d, *city B*, Demand)	0.693	0.627	0.574	0.571	0.590	0.588	0.576	**0.539(-5.6%)**
Ride-hailing (7d, *city B*, Supply)	0.568	0.519	0.499	0.501	0.503	0.525	0.528	**0.454(-9.0%)**

(b) Risk_(90%) losses on the retail and ride-hailing datasets.



	ConvTrans	Seq2Seq	MQRNN	DeepAR	DMVST	ST-MGCN	TFT	CausalTrans
Retail	0.192$^{\diamond}$	0.157$^{\diamond}$	0.152$^{\diamond}$	0.156	0.156	0.155	0.147$^{\diamond}$	**0.143(-2.8%)**
Ride-hailing (1d, *city A*, Demand)	0.238	0.208	0.205	0.205	0.208	0.195	0.192	**0.164(-14.6%)**
Ride-hailing (1d, *city A*, Supply)	0.212	0.177	0.164	0.162	0.173	0.165	0.160	**0.142(-11.3%)**
Ride-hailing (1d, *city B*, Demand)	0.208	0.176	0.159	0.158	0.170	0.157	0.155	**0.145(-6.5%)**
Ride-hailing (1d, *city B*, Supply)	0.205	0.197	0.157	0.188	0.169	0.151	0.149	**0.139(-6.7%)**
Ride-hailing (7d, *city A*, Demand)	0.324	0.306	0.276	0.289	0.286	0.280	0.297	**0.244(-11.6%)**
Ride-hailing (7d, *city A*, Supply)	0.259	0.233	0.207	0.204	0.237	0.248	0.237	**0.173(-15.2%)**
Ride-hailing (7d, *city B*, Demand)	0.288	0.269	0.241	0.240	0.252	0.255	0.238	**0.216(-9.3%)**
Ride-hailing (7d, *city B*, Supply)	0.214	0.184	0.177	0.179	0.168	0.197	0.204	**0.153(-8.9%)**

- Use grid search to optimize hyperparameters;
- ***DeepAR*** outperforms ***Seq2Seq*** and ***MQRNN*** because of Poisson and weather covariates;
- ***CausalTrans*** outperforms other methods primarily due to causal estimator ***DML***;
- ***CausalTrans*** achieves lower losses on supply than demand based on both causal relationship;
- Long-term prediction focuses on unbiased distribution estimation.

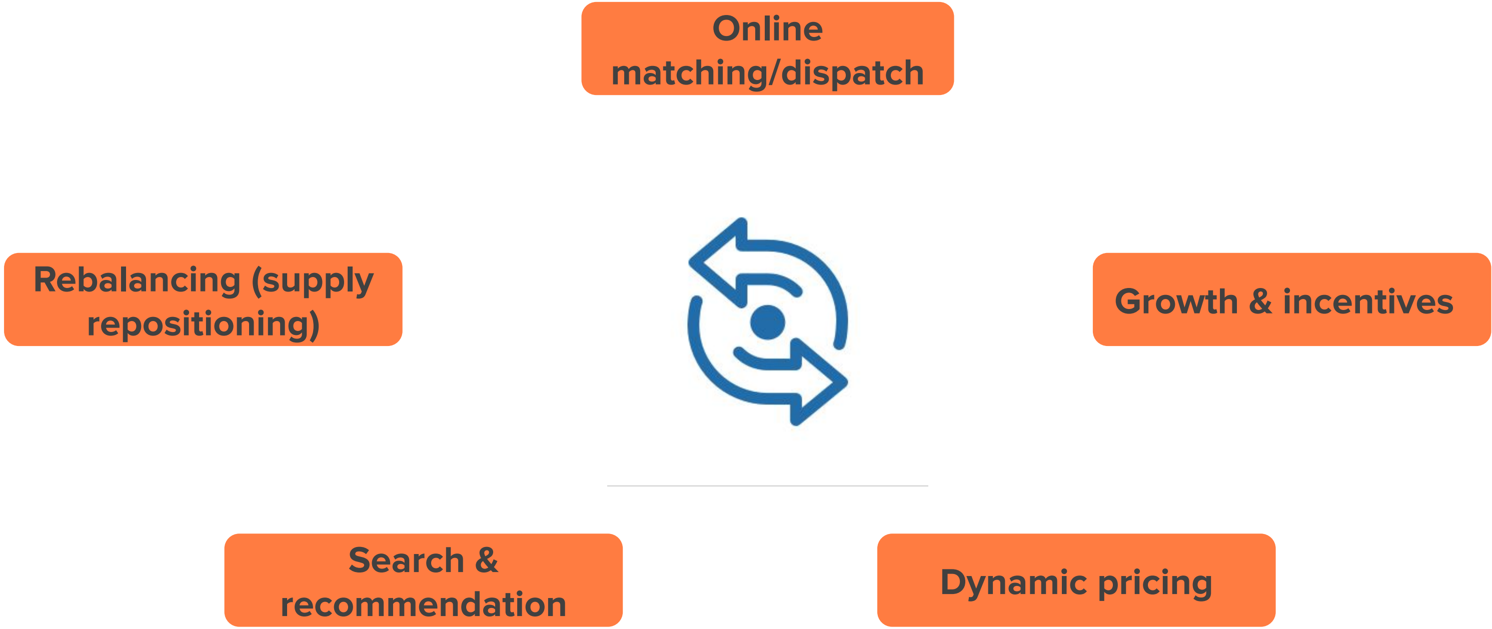
Policy Optimization

Tony Qin

foreva.ai (Ex Lyft, DiDi)

Overview

Core Problems



Online matching/dispatch

Rebalancing (supply repositioning)

Growth & incentives

Search & recommendation

Dynamic pricing

Overview

Heterogeneity

- Ridesharing, vacation rental, retail, jobs, food delivery, consulting works, …
- Core problems in each domain have important unique characteristics.
- Primary focus on ridesharing, but briefly covering other select domains whenever appropriate

Challenges

- Multi-agent, multi-task coordination
- Real-time decision-making
- Fairness

Reinforcement Learning Primer

State, s_t and observation o_t

- $s_t \in S$, state space (discrete or continuous)
- For example, agent’s location
- Fully observable environment: $o_t = s_t$
- Partially observable: $o_t = o(s_t)$, e.g. what agent sees

Action, a_t

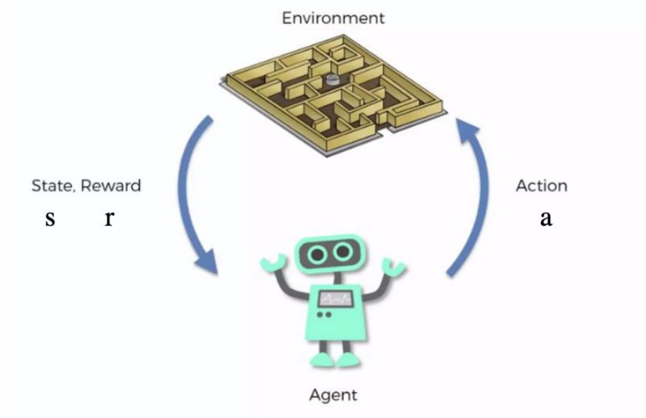
- $a_t \in A$, action space (discrete or continuous)
- For example, going forward/backward, turning left/right

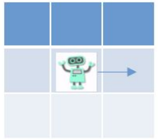
Reward, r_t

- $r_t \sim R(s_t, a_t)$, R is the reward function.
- For example, reaching the exit, reaching some goals, penalty for move (-1 per move)

Markovian property

- $s_{t+1} \sim P(S_{t+1} | s_t, a_t)$, distribution governed by transition



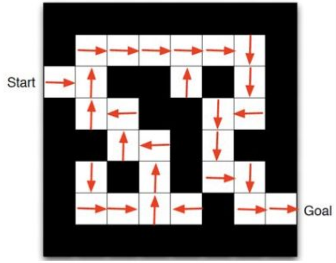


In Maze example, transition is deterministic.

Reinforcement Learning Primer

Policy

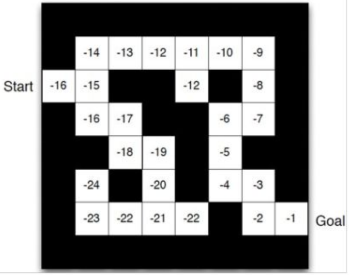
- Governs agent's decision/behavior
- $a \sim \pi(s)$
- Function of state: deterministic or stochastic



Arrow is output of policy: action to be executed at given state

Value function

- Estimate of future long-term reward
- $Q^\pi(s,a) \coloneqq E[r_0 + \gamma r_1 + \gamma^2 r_2 + \cdots | s_0 = s, a_0 = a]$
- $V^\pi(s) \coloneqq E[r_0 + \gamma r_1 + \gamma^2 r_2 + \cdots | s_0 = s]$



Reward =-1 for each move

State values shown

Environment model

- Dynamics that governs the change in state with actions
- Transition probabilities P and reward function R.

Reinforcement Learning Primer

Value-based methods

- TD-learning (V function, on-policy), Q-learning (Q function, off-policy)
- DQN (deep Q-networks)

Policy-based methods

- Policy gradient, advantage (Q-V)
- REINFORCE
- Actor-critic (AC), SAC
- PPO

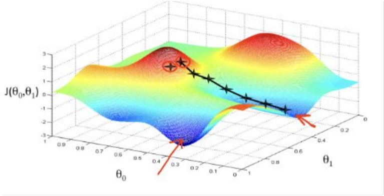
$\max_\pi J(\pi) := V^\pi(s_0)$

Parametrize π as π(θ), then can do stochastic gradient ascent:

$$\theta_{t+1} = \theta_t + \alpha \widehat{\nabla J(\theta_t)}$$

$$\nabla J(\theta) = \sum_s \mu_\pi(s) \sum_a Q^\pi(s,a) \nabla_\theta \pi(a|s,\theta)$$

$\mu_\pi(s)$: on-policy distribution of state s under π



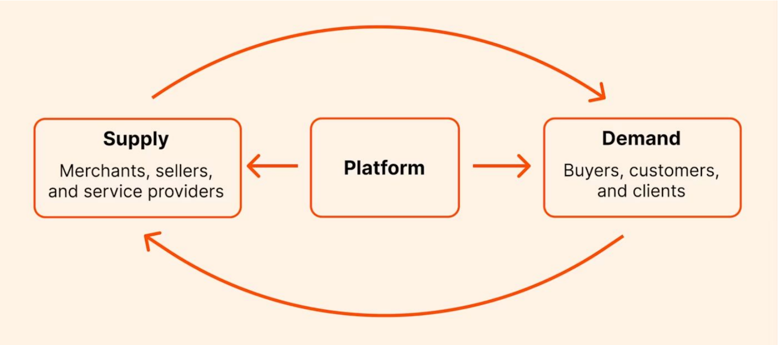
Model-based methods

- Maintains a learning model of the environment, i.e., P and R.

Transactions in a Marketplace

Pairing of supply and demand

- Centralized decision: ridesharing, food delivery
- Search and select: AirBnb, LinkedIn
- Rebalancing: specific to spatiotemporal operations



Supply
Merchants, sellers, and service providers
Platform
Demand
Buyers, customers, and clients

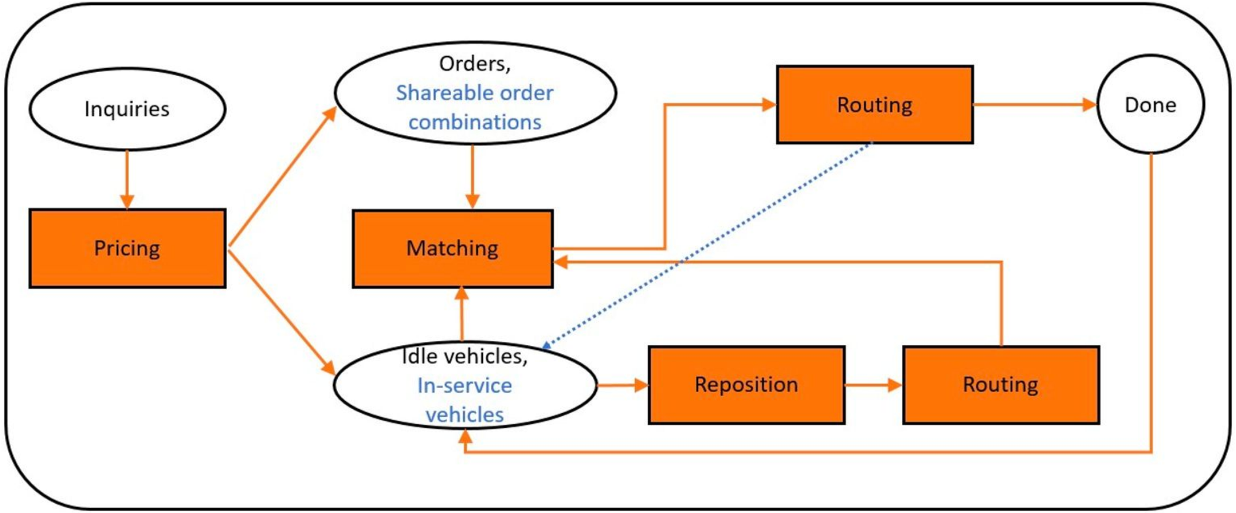
Pricing

- Centralized: set by platform
- Decentralized
  - set by supplies (or service providers): Airbnb, Amazon
  - set via bidding: Angi
  - set by both (buyers set budget, service providers send bids): UpWork
- No price involved: LinkedIn (but platform charges subscription)

Growth & Incentives

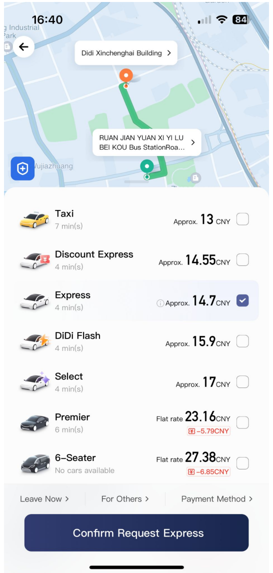
- Maintain both supply and demand populations

Ridesharing System Architecture



Qin, et al., 2024. Reinforcement Learning in the Ridesharing Marketplace. *Synthesis Lectures on Learning, Networks, and Algorithms,* Springer*.*

Online Matching



16:40

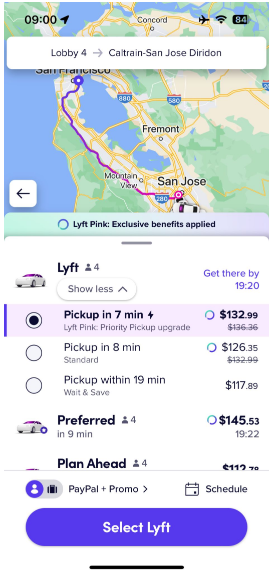
Didi Xinchenghai Building >

RUAN JIAN YUAN XI YI LU BEI KOU Bus StationRoa… >

Taxi 7 min(s)	Approx. 13 CNY	□
Discount Express 4 min(s)	Approx. 14.55CNY	□
Express 4 min(s)	Approx. 14.7CNY	☑
DiDi Flash 4 min(s)	Approx. 15.9CNY	□
Select 4 min(s)	Approx. 17CNY	□
Premier 6 min(s)	Flat rate 23.16CNY -5.79CNY	□
6-Seater No cars available	Flat rate 27.38CNY -6.85CNY	□

Leave Now > For Others > Payment Method >

Confirm Request Express



09:00

Lobby 4 → Caltrain-San Jose Diridon

Lyft Pink: Exclusive benefits applied

Lyft 4 — Show less — Get there by 19:20

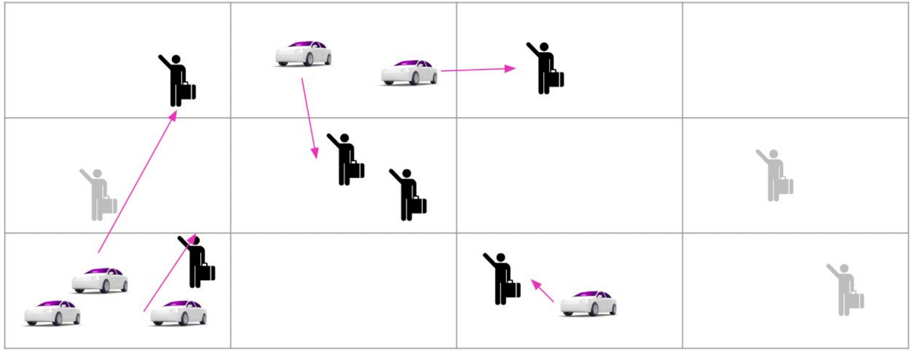
●	Pickup in 7 min Lyft Pink: Priority Pickup upgrade	$132.99 ~~$136.36~~
○	Pickup in 8 min Standard	$126.35 ~~$132.99~~
○	Pickup within 19 min Wait & Save	$117.89

Preferred 4 — in 9 min — $145.53 19:22

Plan Ahead 4 — $112.78

PayPal + Promo > Schedule

Select Lyft



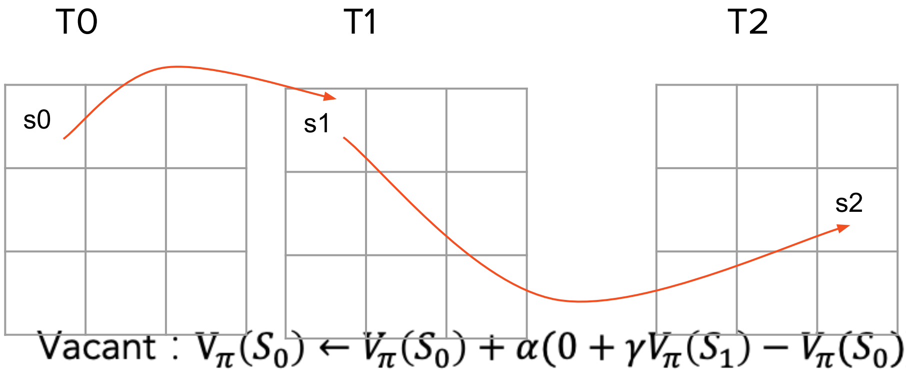
Online Matching | Driver-centric

MDP (Markov Decision Process)

- Modeled around a driver: (location, time, supply-demand context)

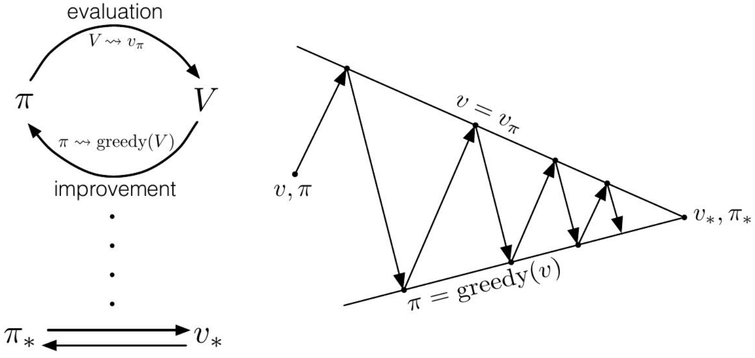
Training

- Offline batch learning: TD learning



Vacant : $V_\pi(S_0) \leftarrow V_\pi(S_0) + \alpha(0 + \gamma V_\pi(S_1) - V_\pi(S_0))$

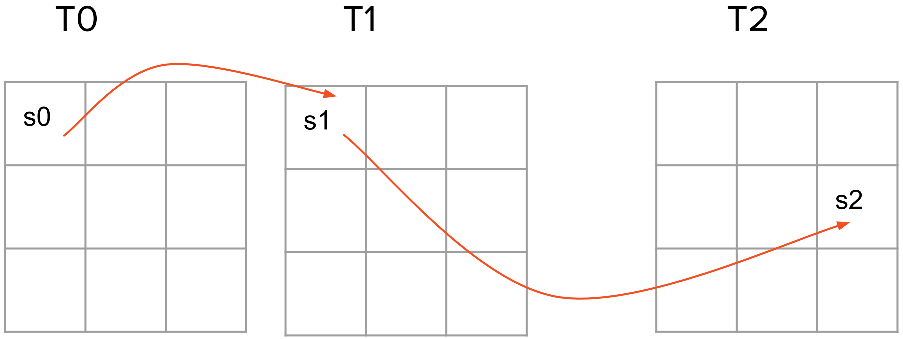
Serving : $V_\pi(S_1) \leftarrow V_\pi(S_1) + \alpha(\hat{r} + \gamma^2 V_\pi(S_2) - V_\pi(S_1))$



[Sutton textbook]

[Xu, et al., 2018; Wang, Qin, Tang, et al., 2018; Qin, et al., 2020]

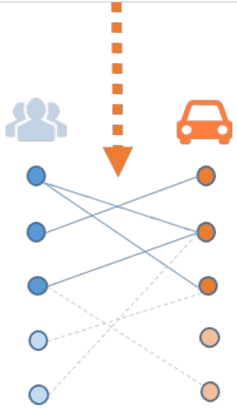
Batch RL for Online Matching



$$\text{Vacant}: V_\pi(S_0) \leftarrow V_\pi(S_0) + \alpha(0 + \gamma V_\pi(S_1) - V_\pi(S_0))$$
$$\text{Serving}: V_\pi(S_1) \leftarrow V_\pi(S_1) + \alpha(\hat{r} + \gamma^2 V_\pi(S_2) - V_\pi(S_1))$$

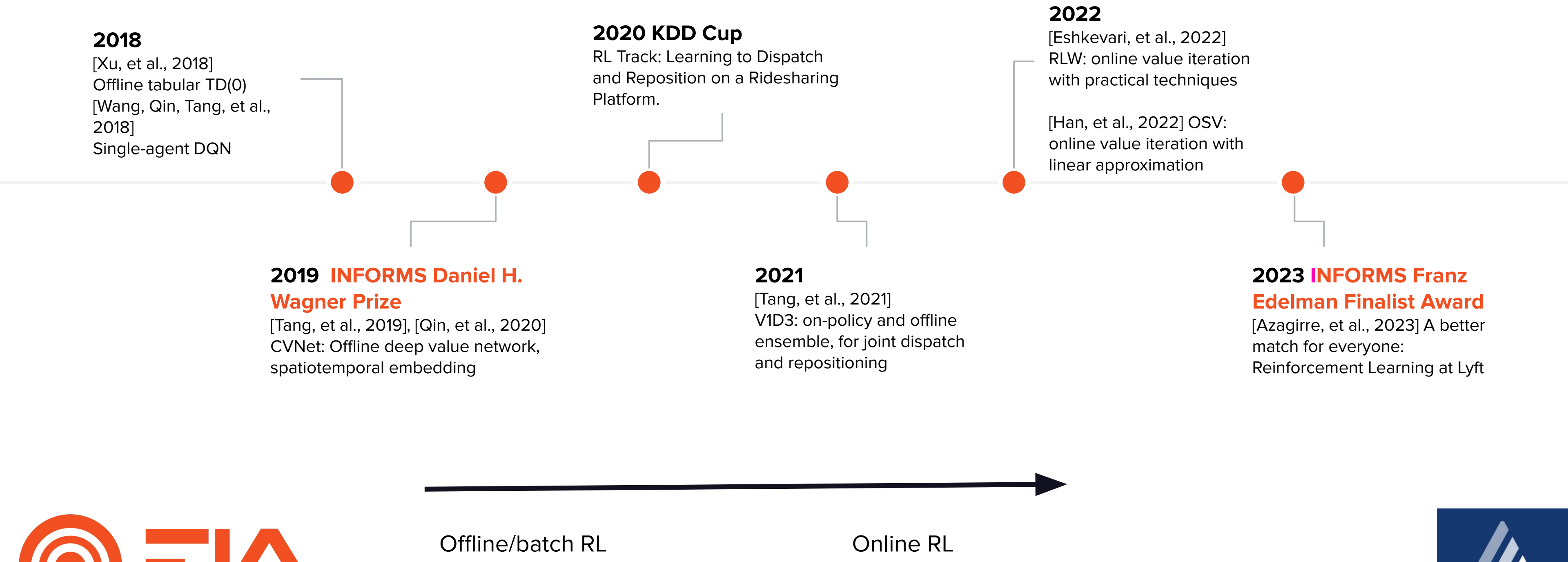
Temporal-difference Learning

$$w_{o^{(i)},x^{(j)}}(V^{\pi_d}) := \overbrace{\hat{p}^{(i)}}^{\text{Est trip price}} + \overbrace{\gamma^{(\hat{\tau}_o^{(i)}+\hat{\tau}_e^{(i)})} V^{\pi_d}(g({l_d}^{(i)}, {t_d}^{(i)}))}^{\text{Discounted value of destination}} - \overbrace{V^{\pi_d}(s(x^{(j)}))}^{\text{Value of origin}}$$



[Xu, et al., 2018; Wang, Qin, Tang, et al., 2018; Qin, et al., 2020]

Trend



2018
[Xu, et al., 2018]
Offline tabular TD(0)
[Wang, Qin, Tang, et al., 2018]
Single-agent DQN

2019 INFORMS Daniel H. Wagner Prize
[Tang, et al., 2019], [Qin, et al., 2020]
CVNet: Offline deep value network, spatiotemporal embedding

2020 KDD Cup
RL Track: Learning to Dispatch and Reposition on a Ridesharing Platform.

2021
[Tang, et al., 2021]
V1D3: on-policy and offline ensemble, for joint dispatch and repositioning

2022
[Eshkevari, et al., 2022]
RLW: online value iteration with practical techniques

[Han, et al., 2022] OSV: online value iteration with linear approximation

2023 INFORMS Franz Edelman Finalist Award
[Azagirre, et al., 2023] A better match for everyone: Reinforcement Learning at Lyft

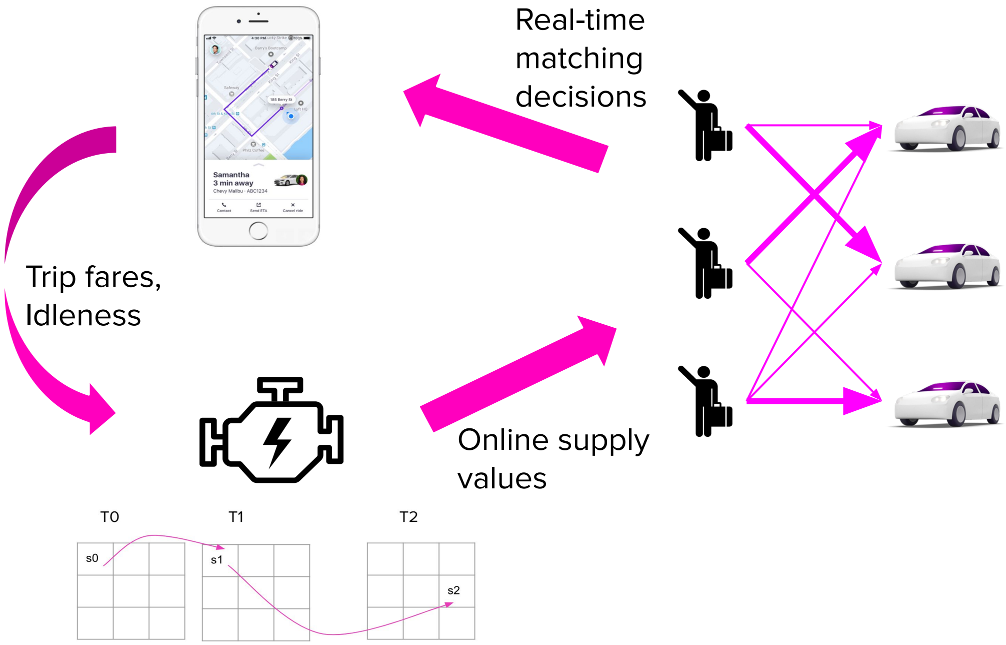
Offline/batch RL

Online RL

Online RL @Lyft

Online value iteration

- Learning driver values online
- Generate real-time matching decisions



Real-time matching decisions

Trip fares, Idleness

Online supply values

Learning the driver values online and on-policy

First full-scale industry deployment of an online RL method

A better match for everyone: Reinforcement Learning at Lyft. Xabi Azagirre, Akshay Balwally, Guillaume Candelli, Nicholas Chamandy, Benjamin Han, Alona King, Hyungjun Lee, Martin Loncaric, Sebastien Martin, Vijay Narasiman, Zhiwei (Tony) Qin, Baptiste Richard, Sara Smoot, Sean Taylor, Garrett van Ryzin, Di Wu, Fei Yu, Alex Zamoshchin. INFORMS Journal on Applied Analytics. 2023.

