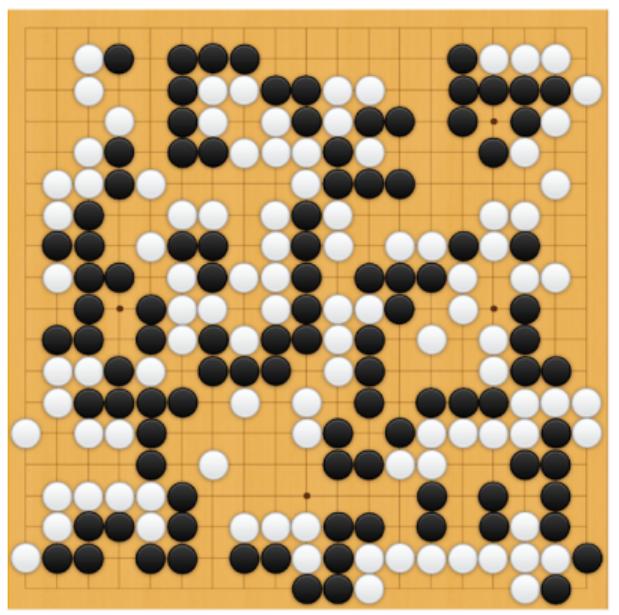


Reinforcement Learning in Nonstationary Environments

Chengchun Shi

Assistant Professor of Data Science
London School of Economics and Political Science

Developing AI with Reinforcement Learning



The image shows a Go board with black and white stones. On the right side, there is a banner with the text "THE ULTIMATE GO CHALLENGE" and "GAME 3 OF 3" above the date "27 MAY 2017". Below the banner, there is a black dot, a circular icon containing a blue neural network logo, the text "vs", a circular icon containing a portrait of a man (Ke Jie), and a white dot. To the left of the circular icons, there is a yellow trophy icon and the text "AlphaGo" followed by "Winner of Match 3". At the bottom, there is a large button with the text "RESULT B + Res".

THE ULTIMATE GO CHALLENGE
GAME 3 OF 3
27 MAY 2017

vs

AlphaGo
Winner of Match 3

Ke Jie

RESULT B + Res

Project I

Testing Stationarity and Change Point Detection in Reinforcement Learning

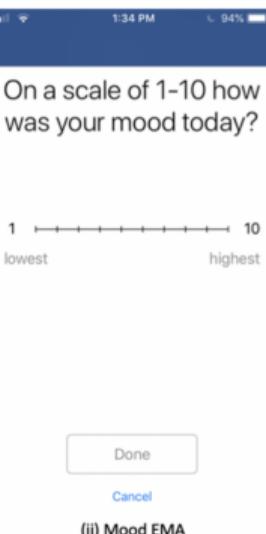
Joint work with Mengbing Li, Zhenke Wu and Piotr Fryzlewicz

Intern Health Study (IHS)

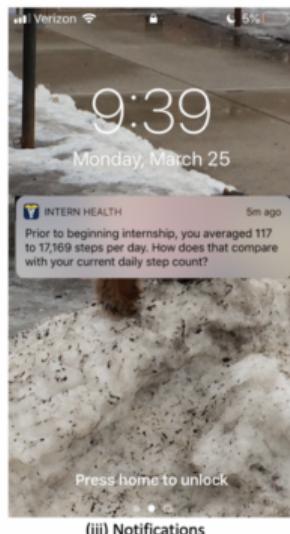
- **Data:** Intern Health Study (NeCamp et al., 2020)
- **Subject:** First-year medical interns working in stressful environments (e.g., long work hours and sleep deprivation)
- **Objective:** Promote physical well-being
- **Intervention:** Determine whether to send certain text message to a subject