Online Matching | Online RL

How does online RL differ from batch RL in practice?

- Algorithm: needs to handle more uncertainty since trips may not have fully completed
- Engineering: high performance system to meet the high throughput requirement of a large-scale rideshare platform

Algorithmic techniques

- Expectation-based value updates
- Reward smoothing
- Assignment graph edge standardization
- Real-time adaptive graph pruning
- ADAM for value updates
  - RMSProp: adapt to the variance, magnitude, and frequency of updates
- Linear factorization and sparse coding for value approximation
  - Geo and time features

DiDi [Eshkevari et al., 2022]

Lyft [Han et al., 2022]

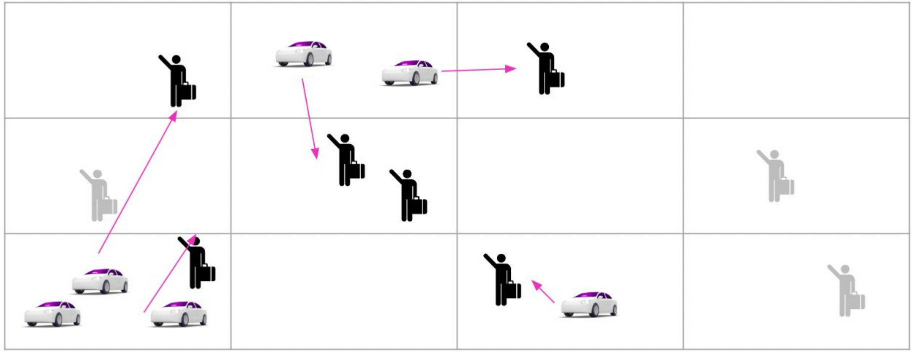
Online Matching | System-centric

Global state context

- Supply and demand over all grid cells
- Other relevant information: waiting time, intents

Action space

- Combinatorial by nature
- Remedy: serialize the assignments



Online Matching | System-centric

State space

- Current time
- Vehicles status: # vehicles with destination d and particular remaining ETA
- Rides status: # riders requesting rides from o to d

Action space

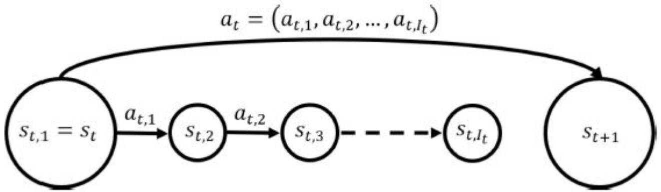
- Match a driver to a request, reposition a driver, idle (“do nothing”)

Reward

$$c(s_{t,i}, a_{t,i}) = \begin{cases} c_t^f(o, d, \eta), & \text{if action } a_{t,i} \text{ implies car-passenger matching,} \\ -c_t^e(o, d), & \text{if action } a_{t,i} \text{ implies empty-car routing.} \end{cases}$$

Sequential decision-making process

- System action at t, a_t
- Decompose into a sequence of atomic actions, each addressing a single available vehicle
- **Atomic action**, a_ti = feasible “trip” (o_ti, d_ti)
- After all available vehicles have been covered, system state s_t -> s_t+1 (random request arrivals, time advance, idle drivers)



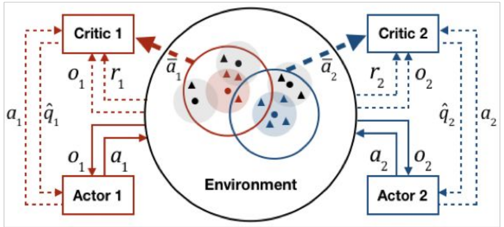
[Feng et al., 2021]

Learning

- PPO objective
- MC rollout for advantage estimation

Online Matching | Multi-agent RL

Mean-field multi-agent RL



[Li et al. 2019]

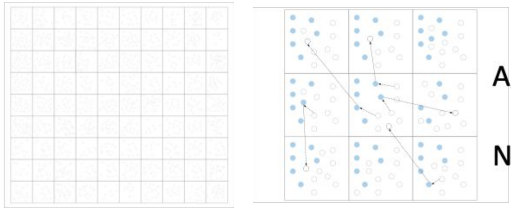
Agent: vehicle

Joint action: mean action of neighborhood

Global reward

Multi-agent RL with KL-divergence regularization

$$\min L = \| Q(s,a) - \left[r + \gamma \bar{Q}(s',a')\right] \|_2 + \lambda D_{KL}$$

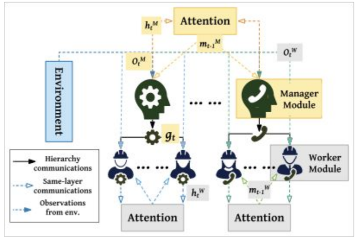


[Zhou et al. 2019]

Agent: vehicle

No explicit communication

Hierarchical Multi-agent RL



[Jin et al. 2019]

Agent: grid cell(s)

Explicit communication

Online Matching | ADP

Approximate DP

Objective: $$\max_{\pi\in\Pi} E[\sum_{t=0}^{T} \gamma^{t} C(S_{t}, X_{t}^{\pi}(S_{t}))|S_{0}]$$

$$\left.\begin{array}{ll}\text{Supply constraints:} & \sum_{d\in D(a)} x_{tad} = R_{ta},\ \forall\, a \in A \\ \text{Demand constraints:} & \sum_{a\in A(d)} x_{tad} \le D_{td},\ \forall\, d \in D \\ \text{Non-negativity:} & x_{tad} \ge 0,\ \forall\, a, d\end{array}\right\} \chi_t$$

$C_t(S_t, x_t) = \sum_{a,d} c_{tad} x_{tad}$ is the reward function, where c_{tad} is the profit (or revenue) of matching a driver at a to a request over route d at time t.

$X_t^{\pi}(S_t)$ = a function that determines x_t given S_t, i.e. the solver for the assignment problem

Approximate DP

- References: [Simao, et al., 2009] and [Al Kanj, et al., 2020]
- Bellman equation for the optimal policy X*

$$X_t^*(S_t) = \arg \max_{x_t \in \chi_t} (C_t(S_t, x_t) + \gamma E[V_{t+1}(S_{t+1})|S_t, x_t]$$

- Post-decision state

$$\begin{tikzcd} S_t \arrow[r, "x_t"'] & S_t^X \arrow[r, "D_t"'] & S_{t+1} \end{tikzcd}$$

$$X_t^*(S_t) = \arg \max_{x_t \in \chi_t} (C_t(S_t, x_t) + \gamma V_t^X(S_t^X)),$$

where $V_t^X(S_t^X) = E[V_{t+1}(S_{t+1})|S_t^X]$, and $V_t(S_t) = \max_{x_t \in \chi_t} (C_t(S_t, x_t) + \gamma V_t^X(S_t^X))$.

Dual-based Forward-looking Value Functions

- Marginal values as decomposition of the system-level forward-looking value

$$V_t^X(S_t^X) \approx \bar{V}_t^n(S_t^X) = \sum_{a'} \bar{v}^n_{t'(a,d),a'} \sum_{a,d} \delta_{a'}(a,d)x_{tad} = \sum_{a,d} \bar{v}^n_{t',a'} x_{tad}$$

$\bar{v}^n_{t',a'}$ represents the marginal cumulative value of a driver in a’ from time t’ over a future horizon.

- Algorithm for steps 1 to T within the n-th episode of dispatch

$$\max_{x_t \in \mathcal{X}_t} \sum_{a,d} (c_{tad} + \gamma \bar{v}^{n-1}_{t',a'})x_{tad} .$$

Solve step-t matching problem.

$$\bar{v}^n_{t,a} \leftarrow (1 - \alpha_n)\bar{v}^{n-1}_{t,a} + \alpha_n u_{t,a}$$

Update value function at (t,a) using supply dual variables $u_{t,a}$ via TD-like model-free updates.

Online Matching | Decentralized

Decentralized mechanism

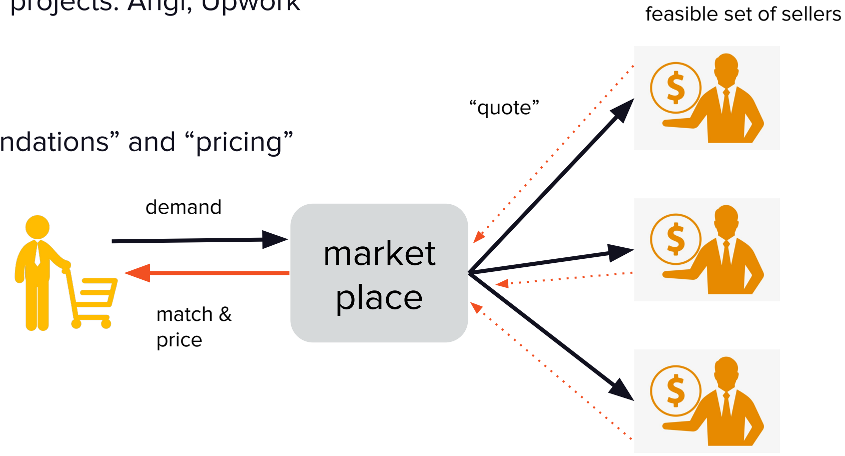
- Demand broadcasting
  - Platform broadcasts the demand to multiple service providers.
  - Service providers bid for the request: by price or by acceptance speed.

Applications

- Some ridesharing platforms (mostly in the early years of this vertical)
- Handyman or consulting projects: Angi, Upwork

Variations

- See “search & recommendations” and “pricing”



Online Matching | Demand Broadcasting

Ridesharing case study

- Zhang et al. *A Taxi Order Dispatch Model Based on Combinatorial Optimization* (KDD 2017)

Motivations

- Nearest-driver matching: ignores global optimization
- **Global Success Rate**: Maximizing completed rides, not just immediate matches.
- **User Experience**: Reducing wait times (dispatch time) and cancellations.

Technical highlights

- **Driver Acceptance Prediction**: Logistic Regression (LR) model estimates acceptance probabilities.
- **Order Dispatch as a Constrained Optimization Problem**:
  - Orders are assigned to multiple drivers.
  - The first driver to accept wins the order.
  - Hill-Climbing Algorithm optimizes the global success rate.

Rebalancing (Supply Repositioning)

Motivation

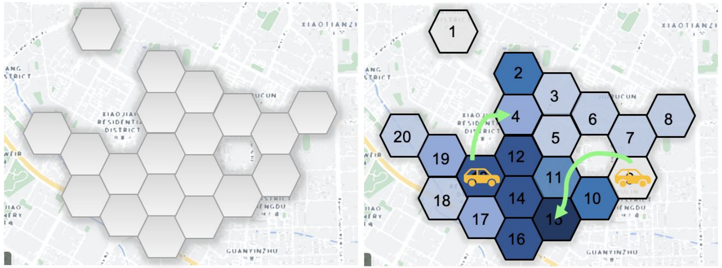
- To proactively relocate idle vehicles to improve individual or system-level income performance

Driver perspective

- Virtual “ride” with platform guidance
- Usually not a long trip: driver acceptance

System perspective

- Intervention of supply distribution over a grid system



(a) (b)

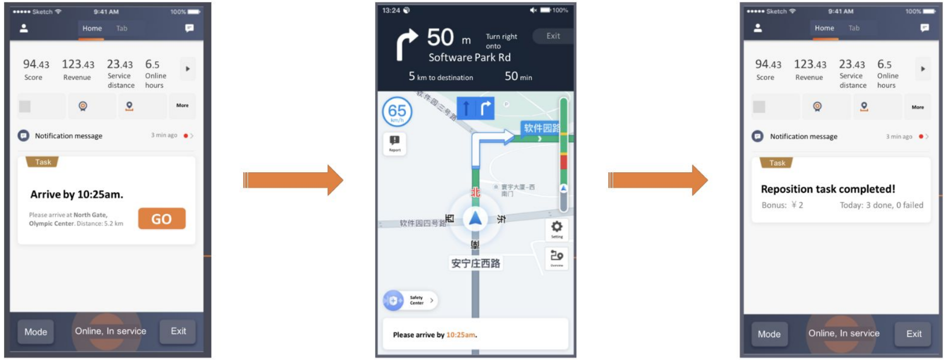
[Wei et al., 2023]

Idle vehicle

Task Received

Reposition

Task Completed



94.43 Score | 123.43 Revenue | 23.43 Service distance | 6.5 Online hours

Notification message 3 min ago

Arrive by 10:25am.

Please arrive at North Gate, Olympic Center. Distance: 5.2 km

50 m Turn right onto Software Park Rd

5 km to destination 50 min

Please arrive by 10:25am.

94.43 Score | 123.43 Revenue | 23.43 Service distance | 6.5 Online hours

Notification message 3 min ago

Reposition task completed!

Bonus: ￥2 Today: 3 done, 0 failed

Illustrations of a driver repositioning assistant

[Jiao et al., 2021]

Rebalancing (Supply Repositioning)

Driver-centric formulation

- Each vehicle executes repositioning independently.

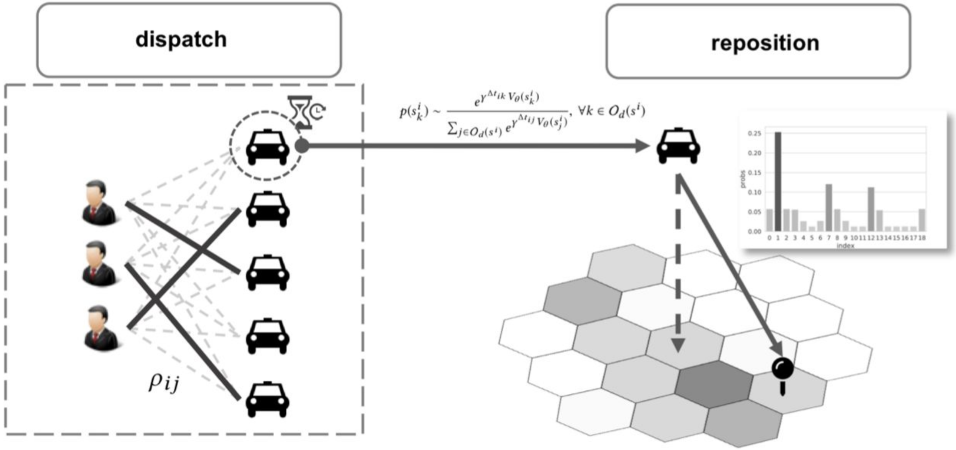
State space

- Can share with the dispatch/matching case

Action space

- Neighboring cells in a grid-cell system

$$p(s_k^i) \sim \frac{e^{\gamma^{\Delta t_{ik}} V_\theta(s_k^i)}}{\sum_{j \in \mathcal{O}_d(s^i)} e^{\gamma^{\Delta t_{ij}} V_\theta(s_j^i)}}, \ \forall k \in \mathcal{O}_d(s^i)$$



dispatch
reposition
$p(s_k^i) \sim \frac{e^{\gamma^{\Delta t_{ik}} V_\theta(s_k^i)}}{\sum_{j \in O_d(s^i)} e^{\gamma^{\Delta t_{ij}} V_\theta(s_j^i)}}, \forall k \in O_d(s^i)$
ρ_{ij}
0.25
0.20
0.15
0.10
0.05
0.00
probs
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
index

Tang, X., Zhang, F., Qin, Z., Wang, Y., Shi, D., Song, B., Tong, Y., Zhu, H. and Ye, J., 2021, August. Value Function is All You Need: A Unified Learning Framework for Ride Hailing Platforms. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining* (pp. 3605-3615).

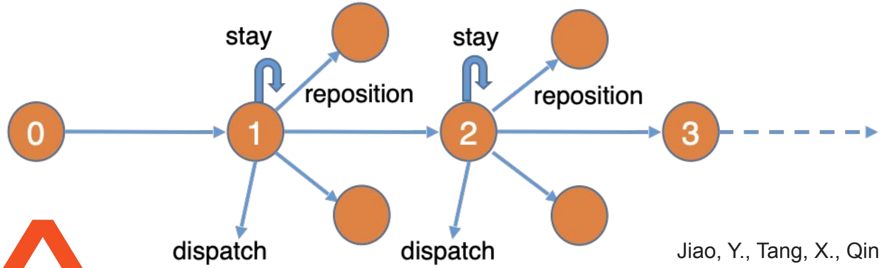
Rebalancing (Supply Repositioning) | Taxi Routing

Dynamics

- Driver performs repositioning upon idling, incurring a non-positive cost.
- During repositioning, the platform can assign an order to the driver at any time.
- The driver can also stay at the current grid cell.

Environment model

- At state s, probability of being dispatched: $P^{(s)}{}_{d,}$ probability of being idle: $P^{(s)}_{id} = 1 - P^{(s)}\mathrm{d}$
- Current state: s_0, target state: s_i, ETA: $\Delta t(s_0, s_i)$



Jiao, Y., Tang, X., Qin, Z.T., Li, S., Zhang, F., Zhu, H. and Ye, J., 2021. Real-world ride-hailing vehicle repositioning using deep reinforcement learning. *Transportation Research Part C: Emerging Technologies*, *130*, p.103289.

Rebalancing (Supply Repositioning)

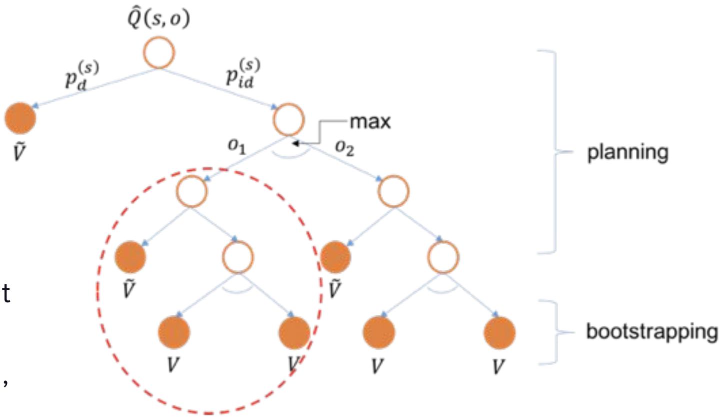
Interpretation

- Tree search = computing and comparing expected path values

Other approaches

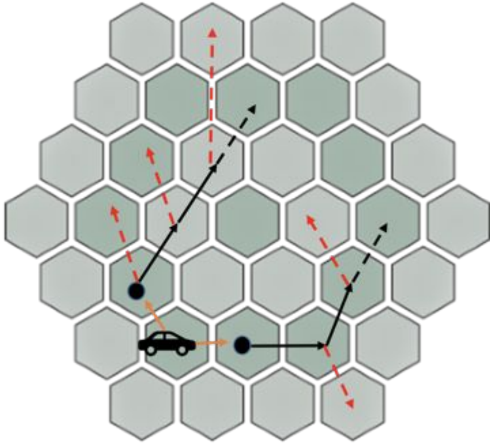
- Coordination through independent learning + global contextual info [Lin et al., 2018], [Oda & Joe-Wong, 2018], [Zhang et al., 2020]
- Bi-level MARL: [Shou & Di, 2020]

Tree search



[Jiao et al., 2021]

Expected path value



Black solid: reposition
Red dotted: dispatch
Black dotted: dispatch or reposition

Rebalancing (Supply Repositioning)

System-centric formulation

- Determines repositioning actions for all vehicles: joint actions

Modeling

- Global state information
- Aggregate level actions: e.g., number of idle vehicles to reposition from cell i to cell j at time t
  - Problem size independent from number of vehicles

Goal

- Influence future supply distribution to match better with demand so as to maximize total income and aggregate utilization

RL works

- [Feng et al., 2021] PPO
- [Mao et al., 2020] Batch AC: outputs a distribution of vehicles to allocate to each neighboring destinations

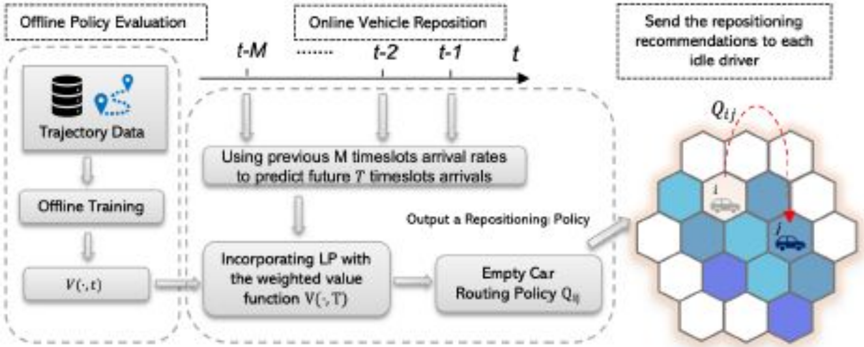
Rebalancing (Supply Repositioning) | MPC

Model-predictive control

- Uses short-term demand forecasting to plan future actions
- Usually ignores (and is impractical to consider) long-term effects due to computational complexity

LP formulation with lookahead

- LP based on fluid model for Short-Term Optimization
  - Models **repositioning rate** (only a fraction of drivers comply).
  - Relaxes flow constraints to account for system nonstationarity.
- Reinforcement Learning for Long-Term Rewards
  - Uses **value function approximation** to capture future demand-supply effects.
  - Trained using **historical driver trajectories** with TD learning.
- Real-Time Prediction & Optimization
  - **LSTM-CNN** based arrival rate prediction.
  - Online updates to optimize **fleet movement over multiple time steps (T-step lookahead).**


Offline Policy Evaluation
Online Vehicle Reposition
Send the repositioning recommendations to each idle driver
t-M
t-2
t-1
t
Trajectory Data
Offline Training
$V(\cdot,t)$
Using previous M timeslots arrival rates to predict future T timeslots arrivals
Output a Repositioning Policy
Incorporating LP with the weighted value function $V(\cdot,T)$
Empty Car Routing Policy Q_{ij}
Q_{ij}
i
j

Wei, H., Yang, Z., Liu, X., Qin, Z., Tang, X. and Ying, L., 2023. A reinforcement learning and prediction-based lookahead policy for vehicle repositioning in online ride-hailing systems. *IEEE Transactions on Intelligent Transportation Systems*, *25*(2), pp.1846-1856.