(i) App Dashboard



(ii) Mood EMA



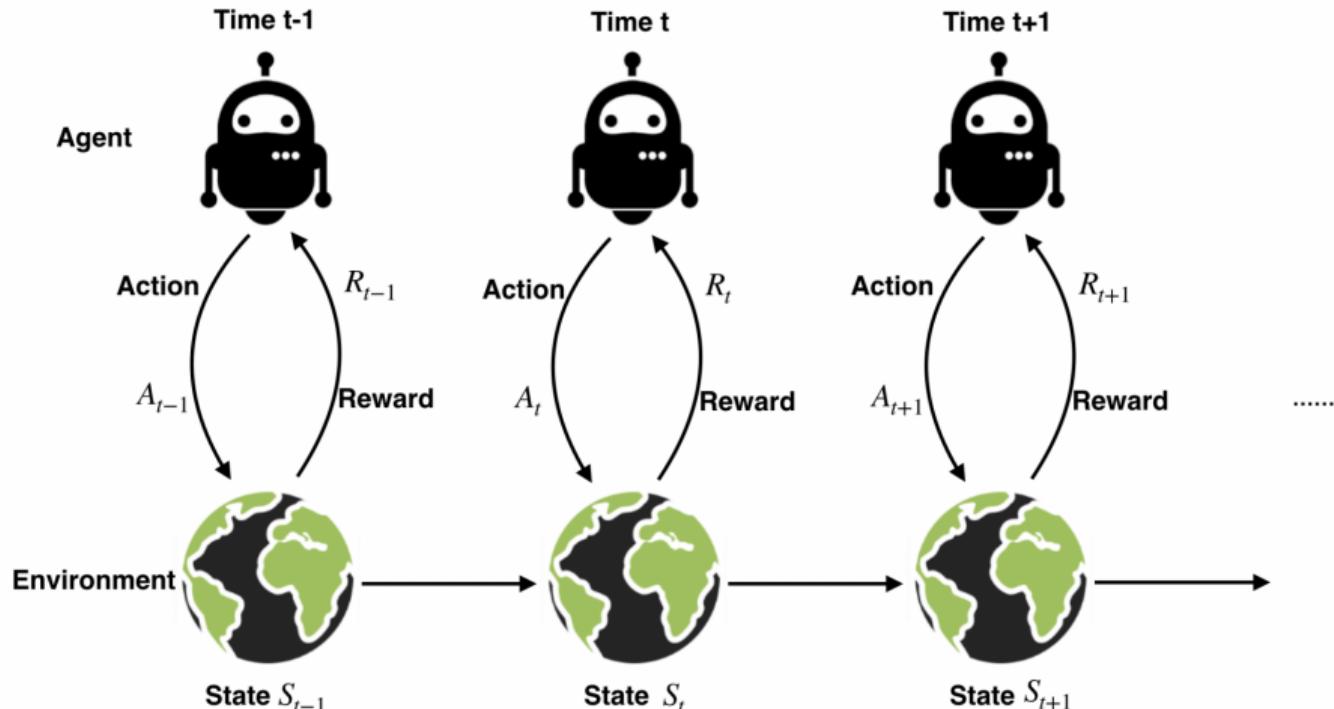
(iii) Notifications

Intern Health Study (Cont'd)

Table 1. Examples of 6 different groups of notifications.

Notification groups	Life insight	Tip
Mood	Your mood has ranges from 7 to 9 over the past 2 weeks. The average intern's daily mood goes down by 7.5% after intern year begins.	Treat yourself to your favorite meal. You've earned it!
Activity	Prior to beginning internship, you averaged 117 to 17,169 steps per day. How does that compare with your current daily step count?	Exercising releases endorphins which may improve mood. Staying fit and healthy can help increase your energy level.
Sleep	The average nightly sleep duration for an intern is 6 hours 42 minutes. Your average since starting internship is 7 hours 47 minutes.	Try to get 6 to 8 hours of sleep each night if possible. Notice how even small increases in sleep may help you to function at peak capacity & better manage the stresses of internship.

Sequential Decision Making



Objective: find an optimal policy that maximizes the cumulative reward

Reinforcement Learning

- **RL algorithms:** trust region policy optimization (Schulman et al., 2015), deep Q-network (DQN, Mnih et al., 2015), asynchronous advantage actor-critic (Mnih et al., 2016), quantile regression DQN (Dabney et al., 2018).
- **Foundations** of RL:
 - **Markov decision process** (MDP, Puterman, 1994): ensures the optimal policy is *stationary*, and is *not* history-dependent.
 - **Markov assumption:** conditional on the present (e.g., S_t , A_t), the future (R_t , S_{t+1}) and the past data history are independent
 - **Stationarity assumption:** the Markov transition kernel, e.g., the conditional distribution of (R_t , S_{t+1}) given ($S_t = s$, $A_t = a$) is stationary over time

Stationarity Assumption

- Stationarity assumption is likely to hold in many **OpenAI Gym** environments
- However, it can be violated in the **real world** environment
- Treatment effects can be **nonstationary**
 - COVID vaccine effectiveness decays over time
 - The treatment effect of activity suggestions may transition from positive to negative
- Environments can be **nonstationary**
 - COVID mutations, invention of vaccines
 - In the context of mobile-delivered prompts, the longer a person is under intervention, the more they may habituate to the prompts or become overburdened
- Without stationarity, the optimal policy is **nonstationary** as well
- Crucial for policy maker to take nonstationarity into account