Search & Recommendation

Overview

- Generalized form of online matching
- Works specifically for decentralized transaction-making
  - In a centralized system (e.g., ridesharing), search collapses into online matching.
- Presents relevant supply options to demand
- Presents relevant requests (demand) to service providers (supply)

Domain-specific problems

- Retail: personalized product ranking (from different vendors)
- Vacation rental: property visibility optimization
- Food delivery: restaurant discovery

Difference from online matching

- Presents multiple service providers to the requester/buyer. The buyer makes the final selection.
- In online matching, the system presents the single option to the buyer.

Search & Recommendation

Objective

- To best meet the buyer's needs so that a transaction is most likely to occur

Trade-off considerations

- Prices associated with the recommendations in the search results
  - Too high: demand lost
  - Too low: platform loses on revenue
- Relevance of the search results to the buyer's demand
  - Jobs: matching roles from hiring teams
  - Rental: location and features of the properties
  - Retail: item features
  - Food delivery: food items that the store offers
- User experience
  - Customer desires quick and relevant match

There's a symmetric problem for recommendations to sellers/freelancers.

Search & Recommendation | Case Studies

Meituan's Takeout Recommendation System

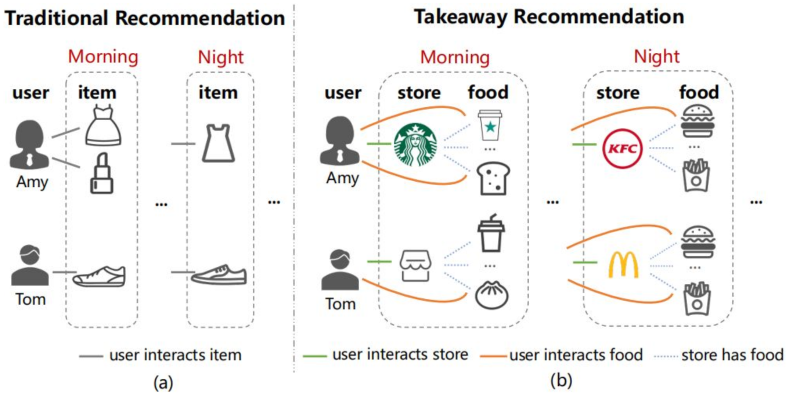
- *Zhang et al. Modeling Dual Period-Varying Preferences for Takeaway Recommendation* (KDD 2023)

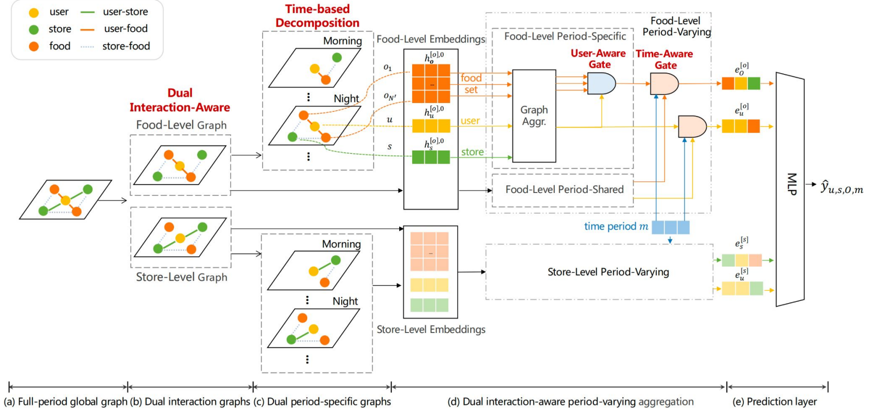
Key challenges

- Dual interaction-aware preferences
- Time-varying preferences

Technical highlights

- Dual interaction-aware module
- Time-based decomposition module
- User-/time-aware gating mechanism





Search & Recommendation | Case Studies

Embeddings for improving search relevance at Instacart

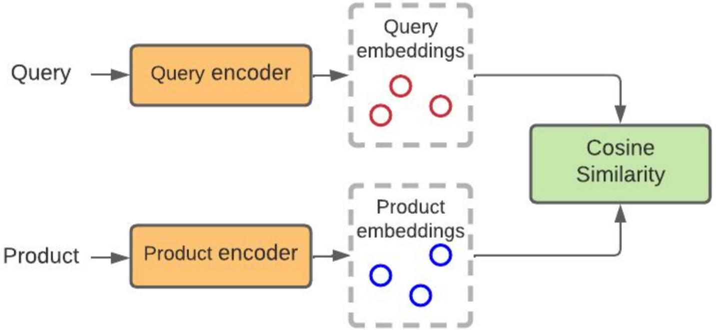
- **I**nstacart **T**ransformer-based **E**mbedding **M**odel for **S**earch (ITEMS)
- Deep learning model for unified representations of search queries and products, improving search relevance, especially for ambiguous or long-tail queries

Model architecture

- Two-Tower Transformer Structure
- Semantic Similarity Assessment

Training & implementation

- Fine-tuned using Instacart's search impression logs, learning from both positive and negative query-product pairs.
- Complements keyword-based and category-based retrieval methods, particularly effective for complex or less common queries.



https://www.instacart.com/company/how-its-made/how-instacart-uses-embeddings-to-improve-search-relevance/

Search & Recommendation | Fairness

Two-sided fairness

- *Patro et al., 2020. FairRec: Two-Sided Fairness for Personalized Recommendations in Two-Sided Platforms* (WWW 2020)
- Focus on **maximizing customer satisfaction** often leads to **unfair exposure distribution for producers** (e.g., sellers, restaurants, content creators).
- **Over-exposure to popular producers** & **under-exposure to less popular ones** → negatively impacts marketplace sustainability.
- A **producer-centric design** may harm **customer experience**, creating a trade-off.

Fair allocation

- **Maximin Share (MMS)**: Guarantees a minimum level of exposure for producers.
- **Envy-Free up to One Item (EF1)**: Ensures customers don't feel significantly disadvantaged.

Algorithm

- **Step 1:** Assigns products ensuring fair exposure among producers.
- **Step 2:** Allocates recommendations in a way that minimizes customer envy.

Also see: Sühr, T., Biega, A.J., Zehlike, M., Gummadi, K.P. and Chakraborty, A., 2019, July. Two-sided fairness for repeated matchings in two-sided markets: A case study of a ride-hailing platform. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 3082-3092).

Search & Recommendation | Case Studies

Fairness in LinkedIn's Recommendation Algorithms

- Yu & Saint-Jacques. Choosing an Algorithmic Fairness Metric for an Online Marketplace: Detecting and quantifying algorithmic bias on LinkedIn

Why fairness in marketplace recommendations?

- Algorithms influence who gets recommended job opportunities, connections, or services.
- Biased recommendations can reduce opportunities for underrepresented groups.
- Need for a precise fairness metric that separates algorithmic bias from human bias.

Technical highlights

- Fairness test based on **marginal candidate outcomes**
- Fairness metric for marketplace recommendations: **equal opportunity for equally qualified candidates**
- Separating algorithmic bias from human bias

Dynamic Pricing

Centralized

- Platform sets the price for each request
- Platform also sets the corresponding pay for the supply
- Typical for ridesharing platforms

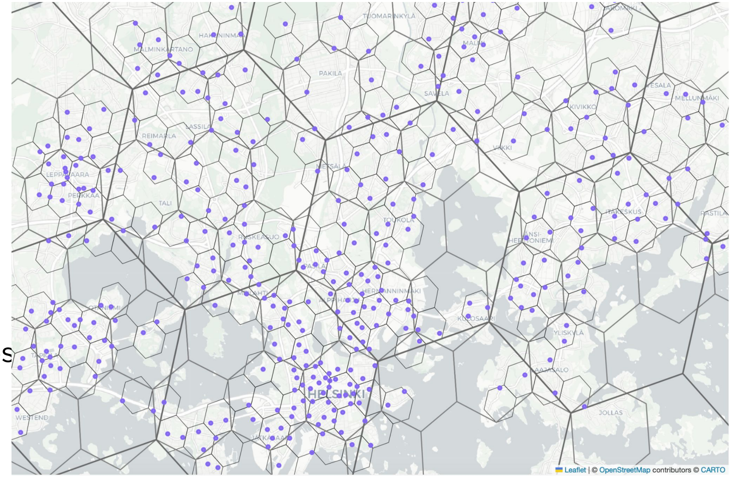
Decentralized

- Seller/service provider sets the price for the item/service offered. Platform provides pricing guidance.
- Service provider bids for a specific demand/request, typically in a demand broadcasting matching mechanism.
- Buyer (service requester) sets the pay for a request. Service provider decides whether to take on a request.
- Typical for vacation rental, retail, and consulting projects

Dynamic Pricing | Centralized (Ridesharing)

Spatiotemporal pricing

- Chen, et al., 2021. Spatial-temporal pricing for ride-sourcing platform with reinforcement learning. *Transportation Research Part C: Emerging Technologies*,.
- Pricing decision for each hexagonal cell: (per-km rate for excess mileage beyond a base trip distance, per-km rate for driver wage)
- Objective: maximizing total profits



Agent

- Global decision-maker
- State info: the numbers of open requests, vacant vehicles, occupied vehicles in each grid cell at time t and historical demand at time t -1
- Different rider and driver elasticity functions as part of the env

Learning

- PPO

Dynamic Pricing | Centralized (Ridesharing)

Joint pricing with online matching

- Dynamic price adjustments through % movement on base price
- Goal: maximize long-term cumulative returns

Macro-lever interaction

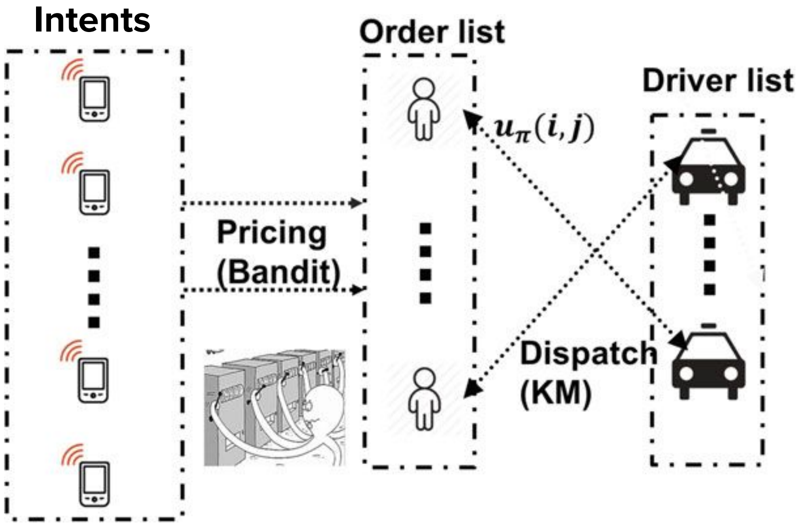
- Price changes affect demand distribution, which in turn has impact on dispatch outcomes, even with the same driver group and matching policy.

Dynamic pricing decisions

- Contextual bandits on a set of discrete price % adjustments

Long-term value

- Conversion probability * trip value computed from TD errors based on supply values

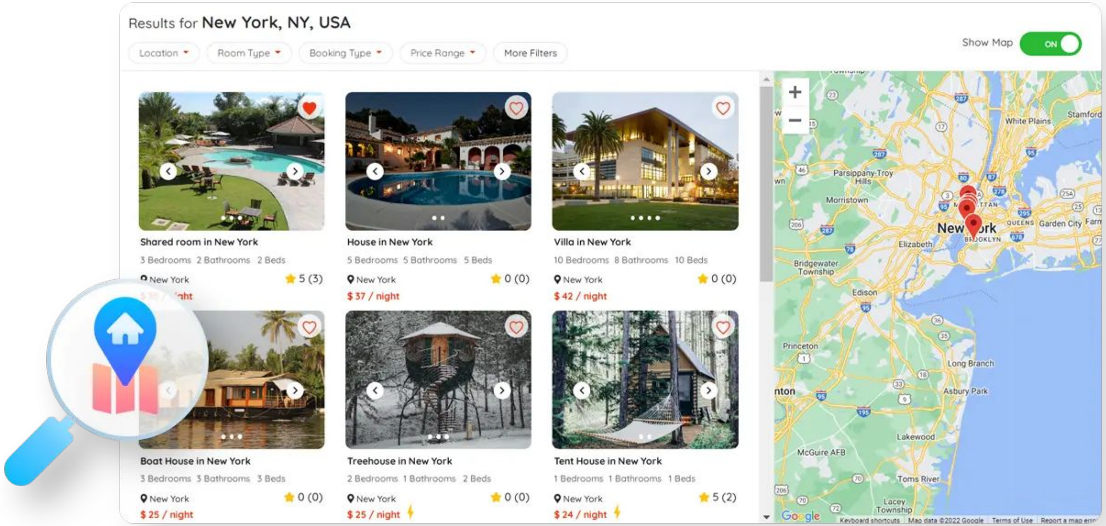


Chen, H., Jiao, Y., Qin, Z., Tang, X., Li, H., An, B., Zhu, H. and Ye, J., 2019, November. InBEDE: Integrating contextual bandit with TD learning for joint pricing and dispatch of ride-hailing platforms. In *2019 IEEE International Conference on Data Mining (ICDM)* (pp. 61-70). IEEE.

Dynamic Pricing | Decentralized (1)

Platforms provides pricing guidance

- Airbnb, Amazon, UpWork
- Service providers (sellers) have “preset” services and set their own price or hourly rate
- Platform guidance helps both sellers and platform.



Results for New York, NY, USA

Location ▾ | Room Type ▾ | Booking Type ▾ | Price Range ▾ | More Filters

Show Map ON

Shared room in New York
3 Bedrooms 2 Bathrooms 2 Beds
New York ★ 5 (3)
$ [illegible] / night

House in New York
5 Bedrooms 5 Bathrooms 5 Beds
New York ★ 0 (0)
$ 37 / night

Villa in New York
10 Bedrooms 8 Bathrooms 10 Beds
New York ★ 0 (0)
$ 42 / night

Boat House in New York
3 Bedrooms 3 Bathrooms 3 Beds
New York ★ 0 (0)
$ 25 / night

Treehouse in New York
2 Bedrooms 1 Bathrooms 2 Beds
New York ★ 0 (0)
$ 25 / night

Tent House in New York
1 Bedrooms 1 Bathrooms 1 Beds
New York ★ 5 (2)
$ 24 / night

Dynamic Pricing | Decentralized (1)

Key difference from ridesharing

- Real-timeness

Time sensitivity

- Pricing adjustments are more gradual, considering longer-term factors such as upcoming events or seasonal trends.
- Ridesharing: Pricing is highly sensitive to real-time conditions

Supply elasticity

- The supply of available properties is relatively fixed in the short term.
- Ridesharing: Driver availability can change quickly in response to surge pricing.

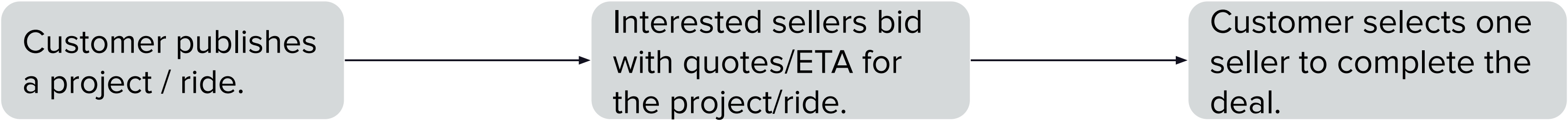
Customer decision-making

- Guests typically plan their stays in advance
- Ridesharing: Consumers often make spontaneous decisions, with price playing a critical role in immediate choice.

Dynamic Pricing | Decentralized (2)

Service providers / sellers bid for a demand / request

- Rideshare in early days
- Contractual projects. Angi, UpWork (project-based fixed price)
- Similar to the previous setting but with on-demand request-specific pricing



Customer publishes a project / ride.

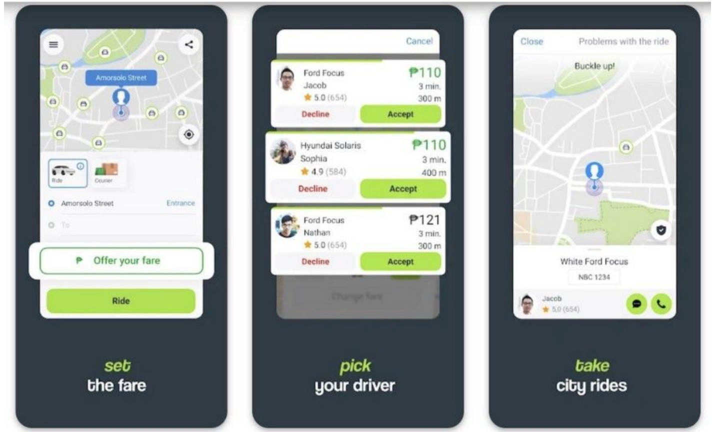
Interested sellers bid with quotes/ETA for the project/ride.

Customer selects one seller to complete the deal.

Dynamic Pricing | Decentralized (3)

Buyer sets the pay for a request

- Some rideshare platforms adopt this mechanism - perceivably more fair.
  - **Riders set their own price** for a ride.
  - **Drivers can accept, decline, or counteroffer** with their own bid.
  - Riders then choose the driver based on price, rating, and estimated arrival time.
- Freelancing (UpWork): clients can set their budgets, and freelancers submit their proposals to specific projects.



Growth & Incentives

Significance

- The value of the platform for either side (supply/seller, demand/buyers) depends on the availability of the other side.
- The size of both user groups have to be in “balance” for the marketplace to be efficient.

Target populations

- Buyers (demand-side): discounts on price
- Sellers (supply-side): bonus in pay
- Other non-monetary incentives: priority in matching or recommendations

When to distribute

- Real-time: triggered by real-time events (e.g., demand intents, cancellation)
- Batch: decisions are more about the precise target group, often for life-cycle management on the platform

Growth & Incentives | Supply-side

Real-time incentives

- More often seen on ridesharing or food delivery platforms
- Vehicle repositioning

Batch incentives

- Ridesharing: target-based driver incentives (ride streaks)
- Food delivery: "Complete 20 deliveries in a week and earn a $100 bonus."
- Earning guarantees for new participants
- Seasonal listing promotions in vacation rentals

Optimization

- Decisions: incentive structure and amount, target group, triggering time
- Learning algorithm for optimizing policy
- Causal inference for estimating uplift effects

Growth & Incentives | Supply-side

Target-based incentives for drivers (ride streaks)

- Complete X rides today to get $Y in bonus.
- Can be tiered: complete X+5 rides today to get an additional $Z in bonus
- Typically targeted in a batch, planned manner: e.g., distribute today, take effect tomorrow

Goal

- To incentivize drivers to stay longer with the platform -> more driver hours -> more supply

Problems / considerations

- Cost: bonus amount, probability of getting the bonus (hitting the target)
- Returns: uplift in driver hours on the same day, and longer-term effects
  - Long-term effects can be negative: driver keeps getting targets that are too hard to hit
- RL on incentive policies: [Shang, et al., 2019]
- Causal inference on uplift effects: [Huang, et al., 2022, Shang, et al., 2021]
- Effects on supply behavior: [Liu, et al., 2023a, Liu, et al., 2023b]

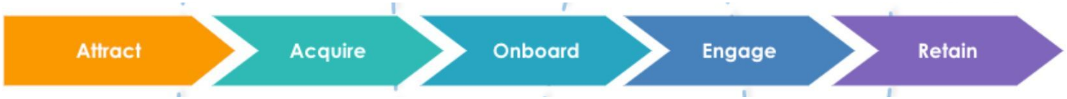
Growth & Incentives | Demand-side

Real-time incentives

- Intent-based discounts
- Cancellation-triggered discounts
- Bundling upon check-out

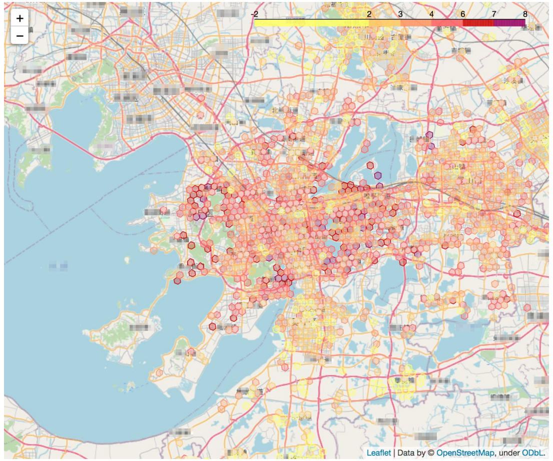
Batch incentives

- Customer life-cycle management
  - First-purchase discounts
  - Tiered loyalty program
- Streak-based rewards



Growth & Incentives | Demand-side

[Wiu et al., 2022]



Intent-based discounts (ridesharing)

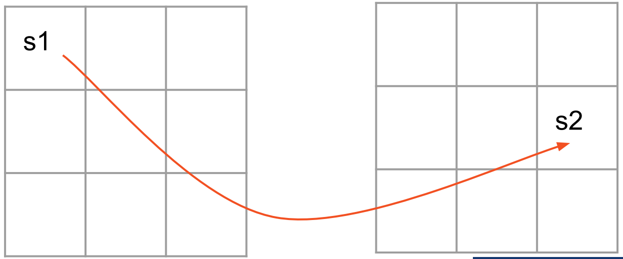
- Ride intent: the action of viewing a quote for a particular ODT (origin-destination-time) combination. Not an actual order yet but a strong signal of potential demand.
- Typically ODT -specific and real-time

Purpose

- To shape demand (spatiotemporal) distribution to align better with future supply distribution to maximize long-term (daily) returns

Problems / approaches

- Estimating trip value: marginal demand value via supply values (TD errors)
- Estimating uplift in probability of order conversion: causal inference



References

Policy Optimization

- Qin, Z., Tang, X., Li, Q., Zhu, H. and Ye, J., 2025. *Reinforcement Learning in the Ridesharing Marketplace*. Synthesis Lectures on Learning, Networks, and Algorithms, Springer.
- Xu, Z., Li, Z., Guan, Q., Zhang, D., Li, Q., Nan, J., Liu, C., Bian, W. and Ye, J., 2018, July. Large-scale order dispatch in on-demand ride-hailing platforms: A learning and planning approach. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 905-913).
- Wang, Z., Qin, Z., Tang, X., Ye, J. and Zhu, H., 2018, November. Deep reinforcement learning with knowledge transfer for online rides order dispatching. In *2018 IEEE International Conference on Data Mining (ICDM)* (pp. 617-626). IEEE.
- Qin, Z., Tang, X., Jiao, Y., Zhang, F., Xu, Z., Zhu, H. and Ye, J., 2020. Ride-hailing order dispatching at DiDi via reinforcement learning. *INFORMS Journal on Applied Analytics*, *50*(5), pp.272-286.
- Tang, X., Qin, Z., Zhang, F., Wang, Z., Xu, Z., Ma, Y., Zhu, H. and Ye, J., 2019, July. A deep value-network based approach for multi-driver order dispatching. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 1780-1790).
- Tang, X., Zhang, F., Qin, Z., Wang, Y., Shi, D., Song, B., Tong, Y., Zhu, H. and Ye, J., 2021, August. Value function is all you need: A unified learning framework for ride hailing platforms. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining* (pp. 3605-3615).

References

Policy optimization

- Han, B., Lee, H. and Martin, S., 2022, August. Real-time rideshare driver supply values using online reinforcement learning. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (pp. 2968-2976).
- Sadeghi Eshkevari, S., Tang, X., Qin, Z., Mei, J., Zhang, C., Meng, Q. and Xu, J., 2022, August. Reinforcement learning in the wild: Scalable RL dispatching algorithm deployed in ridehailing marketplace. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (pp. 3838-3848).
- Azagirre, X., Balwally, A., Candeli, G., Chamandy, N., Han, B., King, A., Lee, H., Loncaric, M., Martin, S., Narasiman, V. and Qin, Z., 2024. A better match for drivers and riders: Reinforcement learning at lyft. *INFORMS Journal on Applied Analytics*, *54*(1), pp.71-83.
- Feng, J., Gluzman, M. and Dai, J.G., 2021, May. Scalable deep reinforcement learning for ride-hailing. In *2021 American Control Conference (ACC)* (pp. 3743-3748). IEEE.
- Li, M., Qin, Z., Jiao, Y., Yang, Y., Wang, J., Wang, C., Wu, G. and Ye, J., 2019, May. Efficient ridesharing order dispatching with mean field multi-agent reinforcement learning. In *The world wide web conference* (pp. 983-994).

References

Policy Optimization

- Jin, J., Zhou, M., Zhang, W., Li, M., Guo, Z., Qin, Z., Jiao, Y., Tang, X., Wang, C., Wang, J. and Wu, G., 2019, November. Coride: joint order dispatching and fleet management for multi-scale ride-hailing platforms. In *Proceedings of the 28th ACM international conference on information and knowledge management* (pp. 1983-1992).
- Zhou, M., Jin, J., Zhang, W., Qin, Z., Jiao, Y., Wang, C., Wu, G., Yu, Y. and Ye, J., 2019, November. Multi-agent reinforcement learning for order-dispatching via order-vehicle distribution matching. In *Proceedings of the 28th ACM international conference on information and knowledge management* (pp. 2645-2653).
- Simao, H.P., Day, J., George, A.P., Gifford, T., Nienow, J. and Powell, W.B., 2009. An approximate dynamic programming algorithm for large-scale fleet management: A case application. *Transportation Science*, *43*(2), pp.178-197.
- Al-Kanj, L., Nascimento, J. and Powell, W.B., 2020. Approximate dynamic programming for planning a ride-hailing system using autonomous fleets of electric vehicles. *European Journal of Operational Research*, *284*(3), pp.1088-1106.
- Zhang, L., Hu, T., Min, Y., Wu, G., Zhang, J., Feng, P., Gong, P. and Ye, J., 2017, August. A taxi order dispatch model based on combinatorial optimization. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 2151-2159).

References

Policy Optimization

- Jiao, Y., Tang, X., Qin, Z.T., Li, S., Zhang, F., Zhu, H. and Ye, J., 2021. Real-world ride-hailing vehicle repositioning using deep reinforcement learning. *Transportation Research Part C: Emerging Technologies*, *130*, p.103289.
- Mao, C. and Shen, Z., 2018. A reinforcement learning framework for the adaptive routing problem in stochastic time-dependent network. *Transportation Research Part C: Emerging Technologies*, *93*, pp.179-197.
- Wei, H., Yang, Z., Liu, X., Qin, Z., Tang, X. and Ying, L., 2023. A reinforcement learning and prediction-based lookahead policy for vehicle repositioning in online ride-hailing systems. *IEEE Transactions on Intelligent Transportation Systems*, *25*(2), pp.1846-1856.
- Zhang, Y., Wu, Y., Le, R., Zhu, Y., Zhuang, F., Han, R., Li, X., Lin, W., An, Z. and Xu, Y., 2023, August. Modeling dual period-varying preferences for takeaway recommendation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (pp. 5628-5638).
- Yu, Y. and Saint-Jacques, G., 2022. Choosing an algorithmic fairness metric for an online marketplace: Detecting and quantifying algorithmic bias on LinkedIn. *arXiv preprint arXiv:2202.07300*.
- Patro, G.K., Biswas, A., Ganguly, N., Gummadi, K.P. and Chakraborty, A., 2020, April. Fairrec: Two-sided fairness for personalized recommendations in two-sided platforms. In *Proceedings of the web conference 2020* (pp. 1194-1204).

References

Policy optimization

- Sühr, T., Biega, A.J., Zehlike, M., Gummadi, K.P. and Chakraborty, A., 2019, July. Two-sided fairness for repeated matchings in two-sided markets: A case study of a ride-hailing platform. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 3082-3092).
- Chen, C., Yao, F., Mo, D., Zhu, J. and Chen, X.M., 2021. Spatial-temporal pricing for ride-sourcing platform with reinforcement learning. *Transportation Research Part C: Emerging Technologies*, *130*, p.103272.
- Chen, H., Jiao, Y., Qin, Z., Tang, X., Li, H., An, B., Zhu, H. and Ye, J., 2019, November. InBEDE: Integrating contextual bandit with TD learning for joint pricing and dispatch of ride-hailing platforms. In *2019 IEEE International Conference on Data Mining (ICDM)* (pp. 61-70). IEEE.
- Shang, W., Li, Q., Qin, Z., Yu, Y., Meng, Y. and Ye, J., 2021. Partially observable environment estimation with uplift inference for reinforcement learning based recommendation. *Machine Learning*, *110*(9), pp.2603-2640.
- Huang, T., Li, Q. and Qin, Z., 2022, December. Multiple tiered treatments optimization with causal inference on response distribution. In *2022 IEEE International Conference on Big Data (Big Data)* (pp. 962-971). IEEE.
- Liu, T., Xu, Z., Vignon, D., Yin, Y., Li, Q. and Qin, Z., 2023. Effects of threshold-based incentives on drivers' labor supply behavior. *Transportation Research Part C: Emerging Technologies*, *152*, p.104140.
- Liu, T., Xu, Z., Vignon, D., Yin, Y., Qin, Z. and Li, Q., 2023. Threshold-based incentives for ride-sourcing drivers: Implications on supply management and welfare effects. *Transportation Research Part C: Emerging Technologies*, *156*, p.104323.

References

Policy optimization

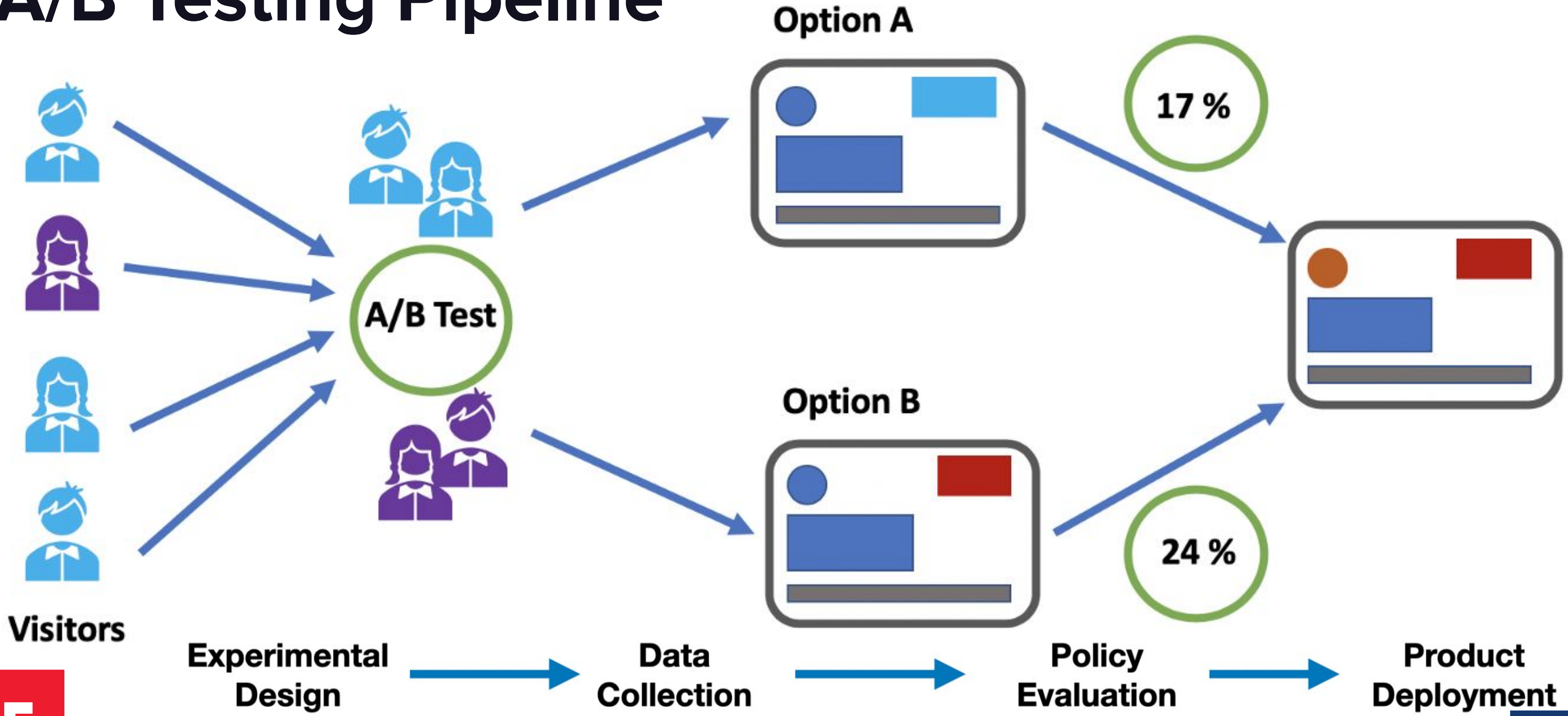
- Wu, Y., Li, Q. and Qin, Z., 2022. Spatio-temporal incentives optimization for ride-hailing services with offline deep reinforcement learning. *arXiv preprint arXiv:2211.03240*.

A/B Testing: Policy Evaluation & Experimental Design

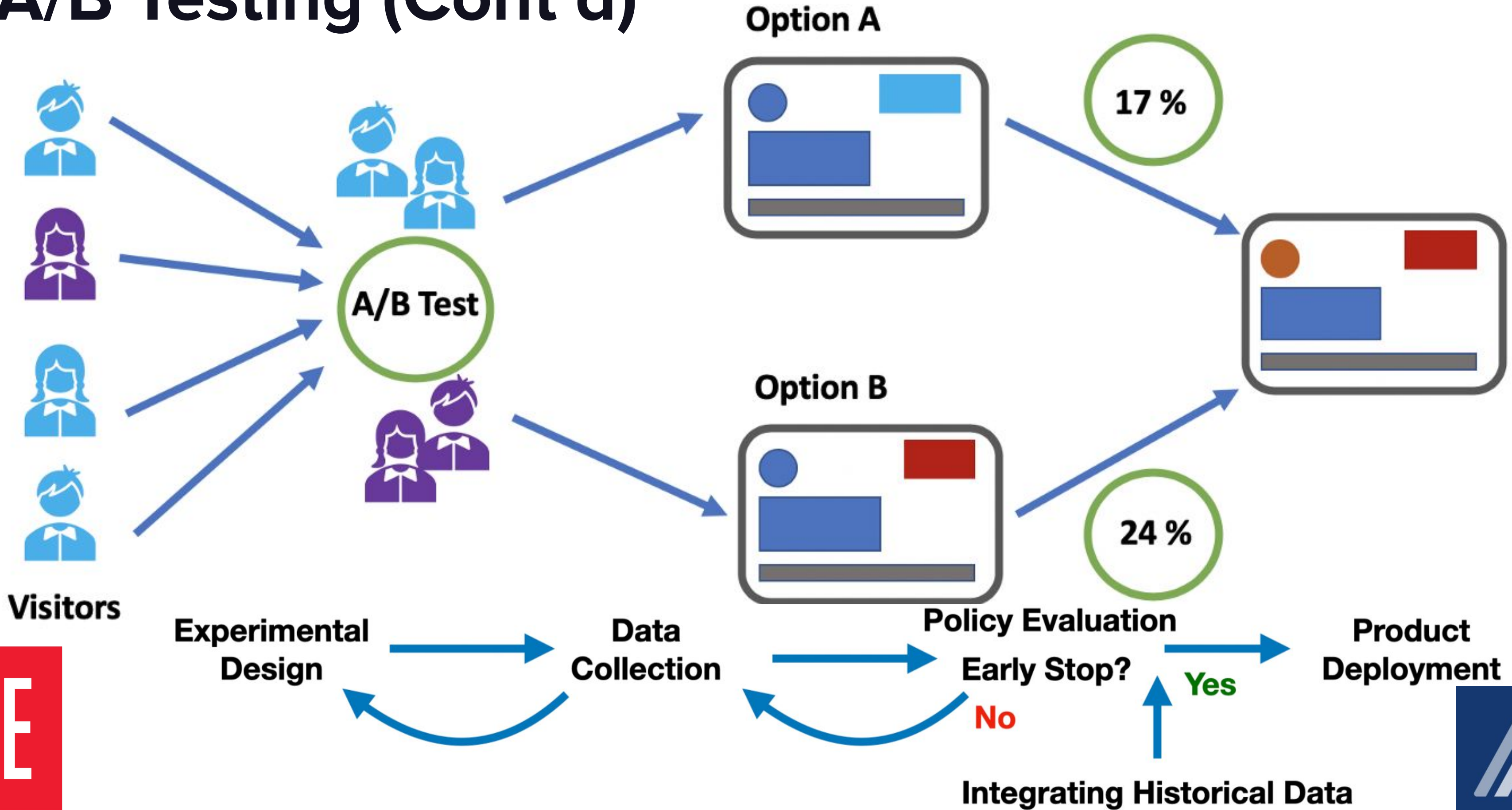
Chengchun Shi

London School of Economics and Political Science

A/B Testing Pipeline



A/B Testing (Cont’d)



Option A

17 %

A/B Test

Option B

24 %

Visitors

Experimental Design

Data Collection

Policy Evaluation

Early Stop?

Yes

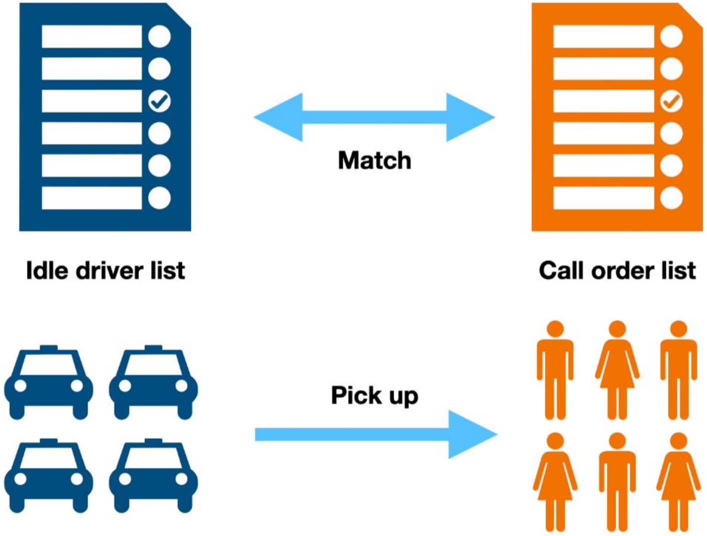
No

Product Deployment

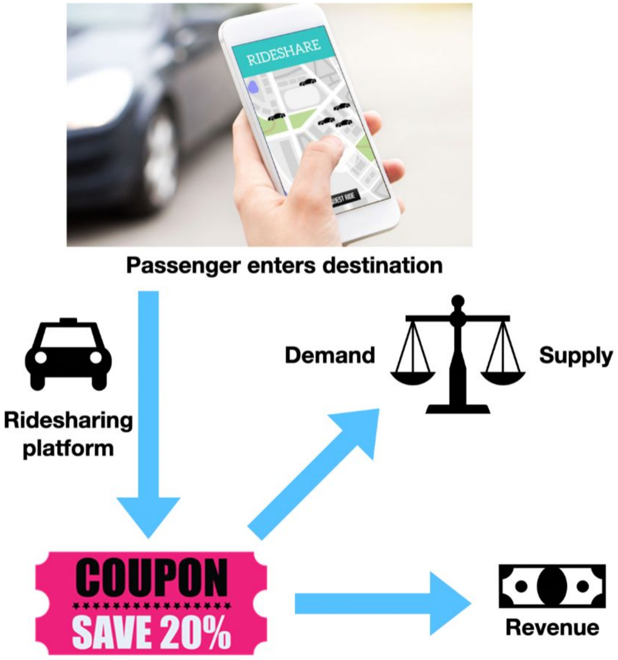
Integrating Historical Data

Examples of Target Policies in Ridesharing

- Order dispatching



- Subsidizing



Example: Order Dispatching

- Online experiments typically last for **two weeks**
- **30 minutes/1 hour** as one time unit, randomized over time
- Data forms a **time series**
- **Observations**:
  - **Outcome**: drivers' income or no. of completed orders
  - **Demand**: no. of call orders
  - **Supply**: no. of idle drivers
- **Treatment** (binary):
  - **New** order dispatching policy **B**
  - **Old** order dispatching policy **A**
- **Target**: **Average treatment effect** (ATE) = difference in average **outcome** between the **new** and **old** policy

Example: Subsidizing

- Randomized over population (e.g., passengers)
- **Panel data**: containing data from multiple individuals, each forms a time series
- **Observations**:
  - Individual-level **outcome**: passenger satisfaction
  - Individual-level **covariate**: passenger's demographics and historical service usage data
  - City-level **demand**: no. of call orders
  - City-level **supply**: no. of idle drivers
- **Treatment** (binary):
  - **New** order subsidizing policy **B**
  - **Old** order subsidizing policy **A**
- **Target**: **Average treatment effect** (ATE) = difference in average **outcome** between the **new** and **old** policy

Overview

Challenges in A/B testing

- Interference effects over time/space
- Partial observability
- Early termination
- Small sample size
- Weak signal
- Solutions to these challenges

Design of online experiments

- Designs and trade-offs
- A selective review of optimal designs
- Case study in ridesharing

Policy Evaluation

- Direct method
- Importance sampling
- Double robust method
- Model-based method
- Uncertainty quantification

Overview

Challenges in A/B testing

- Interference effects over time/space
- Partial observability
- Early termination
- Small sample size
- Weak signal
- Solutions to these challenges

Design of online experiments

- Designs and trade-offs
- A selective review of optimal designs
- Case study in ridesharing

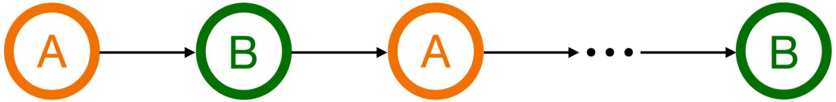
Policy Evaluation

- Direct method
- Importance sampling
- Double robust method
- Model-based method
- Uncertainty quantification

Challenge I: Interference Effects

Time series experiments: Carryover (delayed) effects over time

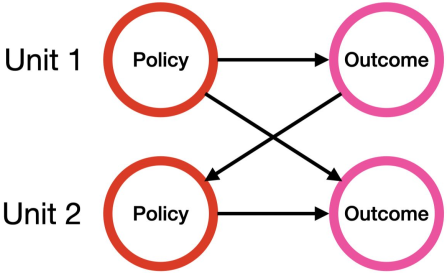
- **Past** treatments influence **future** observations/outcomes (Li, et al., 2024a, Figure 2)
- Under the **alternating-time** or **switchback** design



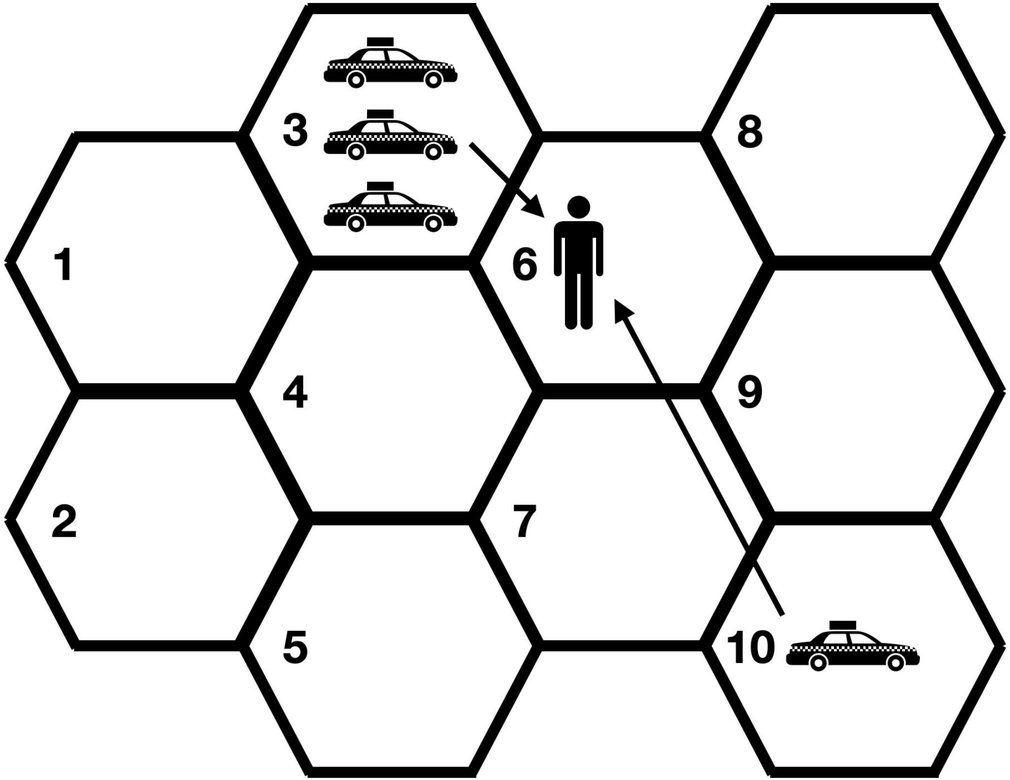
many conventional A/B testing/causal inference methods would **fail** (Shi et al., 2023a)

Multi-unit experiments: Spillover effects across units

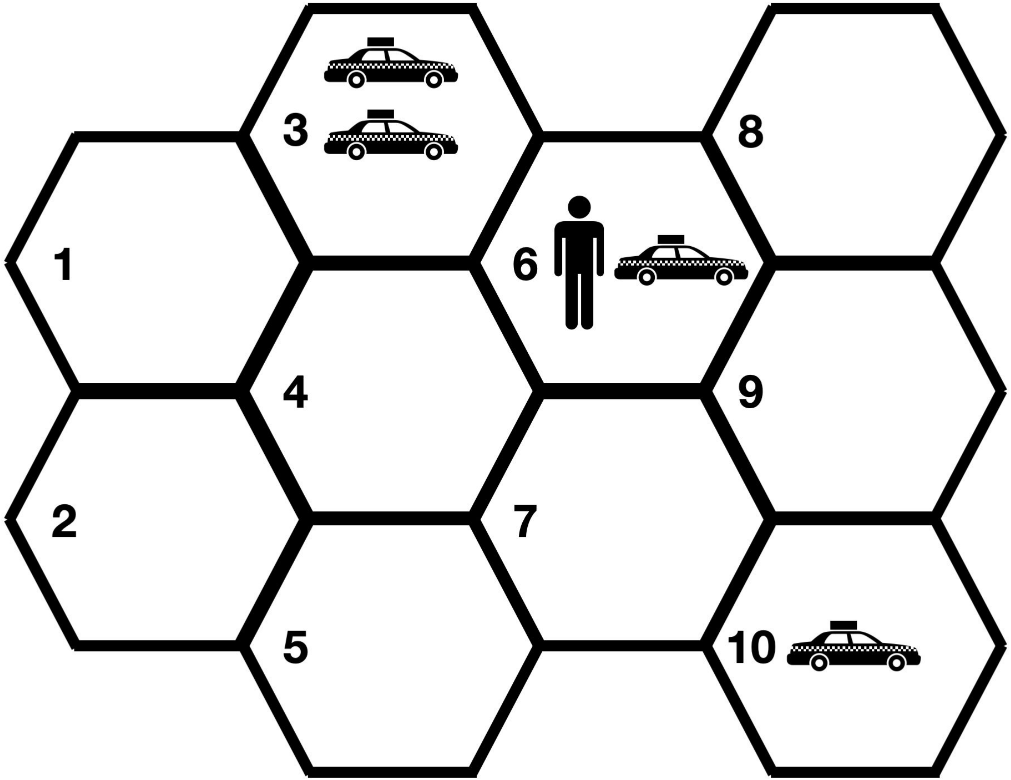
- Each unit's outcome/observation depends on **both** its own treatment and treatments from other units



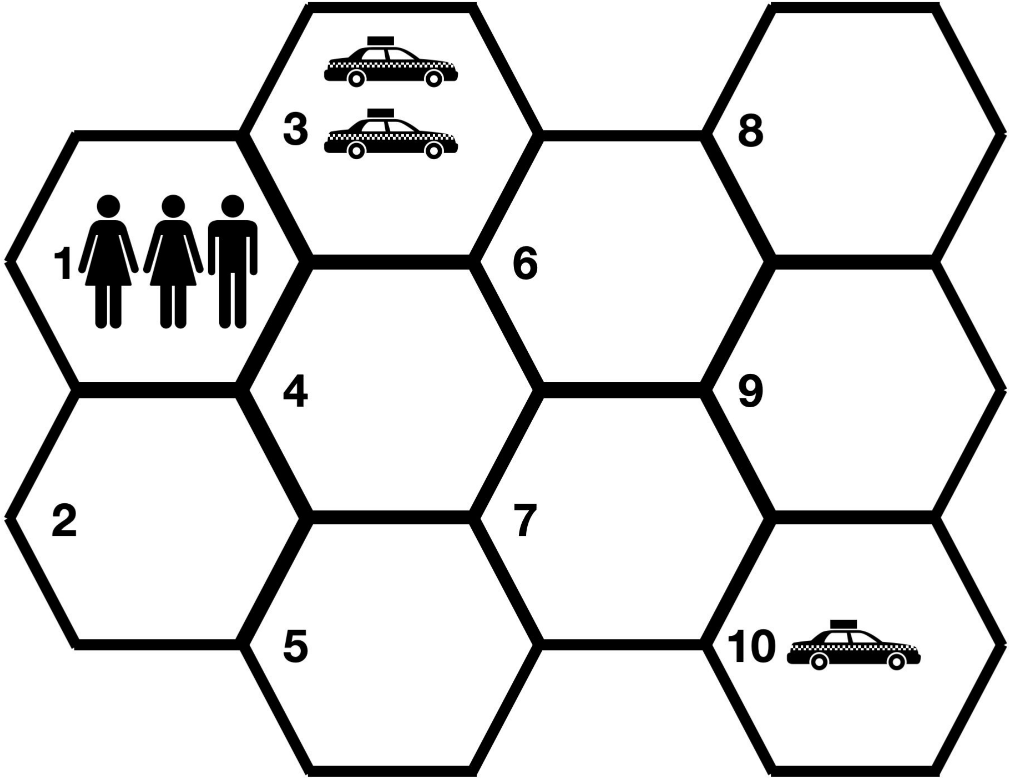
Challenge I: Carryover Effects over Time



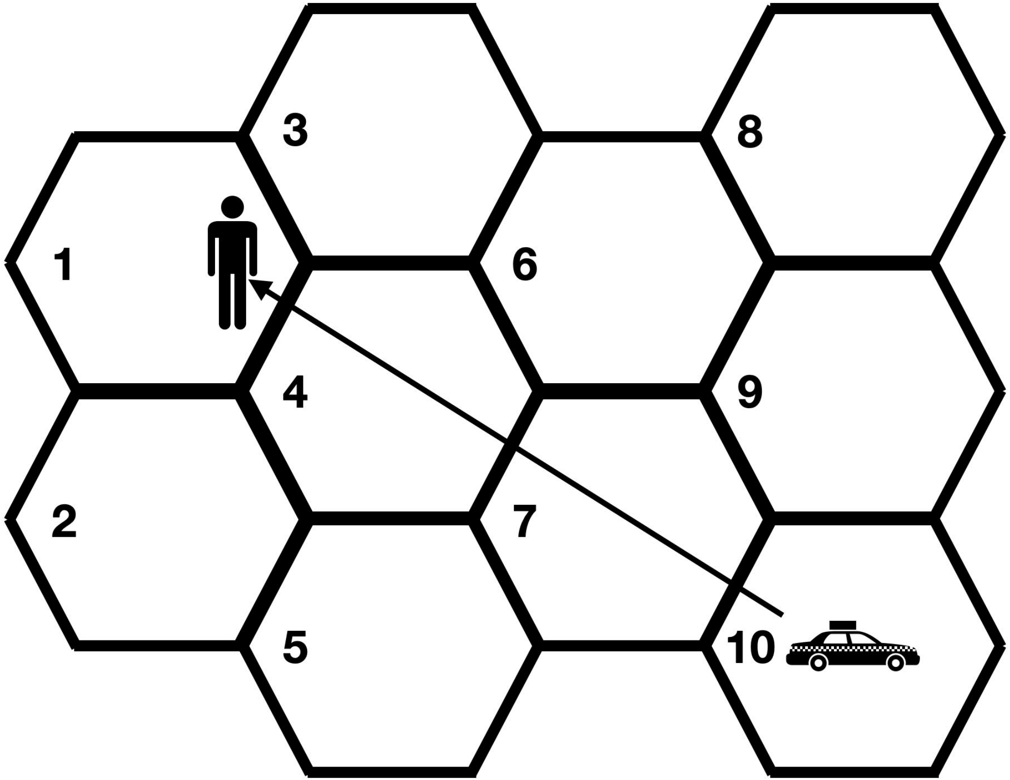
Adopting the Closest Driver Policy



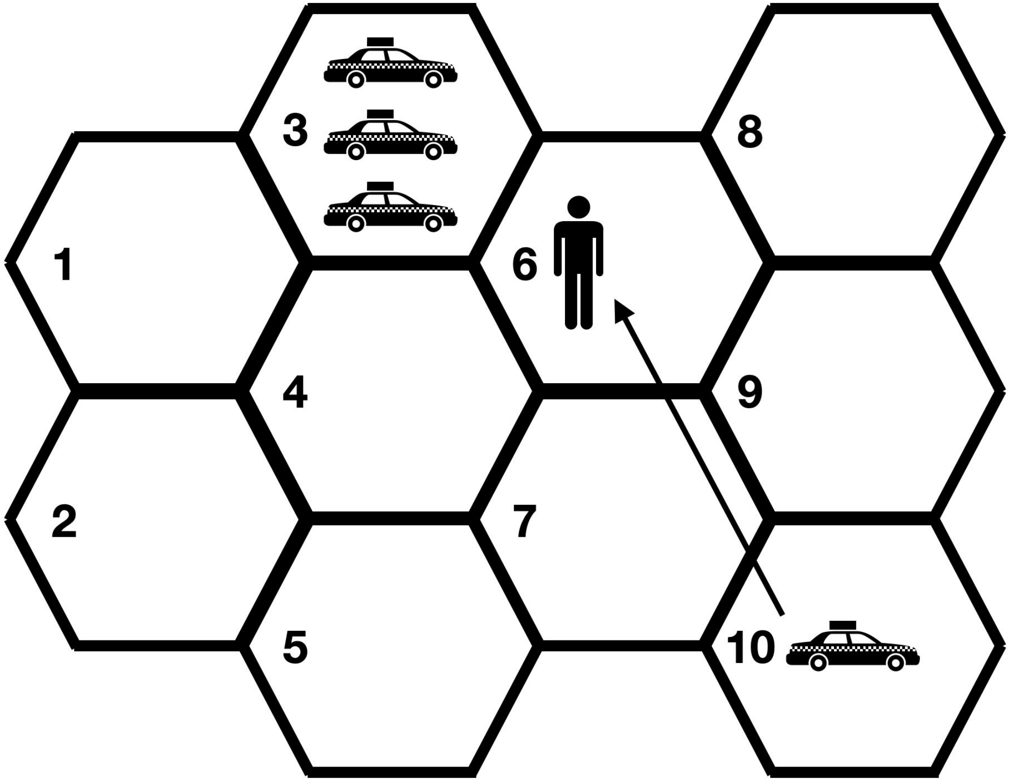
Some Time Later ...



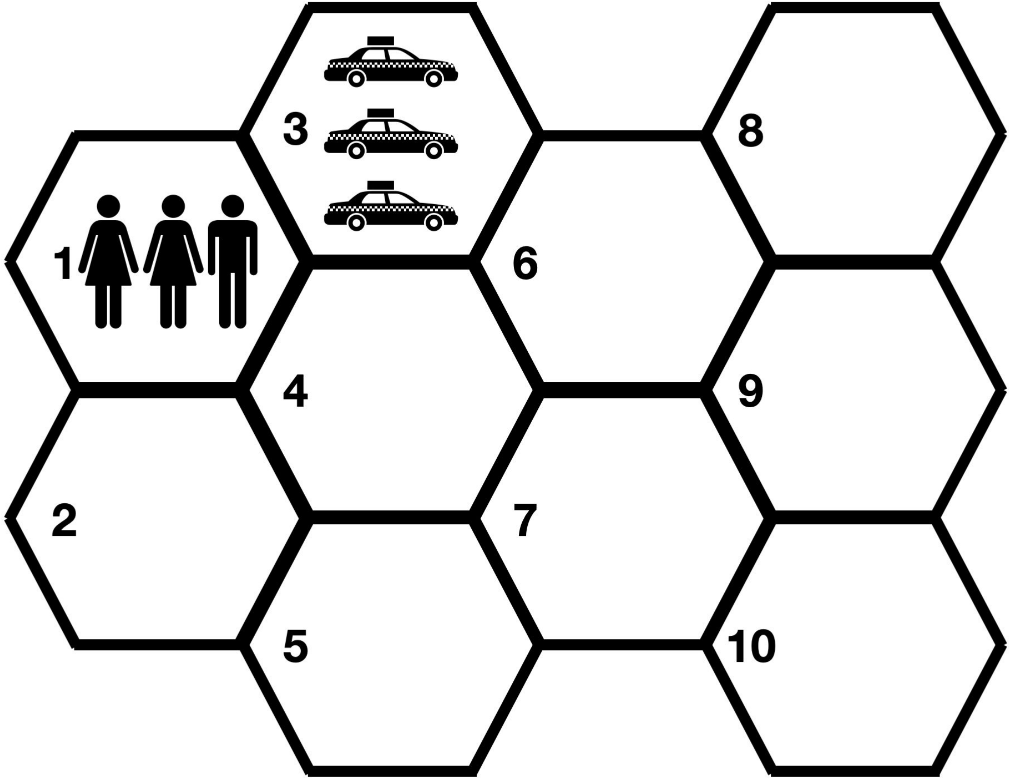
Miss One Order



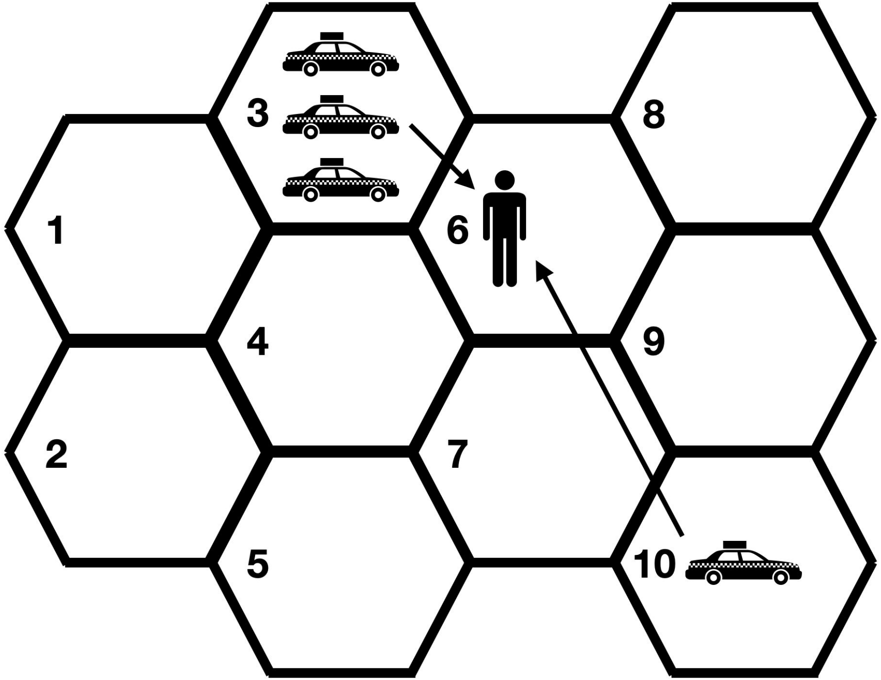
Consider a Different Policy



Able to Match All Orders

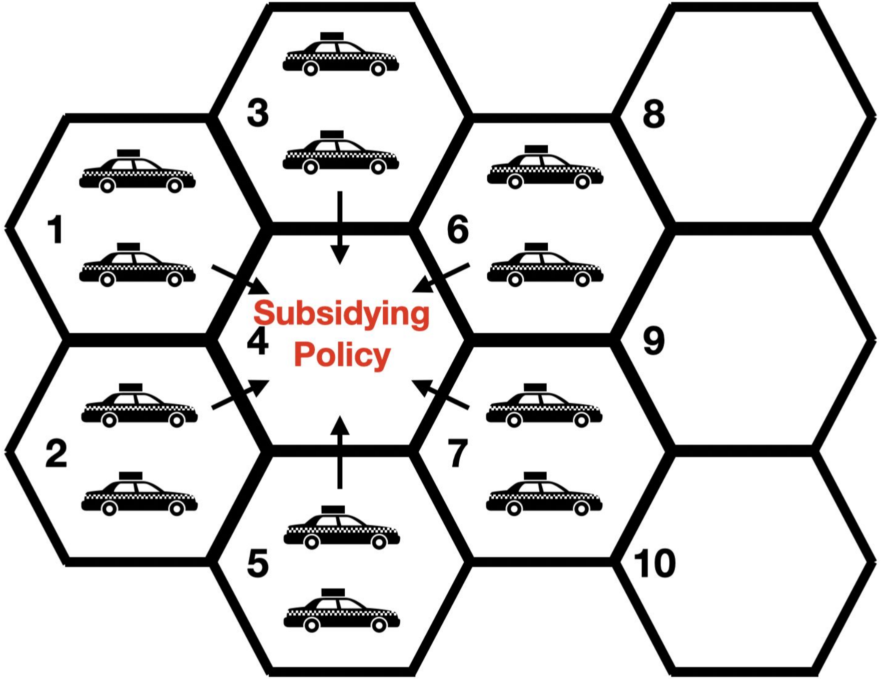


Challenge I: Carryover Effects over Time (Cont'd)



past treatments → distribution of drivers → future outcomes

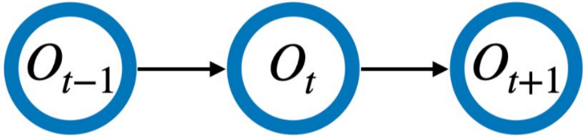
Challenge I: Spillover Effects over Space



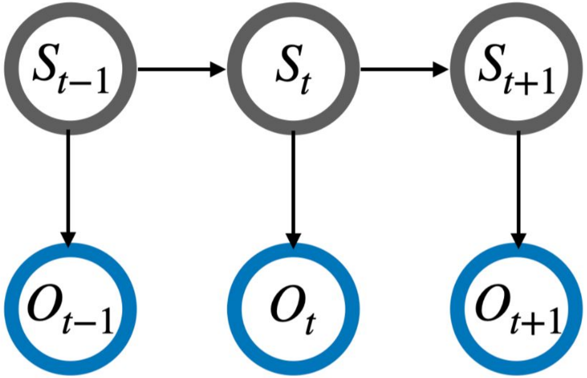
policy in one location → drivers from neighbouring location
→ outcomes in neighbouring location

Challenge II: Partial Observability

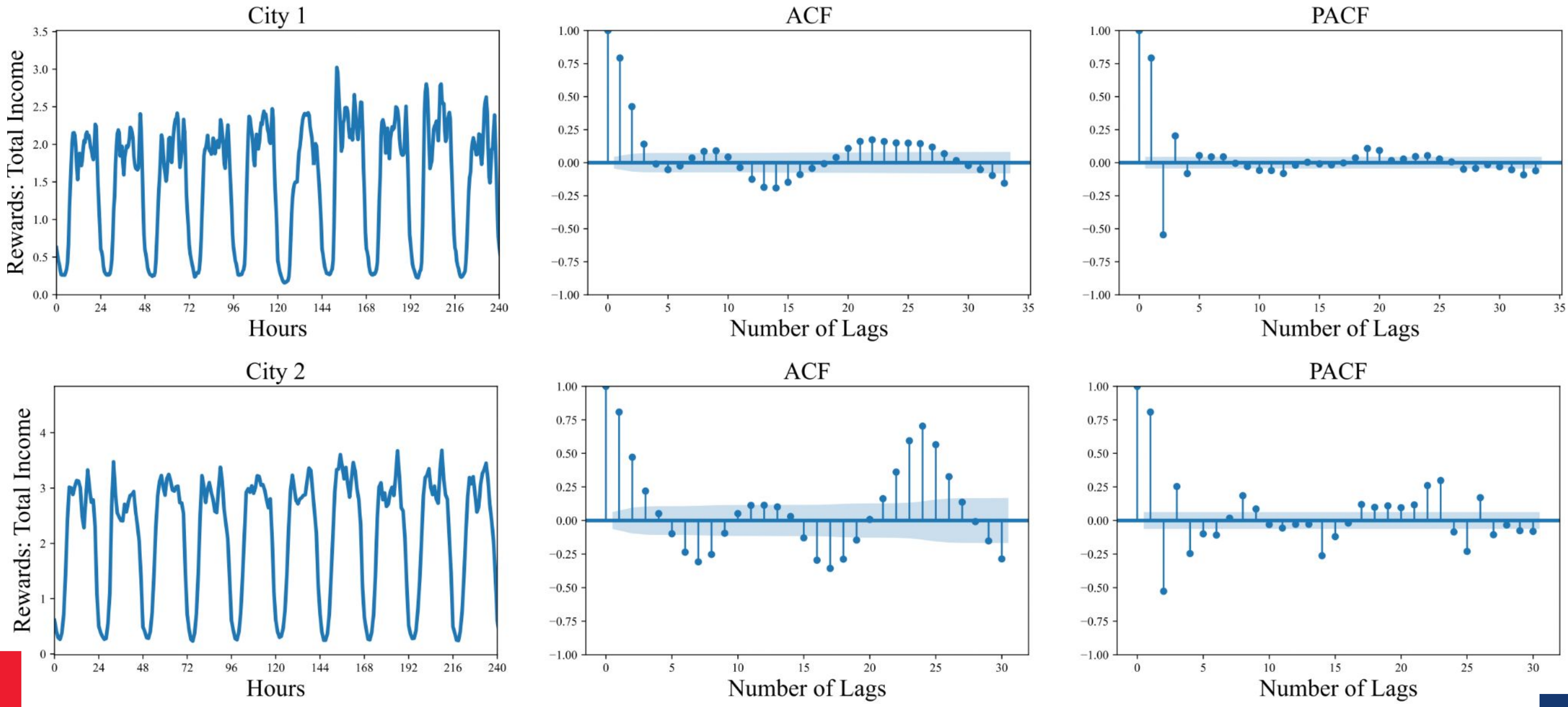
- Fully Observable Markovian Environments



- Partially Observable non-Markovian Environments



Challenge II: Partial Observability



Other Challenges

Challenge III: The need for early termination

- Each experiment takes a considerable time
- Early termination to save time and budget

Challenge IV: Small sample size

- Online experiments last at most 2 weeks (Xu et al., 2018)
- Increasing the variability of the treatment effect estimator

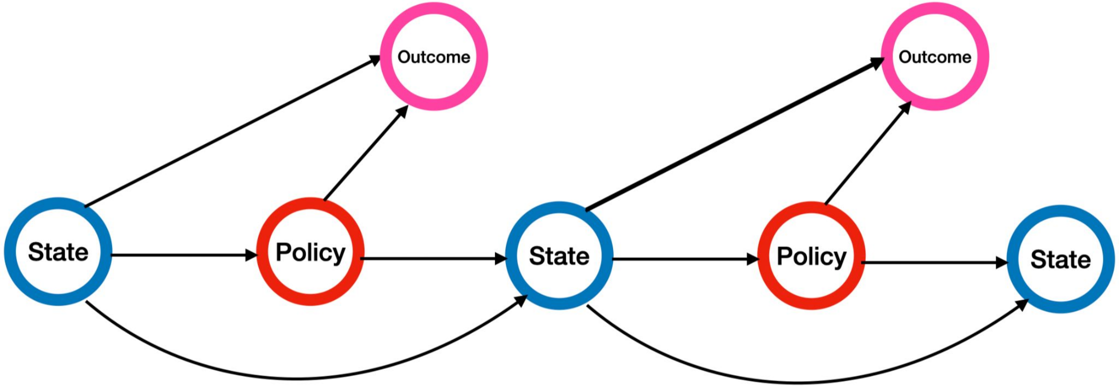
Challenge V: Weak signal

- Size of treatment effects ranges from 0.5% – 2% (Tang et al., 2019)
- Making it challenging to distinguish between new and old policies

Addressing Carryover Effects over Time

RL framework for A/B testing

- Employ **Markov decision processes** (MDPs) to model experimental data (Glynn et al., 2020, Farias et al., 2022, Shi et al., 2023a)
- Capture carryover effects over time using **dynamic system transitions**

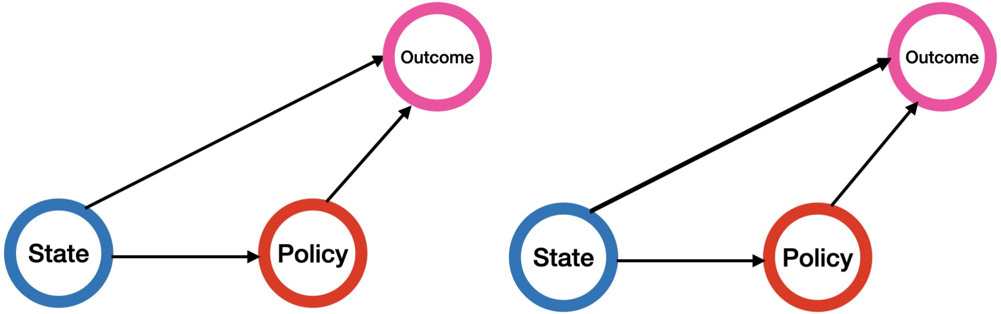


- Past policies impact future outcomes indirectly through future states
- Future states serve as **mediators** between past policies and future outcomes

Addressing Carryover Effects (Cont'd)

RL framework for A/B testing

- Most existing solutions require the independence assumption (see e.g., Larsen et al., 2024; Quin et al., 2024) failing to detect any carryover effect (see the numerical examples in Shi et al., 2023a).



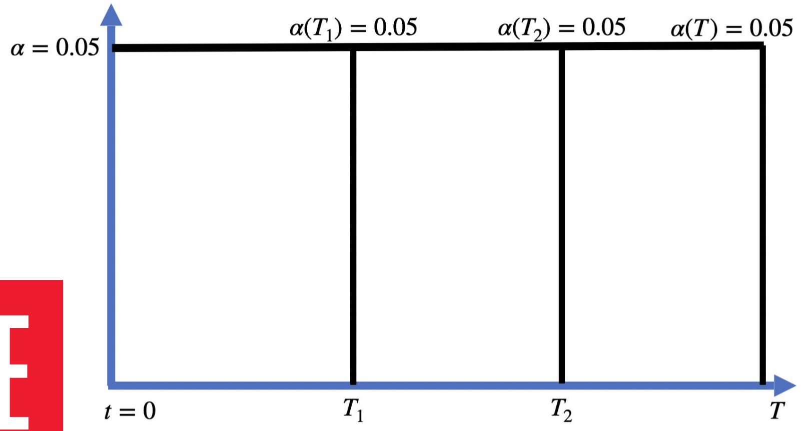
Addressing Spillover Effects, Partial Observability & Early Stopping

(MA)RL framework for A/B testing

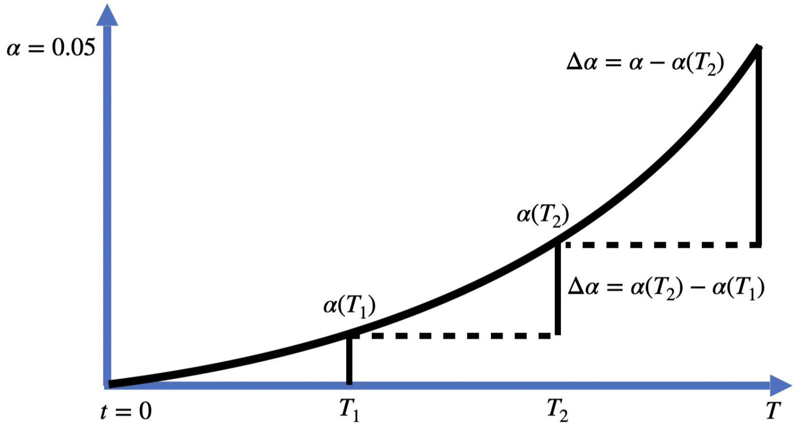
- Employ **multi-agent models** to capture **spillover effects** across units by the **interactions among agents** (Shi et al., 2023b)
- Employ **partially observable MDPs** (POMDPs) to capture **partial observability** (Liang and Recht, 2023; Sun et al., 2024)

Sequential monitoring

- Avoid p-value **peeking**
- Employ **sequential analysis** (e.g., **alpha-spending**) for A/B testing (Jennison et al., 2000)



with peeking



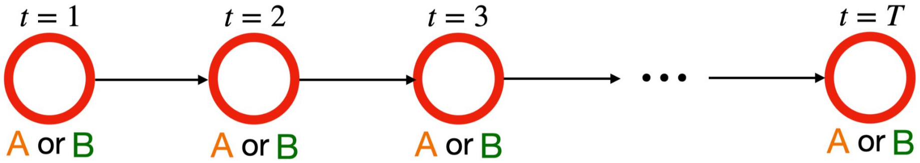
$\alpha = 0.05$
$\Delta\alpha = \alpha - \alpha(T_2)$
$\alpha(T_2)$
$\Delta\alpha = \alpha(T_2) - \alpha(T_1)$
$\alpha(T_1)$
$t = 0$
T_1
T_2
T

without peeking

Addressing Small Samples & Weak Signals

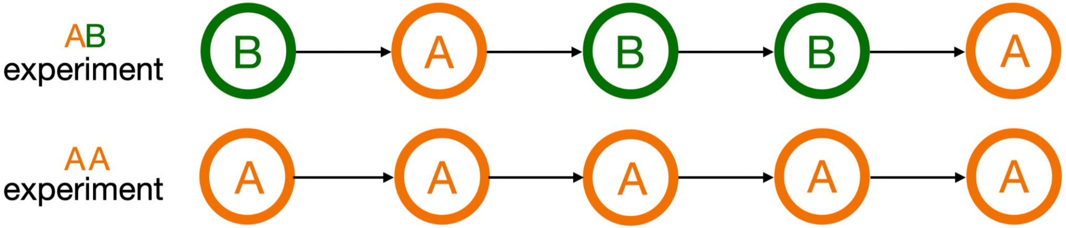
Design of experiments

- Identify **optimal treatment allocation strategy** in online experiments that **minimizes MSE of the ATE estimator**



Data integration

- Combine **experimental data** (**A**/**B**) with **historical data** (**A**/**A**) to **improve ATE estimation** (Li et al., 2023b, 2024b)



Overview

Challenges in A/B testing

- Interference effects over time/space
- Partial observability
- Early termination
- Small sample size
- Weak signal
- Solutions to these challenges

Design of online experiments

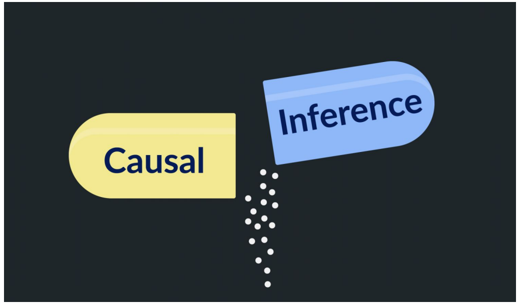
- Designs and trade-offs
- A selective review of optimal designs
- Case study in ridesharing

Policy Evaluation

- Direct method
- Importance sampling
- Double robust method
- Model-based method
- Uncertainty quantification

Off-policy Evaluation (OPE)

Objective: Evaluate the impact of a **target policy** using historical data generated from a different **behavior policy**



We will cover three settings

Settings	**carryover effects**	**spillover effects**
Contextual bandits	✗	✗
RL	✓	✗
MARL	✓	✓

OPE in Contextual Bandits

- A **widely-used** model in medicine and technological industries
- At each time **t**, the agent
  - Observes a **context**
  - Select an **action** (old policy **A** or new policy **B**)
  - Receives an **outcome**
- **Objective**: Given a sequence of i.i.d. **context**-**action**-**outcome** triplets generated by a **behavior policy**,

  b(**A**|**context**) = **Pr**(**action**=**A**|**context**) = 1 - **Pr**(**action**=**B**|**context**) = 1 - **b**(**B**|**context**)

  we aim to estimate the **ATE**: difference in expected **outcome** between **A** and **B**
- Common estimators (Dudik et al., 2014)
  - Direct estimator
  - Importance sampling (IS) estimator
  - Doubly robust (DR) estimator

Direct Estimator

- Let **r** denote the outcome regression (or reward) function

  r(**A**, **context**) = **E**(**outcome**|**action**=**A**, **context**), **r**(**B**, **context**) = **E**(**outcome**|**action**=**B**, **context**)

- ATE can be represented by

  E[**r**(**B**, **context**) - **r**(**A**, **context**)]

- The **direct estimator**:
  - Estimates **r** using supervised learning
  - Approximates the expectation **E** using the empirical context distribution
  - Plugs these estimators into the ATE formula

Importance Sampling Estimator

- The **IS estimator**
  - Estimates the behavior policy **b** using supervised learning
  - Reweights each **outcome** by the **IS ratio** that adjusts the distribution shift between the **target policy** and **b**

    ratio(**action**|**A**, **context**) = **I**(**action** = **A**)/**b**(**A**|**context**)
    ratio(**action**|**B**, **context**) = **I**(**action** = **B**)/**b**(**B**|**context**)

  - Averages these reweighted outcomes to estimate the ATE
- **Extensions**
  - When **b** is small for certain **action**-**context** pairs, IS suffers from a **large variance**
  - **Self-normalized** IS: normalize all IS ratios prior to reweighting
  - **Truncated** IS: truncate **b** from below prior to constructing IS ratios
- **Bias/variance trade-off**
  - Direct estimator suffers from **some bias**, as the outcome regression function **r** needs to be estimated from data
  - IS is **unbiased** when **b** is known as in randomized studies, but suffers from a **large variance**

Doubly Robust Estimator

- **Direct estimator** estimates the outcome regression function **r** to learn ATE
  - Its **consistency** requires consistent estimation of **r**
- **IS** estimates the behavior policy **b** to learn ATE
  - Its **consistency** requires consistent estimation of **b**
- **Doubly robust estimator** estimates **both r** and **b**
  - Its **consistency** requires consistent estimator of **either r or b**, but **not** necessarily both
  - It constructs the following estimating functions

    [**r**(**B**, **context**) - **r**(**A**, **context**)] + [**ratio**(**action**|**B**, **context**)- **ratio**(**action**|**A**, **context**)] (**outcome** - **r**(**action**, **context**))

  - The first term = estimating function in the direct estimator
  - The second term = **augmentation** term to debias the bias of the direct estimator
    - Offer additional robustness against misspecification of **r**
  - Averages these estimating function over the **context**-**action**-**outcome** triplets to estimate ATE

Fact 1: Double Robustness

- Recall the estimating function

  [**r**(**B**, **context**) - **r**(**A**, **context**)] + [**ratio**(**action**|**B**, **context**)- **ratio**(**action**|**A**, **context**)] (**outcome** - **r**(**action**, **context**))

- When **r** is correctly specified:
  - The second augmentation term is of mean zero
  - DR ≈ **direct estimator**, which becomes consistent
- When **b** is correctly specified:
  - The estimating equation has the same expected value to that of IS
  - DR ≈ **IS estimator**, which becomes consistent

Fact 2: Efficiency

- Recall the estimating function

  [**r**(**B**, **context**) - **r**(**A**, **context**)] + [**ratio**(**action**|**B**, **context**)- **ratio**(**action**|**A**, **context**)] (**outcome** - **r**(**action**, **context**))

- When **b** is correctly specified:
  - The estimating function is **unbiased** to the oracle ATE
  - DR's MSE becomes proportional to the **variance** of the estimating function
- Additionally, when **r** is correctly specified**:**
  - The estimating function achieves the **minimal** variance
  - A good working model for **r** improves DR's estimation efficiency
  - The DR estimator achieves the **efficiency bound** (e.g., smallest MSE among a wide class of regular estimators, Tsiatis, 2006)

Fact 3: Efficiency

- When **b** is estimated from data and the model is **correctly specified**:

  MSE(IS with an estimated **b**) <= MSE(IS with the oracle **b**)

- Estimating **b** yields a more efficient estimator, even if we know the oracle **b** (Tsiatis, 2006)
- The same holds true in RL settings (Hanna et al., 2019, 2021)
  - MSE of IS can be reduced through history-dependent IS estimation
  - The longer the history-length, the smaller the variance

Fact 4: Asymptotic Normality

- DR converges at a **parametric rate** (e.g., root-n rate) and remains **asymptotically normal** even when both the estimated **r** and **b** converge **slower** than the parametric rate
  - More specifically, it only requires both nuisance functions to converge faster than the **fourth-root rate**
- This enables us to apply modern deep/machine learning to estimate both **r** and **b**, leading to the **double machine learning** (DML) estimator (Chernozhukov et al., 2018)
  - **Cross-fitting** can be employed for valid statistical inference (e.g., hypothesis testing, confidence interval construction)
- Extensions of DML to RL: **Double reinforcement learning** (DRL, Kallus and Uehara, 2022, Liao et al., 2022)

OPE in Reinforcement Learning

- Focus on the **MDP** model (assuming full observability)
- **Objective**: Given an offline data consisting of a set of **state**-**action**-**reward**-**next-state** tuples generated by a behavior policy

  b(**A**|**state**) = **Pr**(**action**=**A**|**state**) = 1 - **Pr**(**action**=**B**|**state**) = 1 - **b**(**B**|**state**)

  We aim to estimate the **ATE**: the difference in the expected **return** between the two policies **A** and **B**

  return = reward at time 1 + **γ reward** at time 2 + **…** + **γ^t reward** at time t + **…**

  where **γ** denotes the discount factor (allowed to be 1).

- Common estimators (see Uehara et al., 2022 for a recent review):
  - Direct estimator
  - IS estimator
  - DR estimator
  - Model-based estimator

Direct Estimator

- Let **V**(**A**, **state**) **and V**(**B**, **state**) denote value functions (expected **return** starting from a given **state**) under the two **policies**.
- ATE can be represented by

  E[**V**(**B**, **initial state**) - **V**(**A**, **initial state**)]

- The **direct estimator**:
  - Estimates **V** using RL
    - **Fitted value** or **Q-evaluation** (Le et al., 2019)
    - **Least square temporal difference learning** (Sutton et al., 2008; Shi et al., 2022)
    - **RKHS**-based estimator (Liao et al., 2021)
  - Approximates the expectation **E** using the empirical initial state distribution
  - Plugs these estimators into the ATE formula

Importance Sampling Estimator

Sequential importance sampling (SIS, Zhang et al., 2013; Thomas et al., 2015)

- Estimates the behavior policy **b** using supervised learning
- At each time t, reweights the **reward** using the product of IS ratios to address the distributional shift from the initial time to t

ratio(**action** at time 1|**state** at time 1) × … × **ratio**(**action** at time t|**state** at time t)

- Averages these reweighted **rewards** to estimate the ATE
- Suffers from **curse of horizon** (Liu et al., 2018): Variance of the product of ratios grows exponentially fast wrt t
- Extension: **doubly robust estimator** (Jiang and Li, 2016; Thomas and Brunskill, 2016)

Marginalized importance sampling (MIS, Liu et al., 2018; Xie et al., 2019)

- Employ the structure of MDP (e.g., Markov assumption) to break the curse of horizon
- At each time t, reweights the **reward** using the marginalized IS ratio of both the **state** and **action** at time t

ratio(**state** at time t, **action** at time t)

computed via e.g., **minimax learning** (Uehara et al., 2020), **RKHS** (Liao et al., 2022)

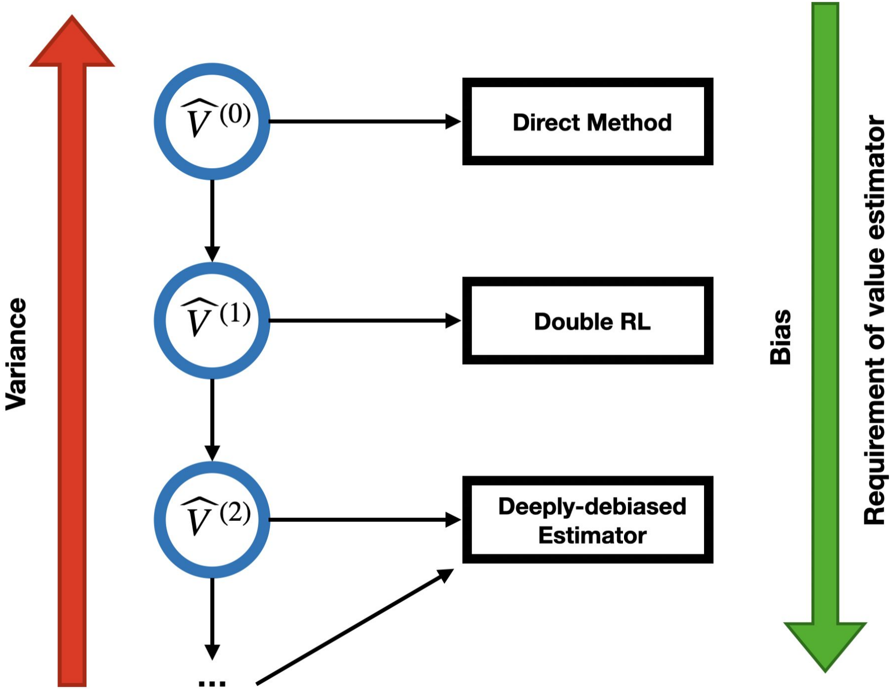
Double Reinforcement Learning

- Double RL extends double ML (Chernozhukov et al., 2018) from bandits to RL (Kallus and Uehara, 2022; Liao et al., 2022)
- Similar to DR, the estimator can be represented by

  Direct Estimator + **Augmentation Term**

- **Augmentation term** relies on the **MIS** ratio and is to
  - **debias** the bias of the direct estimator
  - offer protection against model misspecification of the Q- or value function
- **Fact 1**: DRL is **doubly robust**, e.g., consistent when **either** the value function **or** MIS ratio is correctly specified
- **Fact 2**: DRL achieves the **efficiency bound** in MDPs when **both** nuisance functions are correctly specified
- **Fact 3**: DRL is asymptotically normal when **both** converge faster than the fourth-root n rate
  - which facilitates hypothesis testing and calculation of p-values
- **Fact 4**: In addition to DRL, there exist efficient direct or MIS estimators as well
  - Direct estimators based on linear function approximation (Shi et al., 2022, 2023a) or RKHS (Liao et al., 2021)
  - MIS estimators based on linear function approximation = double RL estimator = direct estimator

Deeply-debiased OPE (Shi et al., 2021)



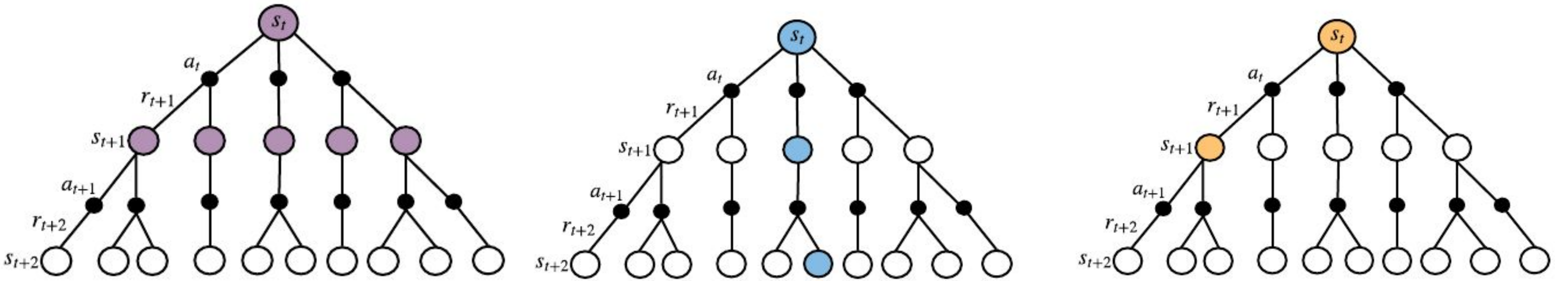
- Constructed based on **high-order influence functions** (Robins et al., 2008, 2017)
- Ensures bias decays to zero much faster than standard deviation to produce valid **p-values**
- Allows nuisance functions to converge at **arbitrary** rates

Model-based Estimator

- Direct, IS and DR are all **model-free** estimators
- **Model-based estimator** estimates the MDP model (reward & state transition function) from the data

  E (**reward**|**action**, **state**) & **P** (**next state**|**action**, **state**)

  and employs **dynamic programming** (DP), **Monte Carlo** (MC) method, or **temporal difference** (TD) learning for policy evaluation; see Sutton and Barto (2018) for a review of these methods



$DP : v(s_t) \leftarrow \mathbb{E}_\pi[r_{t+1} + \lambda v(s_{t+1})]$

$MC : v(s_t) \leftarrow v(s_t) + \eta(R_t - v(s_t))$

$TD : v(s_t) \leftarrow v(s_t) + \eta(r_{t+1} + \lambda v(s_{t+1}) - v(s_t))$

Uncertainty Quantification: A Selective Review

	Model-based	Direct method	Importance sampling	Double robust
Concentration inequalities		Feng et al. (2020)	Thomas et al. (2015)	Thomas et al. (2016) Jiang and Li (2016) Zhou et al. (2023)
Normal approximation		Luckett et al. (2020) Liao et al. (2021) Shi et al. (2022)	Wang et al. (2023)	Shi et al. (2021) Liao et al. (2022) Kallus and Uehara (2022)
Bootstrap	Hanna et al. (2017)	Hao et al. (2021)		Thomas et al. (2016) Hanna et al. (2017)
Empirical likelihood			Dai et al. (2020)	

Extensions

Policy evaluation under weak carryover effects

- Farias et al. (2022) proposed a **difference-in-Q** (DQ) estimator, a direct estimator under the assumption of weak carryover effect
- When compared against other direct estimators (e.g., Shi et al., 2023a):
  - DQ is an **on-policy** estimator that calculates the difference in Q-estimators under the **behavior policy**
  - The direct estimator by Shi et al. (2023a) is **off-policy** which computes Q-estimators under the **target policy**
  - On-policy estimator has smaller **variance** at the cost of a larger **bias** whose order of magnitude depends on the size of carryover effect

Policy evaluation in POMDPs

- **Model-based** methods based on **linear state-space models** (Liang and Recht, 2023; Sun et al., 2024)
- **Model-free** methods using **future-dependent value functions** (Uehara et al., 2023)

Policy evaluation in MARLs

- Adapt **mean-field approximation** designed for policy optimization (Yang et al., 2018) to OPE (Shi et al., 2023b)
- Employ **permutation-invariant** or **graph neural networks** to model spillover effects (Leung and Loupos, 2022; Dai et al., 2024)

Overview

Challenges in A/B testing

- Interference effects over time/space
- Partial observability
- Early termination
- Small sample size
- Weak signal
- Solutions to these challenges

Design of online experiments

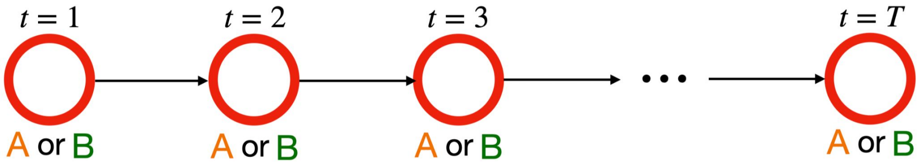
- Designs and trade-offs
- A selective review of optimal designs
- Case study in ridesharing

Policy Evaluation

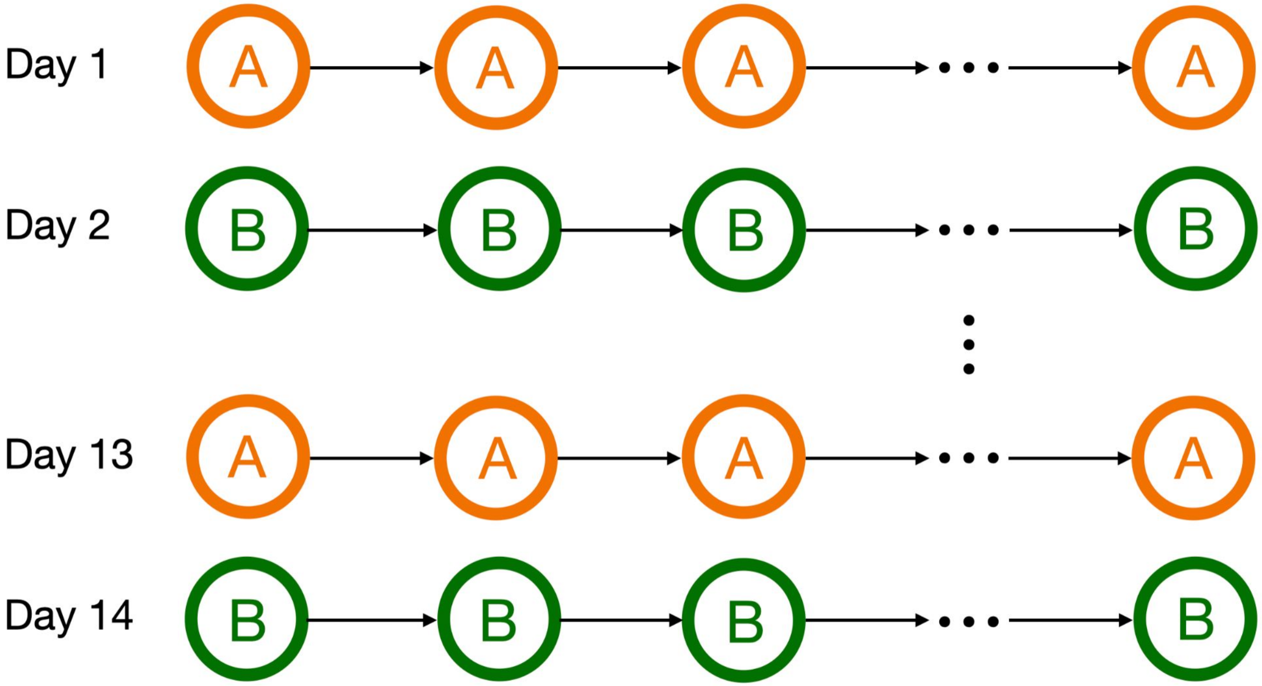
- Direct method
- Importance sampling
- Double robust method
- Model-based method
- Uncertainty quantification

Recap: Order Dispatching

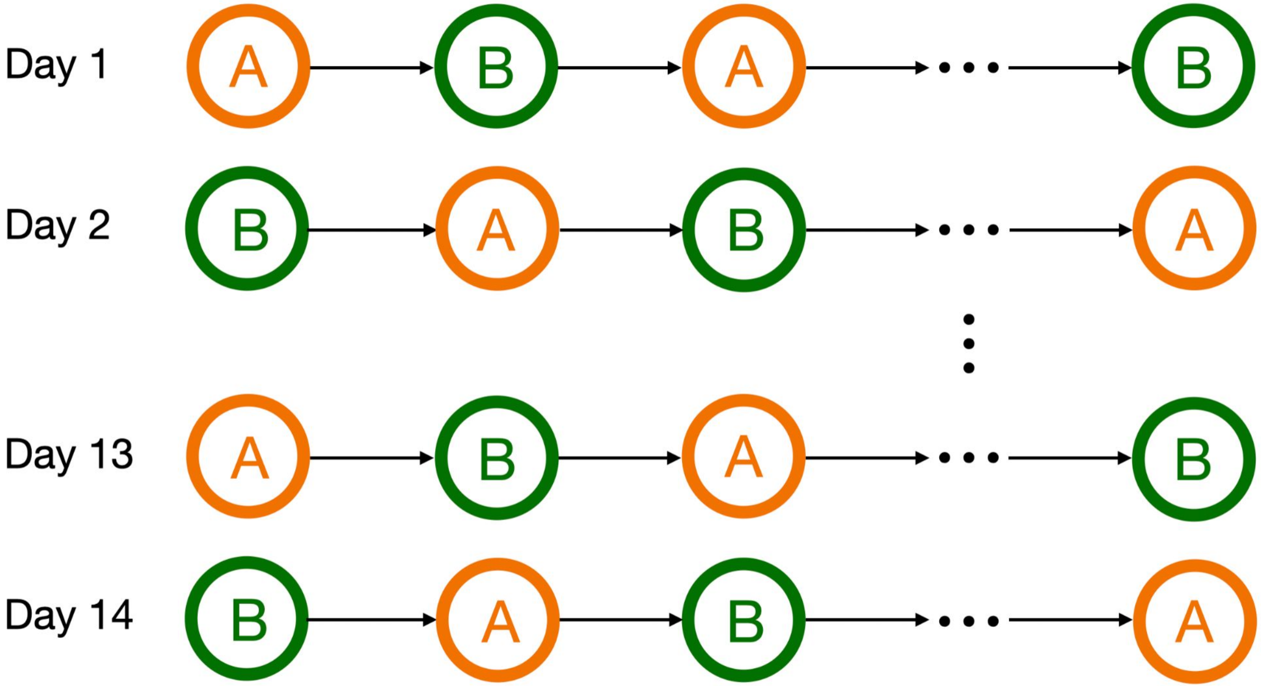
- Online experiments typically last for **two weeks**
- **30 minutes/1 hour** as one time unit, randomized over time
- Data forms a **time series**
- **Observations**:
  - **Outcome**: drivers' income or no. of completed orders
  - **Demand**: no. of call orders
  - **Supply**: no. of idle drivers
- **Treatment** (binary):
  - **New** order dispatching policy **B**
  - **Old** order dispatching policy **A**
- **Target**: **Average treatment effect** (ATE) = difference in average **outcome** between the **new** and **old** policy
- **Objective**: identify **optimal treatment allocation strategy** in online experiments that **minimizes MSE of the ATE estimator**



Alternating Day (AD)



Alternating Time (AT)



AD v.s. AT

Pros of AD

- Within each day, it is **on-policy** and avoids **distributional shift**, as opposed to off-policy designs (e.g., AT)
- On-policy designs are proven **optimal** in **fully observable Markovian** environments (Li et al., 2023)

Pros of AT

- Widely employed in ridesharing companies such as Lyft and Uber (Chamandy, 2016; Luo et al., 2024)
- According to my industrial collaborator, AT yields **less variable ATE estimators** than AD

Q: Why can off-policy designs, such as AT, be more efficient than AD?

A: Due to partial observability ...

A Bandit Example

- A **bandit** setting without carryover effects

  outcome = a I(action = A) + b I(action = B) + e

- ATE equals **b - a** and can be estimated by the sample mean estimator
  - average the outcome under the two policies and take the difference
- The resulting estimator's **MSE** under AD and AT is proportional to

$$\lim_{t\to\infty} \frac{1}{t}\mathrm{Var}(e_1+e_2+e_3+e_4+\cdots+e_t) \quad \text{and} \quad \lim_{t\to\infty} \frac{1}{t}\mathrm{Var}(e_1-e_2+e_3-e_4+\cdots-e_t)$$

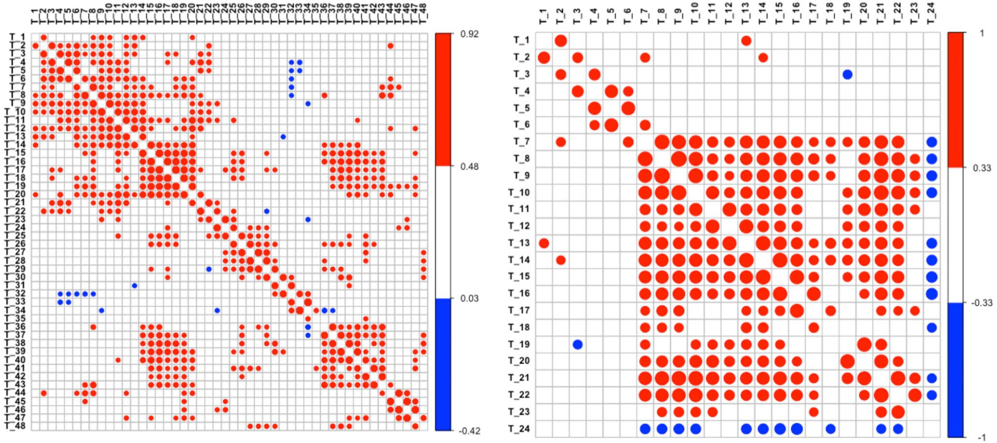
which depends on the residual correlation:

- With **uncorrelated residuals**, both designs yield same MSEs;
- With **positively correlated residuals**:
  - **AD assigns the same treatment** within each day, under which ATE estimator's variance inflates due to **accumulation** of residuals
  - **AT alternates treatments** for adjacent observations, effectively **negating** these residuals, leading to more efficient estimation
- With **negatively correlated residuals**, AD generally outperforms AT

When Can AT Be More Efficient than AD

Key condition: Residuals are positively correlated

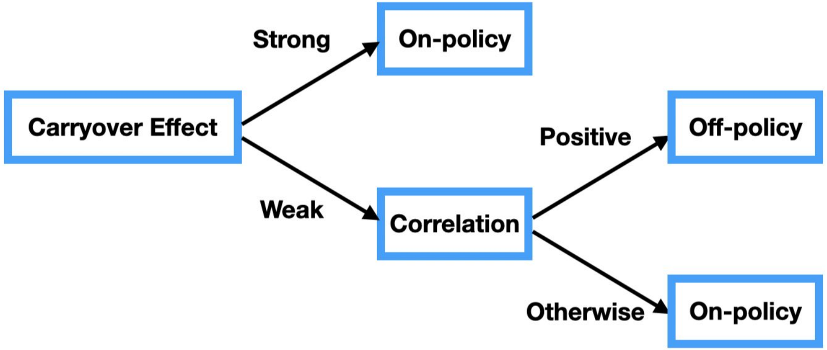
- **Often satisfied** in practice



- **Rule out full observability** (Markovanity) under which residuals are **uncorrelated**
- Can only be met under **partial observability**
- Suggest partial observability is more realistic, aligning with my collaborator’s finding

Designs and Trade-offs

- Previous analysis excludes carryover effects
- Trade-off between **on-policy** and **off-policy** designs (Wen et al., 2024; Xiong et al., 2024)
  - On-policy designs (e.g., AD or Li et al. 2023) are favored in settings with **large carryover effects** to **avoid distributional shifts**
  - Off-policy designs (e.g., AT or switchback) are preferred with **positively correlated residuals** for **variance reduction**



Carryover Effect

Strong

On-policy

Weak

Correlation

Positive

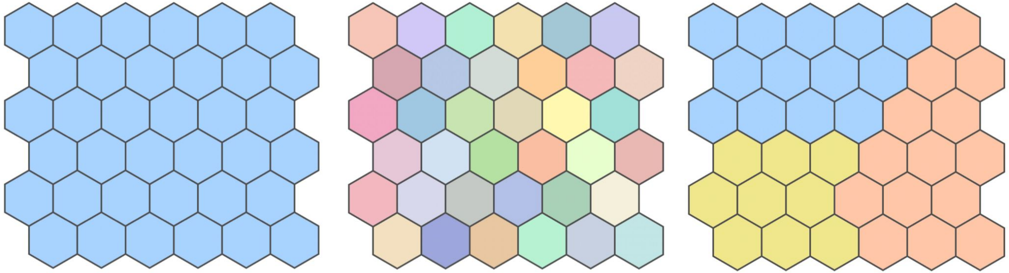
Off-policy

Otherwise

On-policy

Generalizations to Multi-unit Experiments

- **Global design**: Apply same policy to all units at each time and switch policies across time
- **Individual design**: Apply i.i.d. policies to all units at each time
- **Cluster-randomized design**: Group units into clusters; apply i.i.d. policies to all clusters at each time



(a) Global Design (b) Individual Design (c) Cluster-randomized Design

- Trade-offs among the three designs (Ugander et al., 2013; Leung, 2021; Viviano et al., 2023; Yang et al., 2024)
  - Global designs are **on-policy** and are favored in settings with **large spillover effects** to **avoid distributional shifts**
  - Individual designs are **off-policy** are preferred with **positively correlated residuals** across units for **variance reduction**
  - Cluster-randomized designs strike a balance among **interference** and **correlation**, often yielding the best performance

Optimal Designs in Time Series Experiments

MDP Design (Li et al., 2023) – Code available on GitHub

- **Proven optimal** in **Markovian** environments
- **Doubly robust method**: Employ DR for ATE estimation
- **On-policy**: Similar to AD, it assigns the same **policy** within each day and switch **policies** across days
- **Neyman-allocation**: No. of days assigned to **treatment** and **control** proportional to **the variance of daily return**

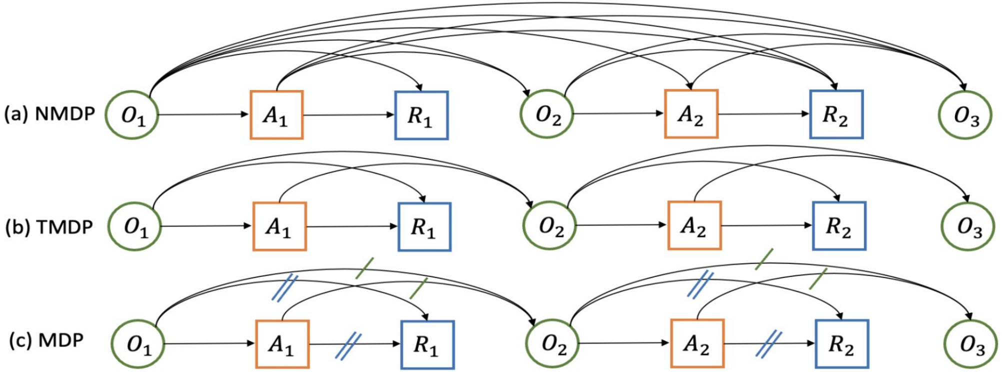


Figure 1: Data structure of NMDP, TMDP, and MDP. (a) In the NMDP, the reward and future observations are determined by all past observation-action pairs. (b) In the TMDP, the reward and future observations depend solely on the current observation-action pairs. (c) In the MDP, the reward and future observations also rely on the current observation-action pairs, with the same-colored slashes indicating identical conditional distributions.

Optimal Designs in Time Series Experiments

Switchback Design (Bojinov et al., 2023)

- **Minimax optimal** among the class of **regular switchback designs**
- **Sequential importance sampling** for ATE estimation – potentially suffering from curse of horizon
- **Off-policy**: Similar to AT, each **policy** is implemented for a specific duration and then switched to the other
- **Randomization frequency**: The optimal duration aligns with **the order of carryover effect**

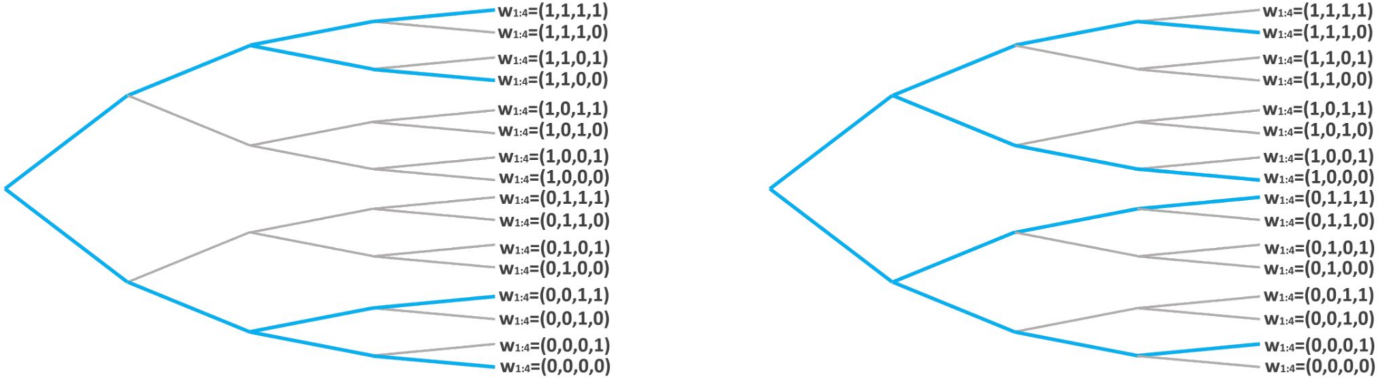


Figure 2 Two designs. The blue lines stand for the possible treatment assignments that a design could administer. Left: regular switchback experiment (Example 3); Right: irregular switchback experiment (Example 4).

Optimal Designs in Time Series Experiments

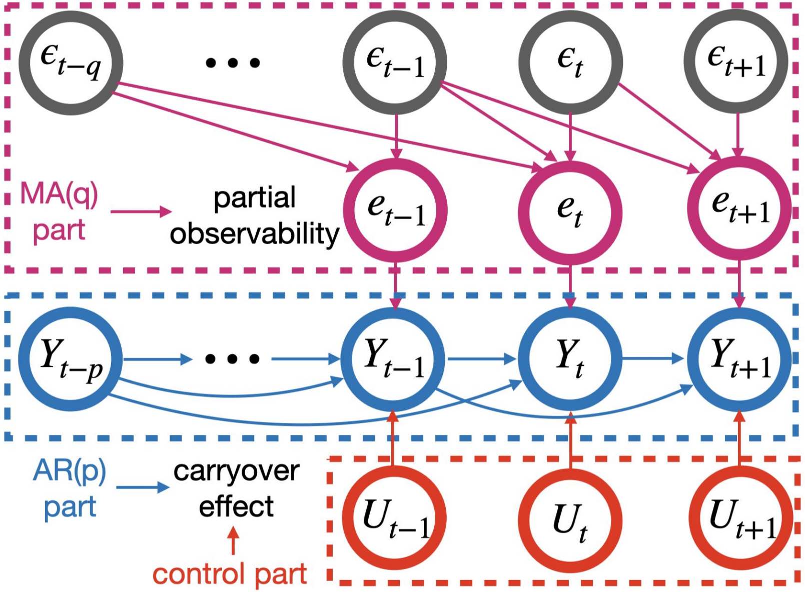
ARMA Design (Sun et al., 2024) – Code on GitHub

- **Proven optimal** in partially observable environments
- **Model-based method**: Employ classical **ARMA** model
  - Autoregressive model for **observations**
  - Moving average model for **residuals**
  - Control component to incorporate **policies**

    ➜ allow **carryover effects** & **partial observability**
- **Theory**: Establish **asymptotic MSEs** of ATE estimators

  ➜ compare different designs
- **Optimization**: Develop an **RL** algorithm

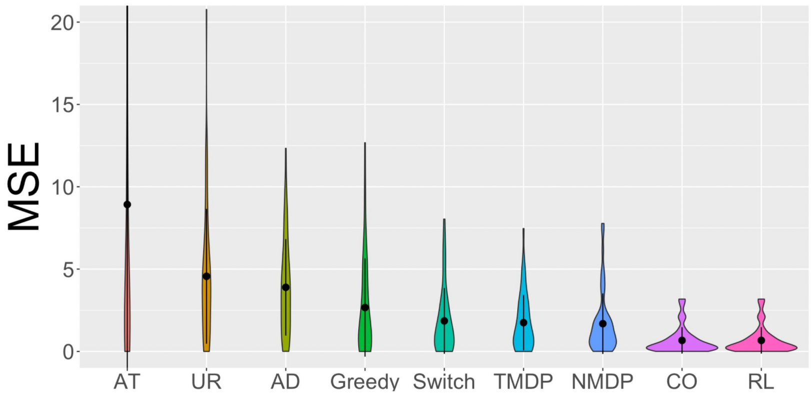
  ➜ compute the optimal design



Case Study: Order Dispatching (Code)

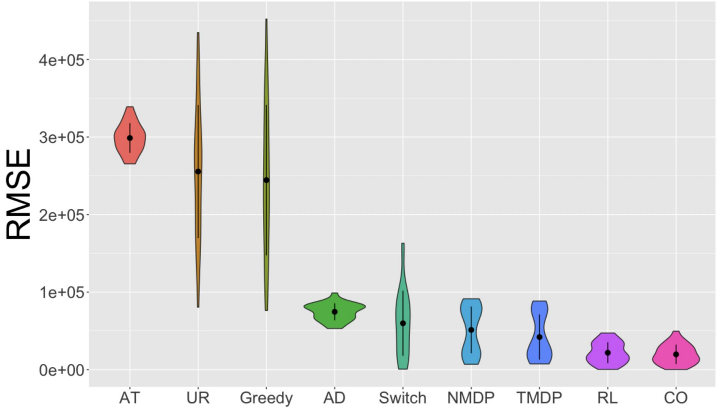
Experiment I: A Synthetic Dispatch Simulator

- Ridesharing environment over 9 × 9 spatial grid (code)
- **New policy**: MDP order dispatch policy (Xu et al., 2018)
- **Old policy**: distance-based policy



Experiment II: City-level Real-data-based Simulator

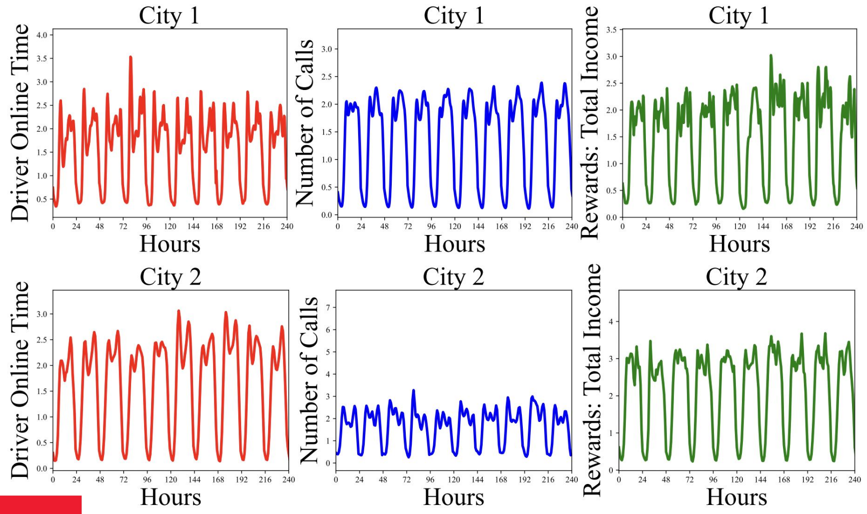
- City divided into 85 hexagonal regions (Tang et al., 2019)
- **Orders**: Generated according to the dataset
- **Drivers**: Behavior learned the dataset



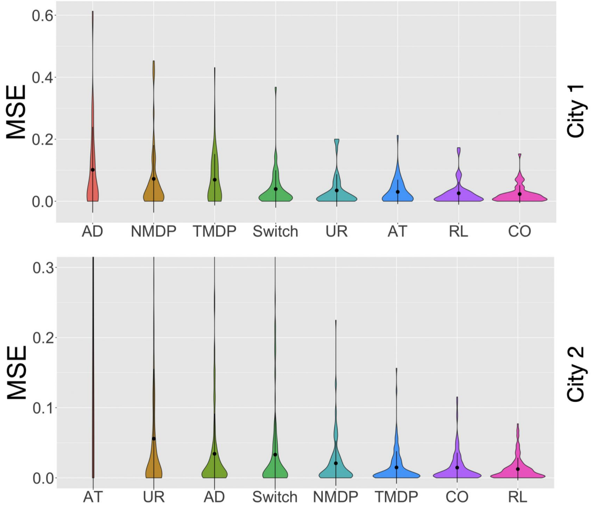
Case Study: Order Dispatching (Cont’d)

Experiment III: Real-data-based Analyses

- **Data** from two different cities



- **Bootstrap-based simulation**



References

A/B Testing

- Tsiatis, A. A. (2006). *Semiparametric theory and missing data* (Vol. 4). New York: Springer.
- Robins, J., Li, L., Tchetgen, E., & van der Vaart, A. (2008). Higher order influence functions and minimax estimation of nonlinear functionals. In *Probability and statistics: essays in honor of David A. Freedman* (Vol. 2, pp. 335-422). Institute of Mathematical Statistics.
- Sutton, R. S., Maei, H., & Szepesvári, C. (2008). A convergent o(n) temporal-difference algorithm for off-policy learning with linear function approximation. *Advances in neural information processing systems*, *21*.
- Ugander, J., Karrer, B., Backstrom, L., & Kleinberg, J. (2013, August). Graph cluster randomization: Network exposure to multiple universes. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 329-337).
- Zhang, B., Tsiatis, A. A., Laber, E. B., & Davidian, M. (2013). Robust estimation of optimal dynamic treatment regimes for sequential treatment decisions. *Biometrika*, *100*(3), 10-1093.
- Dudík, M., Erhan, D., Langford, J., & Li, L. (2014). Doubly robust policy evaluation and optimization. *Statistical science*, *29*(4), 485-511.
- Thomas, P., Theocharous, G., & Ghavamzadeh, M. (2015, February). High-confidence off-policy evaluation. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 29, No. 1).
- Jiang, N., & Li, L. (2016, June). Doubly robust off-policy value evaluation for reinforcement learning. In *International conference on machine learning* (pp. 652-661). PMLR.
- Chamandy N. (2016). Experimental in a ridesharing marketplace. https://eng.lyft.com/experimentation-in-a-ridesharing-marketplace-b39db027a66e

References

A/B Testing

- Thomas, P., & Brunskill, E. (2016, June). Data-efficient off-policy policy evaluation for reinforcement learning. In *International conference on machine learning* (pp. 2139-2148). PMLR.
- Hanna, J., Stone, P., & Niekum, S. (2017, February). Bootstrapping with models: Confidence intervals for off-policy evaluation. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 31, No. 1).
- ROBINS, J. M., LI, L., & MUKHERJEE, R. (2017). Minimax estimation of a functional on a structured high-dimensional model. *The Annals of Statistics*, *45*(5), 1951-1987.
- Chernozhukov, V., Chetverikov, D., Demirer, M., Duflo, E., Hansen, C., Newey, W., & Robins, J. (2018). Double/debiased machine learning for treatment and structural parameters. *The Econometrics Journal*, *21*(1), C1-C68.
- Liu, Q., Li, L., Tang, Z., & Zhou, D. (2018). Breaking the curse of horizon: Infinite-horizon off-policy estimation. *Advances in neural information processing systems*, *31*.
- Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. Cambridge: MIT press.
- Xu, Z., Li, Z., Guan, Q., Zhang, D., Li, Q., Nan, J., ... & Ye, J. (2018, July). Large-scale order dispatch in on-demand ride-hailing platforms: A learning and planning approach. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 905-913).
- Yang, Y., Luo, R., Li, M., Zhou, M., Zhang, W., & Wang, J. (2018, July). Mean field multi-agent reinforcement learning. In *International conference on machine learning* (pp. 5571-5580). PMLR.
- Hanna, J., Niekum, S., & Stone, P. (2019, May). Importance sampling policy evaluation with an estimated behavior policy. In *International Conference on Machine Learning* (pp. 2605-2613). PMLR.

References

A/B Testing

- Le, H., Voloshin, C., & Yue, Y. (2019, May). Batch policy learning under constraints. In *International Conference on Machine Learning* (pp. 3703-3712). PMLR.
- Xie, T., Ma, Y., & Wang, Y. X. (2019). Towards optimal off-policy evaluation for reinforcement learning with marginalized importance sampling. *Advances in neural information processing systems*, *32*.
- Dai, B., Nachum, O., Chow, Y., Li, L., Szepesvári, C., & Schuurmans, D. (2020). Coindice: Off-policy confidence interval estimation. *Advances in neural information processing systems*, *33*, 9398-9411.
- Feng, Y., Ren, T., Tang, Z., & Liu, Q. (2020, November). Accountable off-policy evaluation with kernel bellman statistics. In *International Conference on Machine Learning* (pp. 3102-3111). PMLR.
- Luckett, D. J., Laber, E. B., Kahkoska, A. R., Maahs, D. M., Mayer-Davis, E., & Kosorok, M. R. (2020). Estimating dynamic treatment regimes in mobile health using v-learning. *Journal of the american statistical association*.
- Uehara, M., Huang, J., & Jiang, N. (2020, November). Minimax weight and q-function learning for off-policy evaluation. In *International Conference on Machine Learning* (pp. 9659-9668). PMLR.
- Hanna, J. P., Niekum, S., & Stone, P. (2021). Importance sampling in reinforcement learning with an estimated behavior policy. *Machine Learning*, *110*(6), 1267-1317.
- Liao, P., Klasnja, P., & Murphy, S. (2021). Off-policy estimation of long-term average outcomes with applications to mobile health. *Journal of the American Statistical Association*, *116*(533), 382-391.
- Shi, C., Wan, R., Chernozhukov, V., & Song, R. (2021, July). Deeply-debiased off-policy interval estimation. In *International conference on machine learning* (pp. 9580-9591). PMLR.

References

A/B Testing

- Farias, V., Li, A., Peng, T., & Zheng, A. (2022). Markovian interference in experiments. *Advances in Neural Information Processing Systems*, *35*, 535-549.
- Kallus, N., & Uehara, M. (2022). Efficiently breaking the curse of horizon in off-policy evaluation with double reinforcement learning. *Operations Research*, *70*(6), 3282-3302.
- Leung, M. P. (2022). Rate-optimal cluster-randomized designs for spatial interference. *The Annals of Statistics*, *50*(5), 3064-3087.
- Leung, M. P., & Loupos, P. (2022). Graph Neural Networks for Causal Inference Under Network Confounding. *arXiv preprint arXiv:2211.07823*.
- Liao, P., Qi, Z., Wan, R., Klasnja, P., & Murphy, S. A. (2022). Batch policy learning in average reward markov decision processes. *Annals of statistics*, *50*(6), 3364.
- Shi, C., Zhang, S., Lu, W., & Song, R. (2022). Statistical inference of the value function for reinforcement learning in infinite-horizon settings. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, *84*(3), 765-793.
- Uehara, M., Shi, C., & Kallus, N. (2022). A review of off-policy evaluation in reinforcement learning. *arXiv preprint arXiv:2212.06355*.
- Bojinov, I., Simchi-Levi, D., & Zhao, J. (2023). Design and analysis of switchback experiments. *Management Science*, *69*(7), 3759-3777.
- Liang, T., & Recht, B. (2023). Randomization inference when n equals one. *arXiv preprint arXiv:2310.16989*.

References

A/B Testing

- Li, T., Shi, C., Wang, J., & Zhou, F. (2023a). Optimal treatment allocation for efficient policy evaluation in sequential decision making. *Advances in Neural Information Processing Systems*, *36*, 48890-48905.
- Li, X., Miao, W., Lu, F., & Zhou, X. H. (2023b). Improving efficiency of inference in clinical trials with external control data. *Biometrics*, *79*(1), 394-403.
- Shi, C., Wang, X., Luo, S., Zhu, H., Ye, J., & Song, R. (2023a). Dynamic causal effects evaluation in a/b testing with a reinforcement learning framework. *Journal of the American Statistical Association*, *118*(543), 2059-2071.
- Shi, C., Wan, R., Song, G., Luo, S., Zhu, H., & Song, R. (2023b). A multiagent reinforcement learning framework for off-policy evaluation in two-sided markets. *The Annals of Applied Statistics*, *17*(4), 2701-2722.
- Uehara, M., Kiyohara, H., Bennett, A., Chernozhukov, V., Jiang, N., Kallus, N., ... & Sun, W. (2023). Future-dependent value-based off-policy evaluation in pomdps. *Advances in neural information processing systems*, *36*, 15991-16008.
- Viviano, D., Lei, L., Imbens, G., Karrer, B., Schrijvers, O., & Shi, L. (2023). Causal clustering: design of cluster experiments under network interference. *arXiv preprint arXiv:2310.14983*.
- Wang, J., Qi, Z., & Wong, R. K. (2023). Projected state-action balancing weights for offline reinforcement learning. *The Annals of Statistics*, *51*(4), 1639-1665.
- Zhou, W., Li, Y., Zhu, R., & Qu, A. (2023). Distributional shift-aware off-policy interval estimation: A unified error quantification framework. *arXiv preprint arXiv:2309.13278*.

References

A/B Testing

- Dai, R., Wang, J., Zhou, F., Luo, S., Qin, Z., Shi, C., & Zhu, H. (2024). Causal Deepsets for Off-policy Evaluation under Spatial or Spatio-temporal Interferences. *arXiv preprint arXiv:2407.17910*.
- Larsen, N., Stallrich, J., Sengupta, S., Deng, A., Kohavi, R., & Stevens, N. T. (2024). Statistical challenges in online controlled experiments: A review of a/b testing methodology. *The American Statistician*, *78*(2), 135-149.
- Li, T., Shi, C., Lu, Z., Li, Y., & Zhu, H. (2024a). Evaluating dynamic conditional quantile treatment effects with applications in ridesharing. *Journal of the American Statistical Association*, *119*(547), 1736-1750.
- Li, T., Shi, C., Wen, Q., Sui, Y., Qin, Y., Lai, C., & Zhu, H. (2024b). Combining experimental and historical data for policy evaluation. *arXiv preprint arXiv:2406.00317*.
- Luo, S., Yang, Y., Shi, C., Yao, F., Ye, J., & Zhu, H. (2024). Policy evaluation for temporal and/or spatial dependent experiments. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, *86*(3), 623-649.
- Quin, F., Weyns, D., Galster, M., & Silva, C. C. (2024). A/B testing: A systematic literature review. *Journal of Systems and Software*, 112011.
- Sun, K., Kong, L., Zhu, H., & Shi, C. (2024). ARMA-Design: Optimal Treatment Allocation Strategies for A/B Testing in Partially Observable Time Series Experiments. *arXiv preprint arXiv:2408.05342*.
- Wen, Q., Shi, C., Yang, Y., Tang, N., & Zhu, H. (2024). An analysis of switchback designs in reinforcement learning. *arXiv preprint arXiv:2403.17285*.
- Xiong, R., Chin, A., & Taylor, S. J. (2024). Data-driven switchback experiments: Theoretical tradeoffs and empirical Bayes designs. *arXiv preprint arXiv:2406.06768*.
- Yang, Y., Shi, C., Yao, F., Wang, S., & Zhu, H. (2024). Spatially Randomized Designs Can Enhance Policy Evaluation. *arXiv preprint arXiv:2403.11400*.

LLM for Marketplaces

Tony Qin

foreva.ai (Ex Lyft, DiDi)

LLMs in Two-sided Marketplaces

What are LLMs?

- Transformer-based language models trained on vast amounts of text data.
- Capable of understanding and generating human-like text.
- Amplified by advancement in speech recognition and voice synthesis

Why LLMs in marketplaces?

- Enhance user experience through natural language interactions.
- Improve decision-making (e.g., pricing and incentives) by analyzing unstructured data (e.g., reviews, chat logs).
- Automate customer support, search/recommendations, and more.

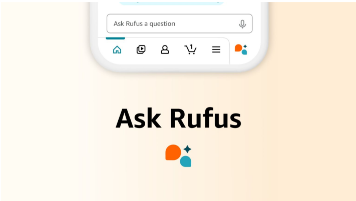
LLM-based Agents

Model

- Similar to RL agents but with natural language as communication vehicle

Examples

- Customer support
- Legal assistant
- Shopping guide
- Sales marketing
- Restaurant phone agent
- Coding

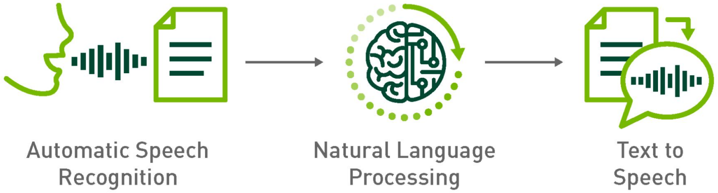


Ask Rufus a question

Ask Rufus

Voice agents

- Speech recognition (ASR)
- Cognitive layer (LLM)
- Voice synthesis (TTS)



https://developer.nvidia.com/blog/how-to-deploy-real-time-text-to-speech-applications-on-gpus-using-tensorrt/

Applications of LLMs in Two-sided Marketplaces

Customer support

- Automate handling of common inquiries (e.g., lost and found, complaints).
- Reduce response times and improve user satisfaction.

Personalized recommendation

- Use LLMs to analyze user preferences and suggest relevant products or services.
- Example: Recommending restaurants based on past orders and reviews.

Dynamic pricing and incentives

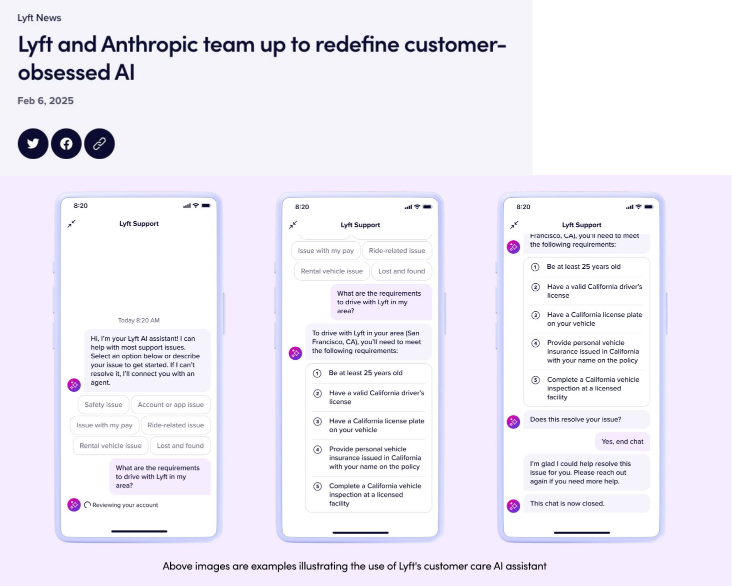
- Analyze unstructured data (e.g., social media, reviews) to adjust pricing strategies.
- Generate personalized incentives for users (e.g., discounts, promotions).

Fraud detection

- Use LLMs to detect fraudulent activities by analyzing text data (e.g., fake reviews, suspicious messages).

Case Studies

Customer support assistants



Lyft News

Lyft and Anthropic team up to redefine customer-obsessed AI

Feb 6, 2025

Today 8:20 AM

Hi, I'm your Lyft AI assistant! I can help with most support issues. Select an option below or describe your issue to get started. If I can't resolve it, I'll connect you with an agent.

What are the requirements to drive with Lyft in my area?

Reviewing your account

To drive with Lyft in your area (San Francisco, CA), you'll need to meet the following requirements:

1. Be at least 25 years old
2. Have a valid California driver's license
3. Have a California license plate on your vehicle
4. Provide personal vehicle insurance issued in California with your name on the policy
5. Complete a California vehicle inspection at a licensed facility

Does this resolve your issue?

Yes, end chat

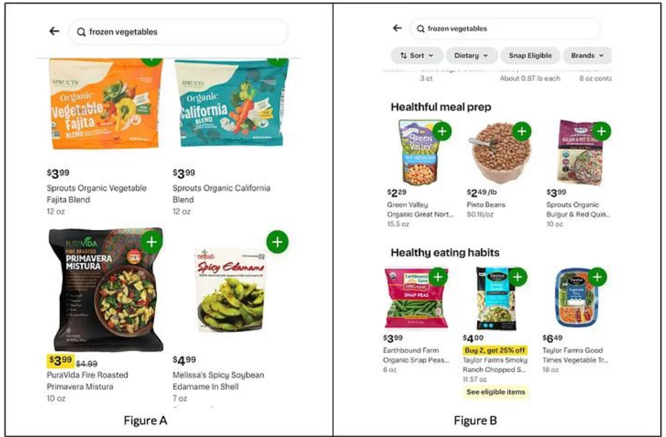
I'm glad I could help resolve this issue for you. Please reach out again if you need more help.

This chat is now closed.

Above images are examples illustrating the use of Lyft's customer care AI assistant

Discovery in search

- https://tech.instacart.com/supercharging-discovery-in-search-with-llms-556c585d4720
- Inspirational and discovery-driven content



$3.99 Sprouts Organic Vegetable Fajita Blend 12 oz

$3.99 Sprouts Organic California Blend 12 oz

$3.99 ~~$4.99~~ PuraVida Fire Roasted Primavera Mistura 10 oz

$4.99 Melissa's Spicy Soybean Edamame In Shell 7 oz

Figure A

Healthful meal prep

$2.29 Green Valley Organic Great Nort... 15.5 oz

$2.49 /lb Pinto Beans $0.16/oz

$3.99 Sprouts Organic Bulgur & Red Quin... 10 oz

Healthy eating habits

$3.99 Earthbound Farm Organic Snap Peas... 6 oz

$4.00 Buy 2, get 25% off Taylor Farms Smoky Ranch Chopped S... 11.57 oz

$6.49 Taylor Farms Good Times Vegetable Tr... 18 oz

Figure B

Example of how we inspire users with new ideas around healthy eating habits for the search query 'frozen vegetables': Figure A shows highly relevant results that directly match the user's intent. Figure B presents inspirational products like grains and beans that pair well with frozen vegetables. These products are presented as carousels with clearly explained titles, highlighting complementary combinations like "Healthful meal prep" to encourage creative cooking. Similarly, we also offer substitute options like fresh vegetables, salad kits, and vegetable trays. These products are presented as a carousel titled 'Healthy Eating Habits,' encouraging users to explore diverse and nutritious ways to incorporate vegetables into their daily meals

Challenges & Future Directions

Challenges

- Accuracy in agent response: hallucination, logical reasoning
- Bias and fairness: e.g., avoid biased recommendations or responses
- Scalability and compute: handling large volumes of real-time interactions

Future directions

- Multimodal agents: combining text, voice, images
- Real-time adaptation: agents that adapt to changing marketplace dynamics, continual learning

Live Demo

Eva AI
is calling
Accept
Decline

Food ordering from restaurant foreva.ai [Qin & Zhou, AAMAS 2025]

Marketplace Simulation

Tony Qin

foreva.ai (Ex Lyft, DiDi)

Overview

Why simulation?

- Test policies (e.g., pricing, matching) without real-world risks.
- Understand marketplace dynamics under different “what-if” scenarios.

Types of simulations

- Macroscopic: High-level modeling of marketplace dynamics.
- Microscopic: Detailed modeling of individual agents (e.g., drivers, riders) - agent-based modeling.

Modeling

Macroscopic modeling

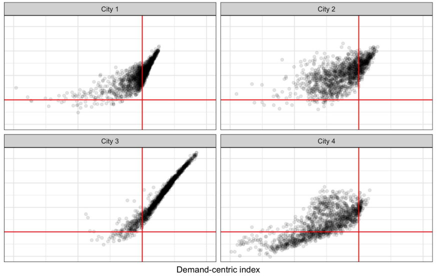
- To understand impact of interventions in supply and demand on marketplace efficiency

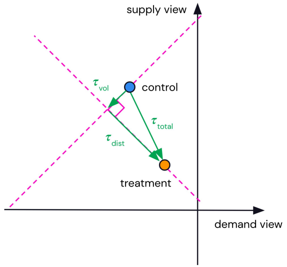
Graphic equilibrium metrics (GEM)

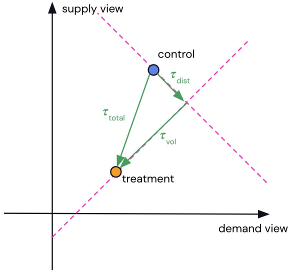
- [Zhou et al., 2021] generalized asymmetric Wasserstein distance between supply and demand
- Dispatch effects accounted for through solving an optimal transport problem

Dual-perspective framework for two-sided marketplaces

- [Chin & Qin, 2023] Supply-demand gap index derived from GEM as expected market condition from a random rider (buyer) or driver (seller) perspective
- Shift in the dual-view indices offer insights on changes in marketplace efficiency.







Modeling

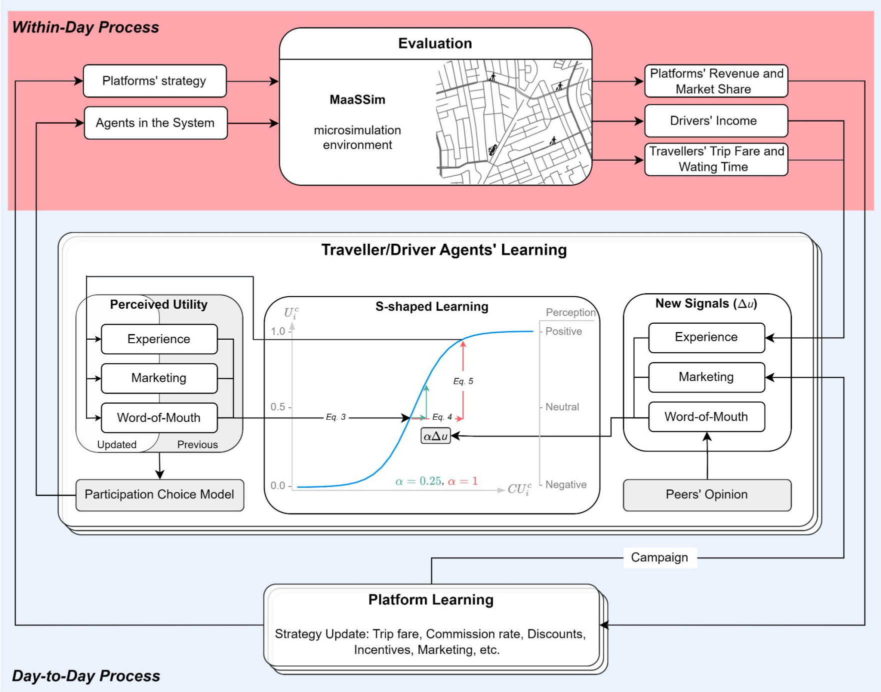
Microscopic modeling

- Dynamics and growth of marketplace through modeling individual agents
- *Modeling the Rise and Fall of Two-Sided Markets*: Ghasemi and Kucharski (2024)
- Uses MaaSSim

Agent participation model

- Choice models over transportation modes and participating (competing) platforms
- Endogenous factors: driver income, rider waiting time, price
- Exogenous factors: marketing, word-of-mouth
- S-shaped learning and adaptation (faster at neutral util)

Evaluating platform policies



Modeling

Generative World Model

- Neural network-based foundation model that creates realistic simulations of market dynamics
- Enables study of complex interactions between participants

MarS

- [Li et al., 2025] MarS: a *Financial Market Simulation Engine Powered by Generative Foundation Model*.
- A financial market simulation engine powered by a Large Market Model (LMM)
- Generating order-level data that reflects actual market behavior, facilitating the testing and development of trading strategies in a controlled environment
- https://arxiv.org/pdf/2409.07486

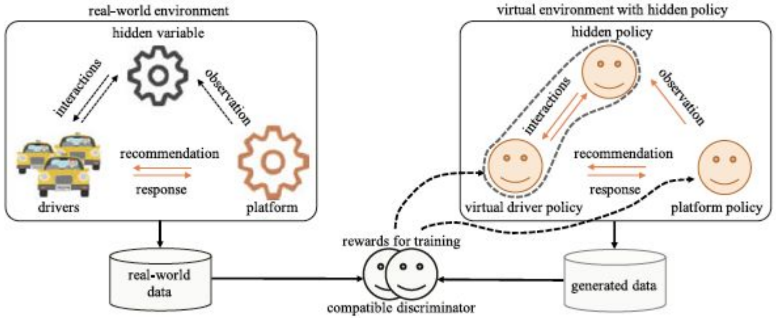
Modeling

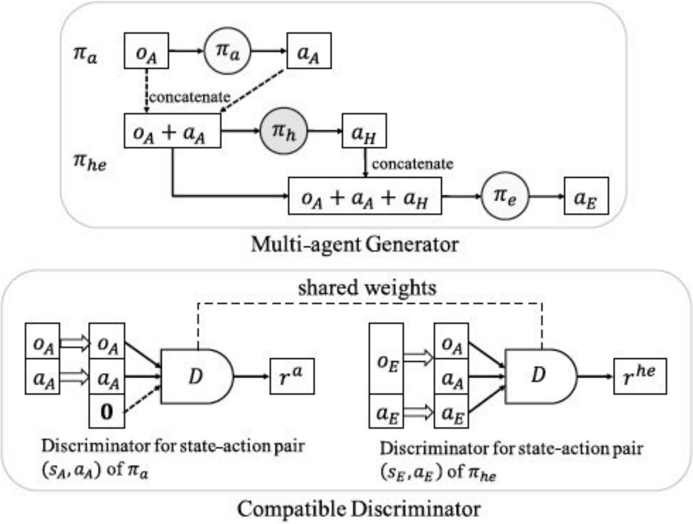
Generative Adversarial Imitation Learning

- Simulate driver behavior for target-based driver incentives
- Explicitly models confounder policy to simulate competitor effects

Generators

- Platform actions, driver and competitor actions





[Shang et al., 2019, 2021]

Tools & Frameworks | Practical Considerations

Complexity

- Assess the intricacy of interactions between participants and choose a tool that can capture these dynamics effectively.

Scalability

- Ensure the framework can handle the scale of your simulation, especially if modeling large marketplaces.

Programming Expertise

- Select a tool that aligns with your team's programming skills to facilitate efficient model development.

Specificity to domain

- While general-purpose ABM tools offer flexibility, domain-specific simulators like MaaSSim can provide tailored features for particular types of marketplaces.

Tools & Frameworks

Repast4Py

Repast

- A suite of advanced ABM tools that support the creation of agent-based simulations in various domains.
- Open source
- https://repast.github.io/repast4py.site/index.html

Variations

- **Repast Simphony:** Designed for standard modeling tasks, providing a rich set of features for building and analyzing simulations.
- **Repast HPC:** Tailored for high-performance computing scenarios, enabling the simulation of large-scale models.

Use case

- Suitable for simulating complex systems, including two-sided marketplaces, where understanding the interactions between different agent types is crucial.

Tools & Frameworks

Simulating rideshare dynamics

- Vehicle capacity: ride-hailing vs ride-pooling
- pre-/post-matching rider cancellation behavior
- Driver acceptance/rejection cancellation behavior
- Rider and driver participations

[Yao and Bekhor, 2021]

- “Ridesharing”: hitch service

[Chaudhari et al., 2020]

- OpenAI Gym-compatible

AMoDeus [Ruch et al., 2018], MATsim [Axhausen et al., 2016]

- Java-based simulation
- GUI and visualization tools

Tools & Frameworks

MaaSSim (Mobility as a Service Simulator)

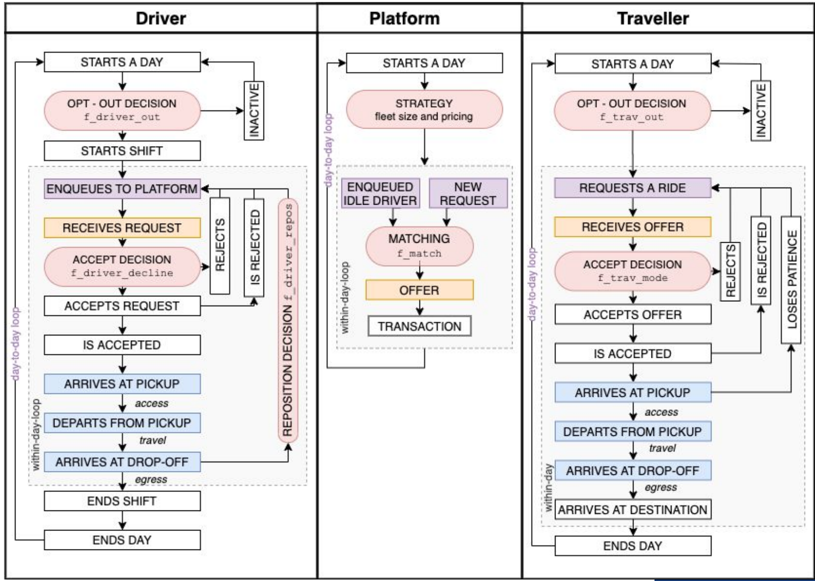
- MaaSSim [Kucharski & Cats, 2020] is an agent-based simulator specifically designed to model mobility services.
- Used in Modelling the Rise and Fall of Two-Sided Mobility Markets with Microsimulation discussed previously

Key features

- It allows for the simulation of day-to-day dynamics in two-sided mobility markets, capturing the decision-making processes of both service providers and consumers.

Use cases

- Analyzing market entry strategies
- Understanding the co-evolutionary behavior of agents in the mobility domain



Practical Challenges in Simulation

Scalability

- Simulating large-scale marketplaces with thousands of agents.

Realism / Fidelity

- Ensuring simulations reflect real-world dynamics.
- Human behaviors

Validation

- Comparing simulation results with real-world data.
- On-policy validation

Industry settings

- Nuances coming from the production systems
- Maintainability

References

Marketplace simulation

- Zhou, F., Luo, S., Qie, X., Ye, J. and Zhu, H., 2021. Graph-based equilibrium metrics for dynamic supply–demand systems with applications to ride-sourcing platforms. *Journal of the American Statistical Association*, *116*(536), pp.1688-1699.
- Chin, A. and Qin, Z., 2023, November. A unified representation framework for rideshare marketplace equilibrium and efficiency. In *Proceedings of the 31st ACM International Conference on Advances in Geographic Information Systems* (pp. 1-11).
- Ghasemi, F. and Kucharski, R., 2024, May. Modelling the rise and fall of two-sided markets. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems* (pp. 679-687).
- Li, J., Liu, Y., Liu, W., Fang, S., Wang, L., Xu, C. and Bian, J., 2024. MarS: a Financial Market Simulation Engine Powered by Generative Foundation Model. *arXiv preprint arXiv:2409.07486*.
- Shang, W., Yu, Y., Li, Q., Qin, Z., Meng, Y. and Ye, J., 2019, July. Environment reconstruction with hidden confounders for reinforcement learning based recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 566-576).
- Shang, W., Li, Q., Qin, Z., Yu, Y., Meng, Y. and Ye, J., 2021. Partially observable environment estimation with uplift inference for reinforcement learning based recommendation. *Machine Learning*, *110*(9), pp.2603-2640.
- Yao, R. and Bekhor, S., 2021. A ridesharing simulation platform that considers dynamic supply-demand interactions. *arXiv preprint arXiv:2104.13463*.

References

Marketplace simulation

- Chaudhari, H.A., Byers, J.W. and Terzi, E., 2020, December. Learn to earn: Enabling coordination within a ride-hailing fleet. In *2020 IEEE International Conference on Big Data (Big Data)* (pp. 1127-1136). IEEE.
- Ruch, C., Hörl, S. and Frazzoli, E., 2018, November. Amodeus, a simulation-based testbed for autonomous mobility-on-demand systems. In *2018 21st international conference on intelligent transportation systems (ITSC)* (pp. 3639-3644). IEEE.
- W Axhausen, K., Horni, A. and Nagel, K., 2016. *The multi-agent transport simulation MATSim* (p. 618). Ubiquity Press.
- Kucharski, R. and Cats, O., 2020. MaaSSim--agent-based two-sided mobility platform simulator. *arXiv preprint arXiv:2011.12827*.

LLM for marketplaces

- Qin. Z. and Zhou. J., 2025. Eva: An LLM-based Multilingual Voice-agent Network for Restaurant Operations. To appear in *Proceedings of the 2025 International Conference on Autonomous Agents and Multi-agent Systems*.

Thank you

February 26, 2025

https://twosidedm.github.io/

zq2107@caa.columbia.edu (Tony)

c.shi7@lse.ac.uk (Chengchun)

htzhu@email.unc.edu (Hongtu)