Challenges

- When the optimal policy is **nonstationary**, using all data is not reasonable
- Natural to use **more recent observations** for policy optimisation
- **Challenging** to select the most recent **best data “segment”** of stationarity
 - Including too many past observations yields a suboptimal policy
 - Using only a few recent observations results in a very noisy policy

Contributions

- **Methodologically**
 - First work on developing consistent test for stationarity in offline RL
 - The test procedure is “**model-free**” (target on the optimal Q-function Q^{opt})
 - Null hypothesis \mathcal{H}_0 : Q^{opt} is stationary over time
 - Alternative hypothesis \mathcal{H}_1 : Q^{opt} varies over time
 - Sequentially apply the test for selecting the **best data “segment”**
- **Empirically**
 - Identify a **better** policy compared to existing RL algorithms in IHS
- **Theoretically**
 - prove our test has good **size** and **power** properties under a **bidirectional** asymptotic framework

Method: Test Statistics

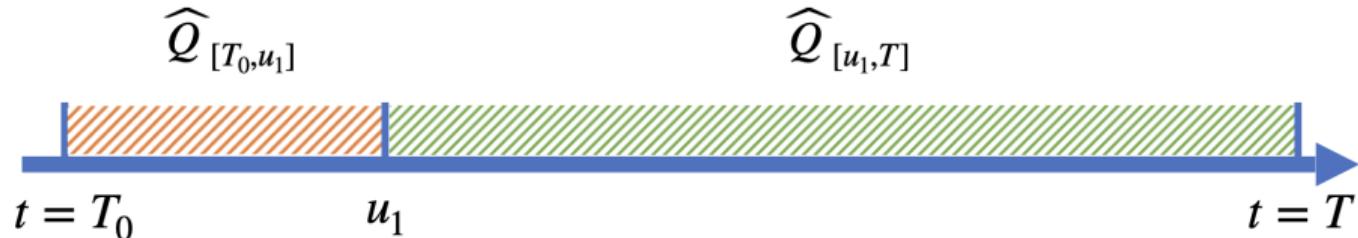
Some key components of the test statistic:

- Model the optimal Q-function via the **sieve** method
 - Ensure the estimator has a tractable limiting distribution
 - Increase the number of sieves to reduce the bias resulting from model misspecification
- Construct **CUSUM**-type test statistics for change detection (detailed later)
 - Widely used in the time series literature
- Obtain critical values using **multiplier bootstrap**
 - Q-estimator is asymptotically normal
 - Test statistic is a complicated function of several Q-estimators
 - Bootstrapped statistic is a function of simulated random normal errors
 - Approximate critical values via the quantile of the bootstrapped statistic

Method: Test Statistics (Cont'd)

- A CUSUM-type test statistic

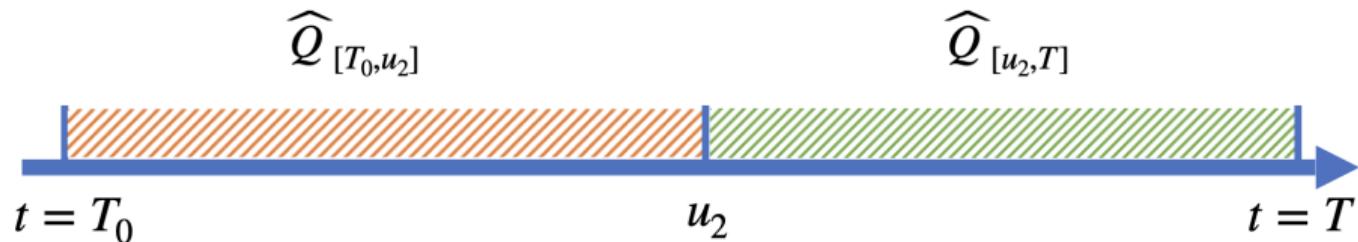
- Select a set of candidate change point locations $\boldsymbol{u} \in [T_0, T]$
- For each \boldsymbol{u} , estimate two Q-functions $\widehat{Q}_{[T_0, \boldsymbol{u}]}$ and $\widehat{Q}_{[\boldsymbol{u}, T]}$
- Construct the test based on their maximal difference



Method: Test Statistics (Cont'd)

- A CUSUM-type test statistic

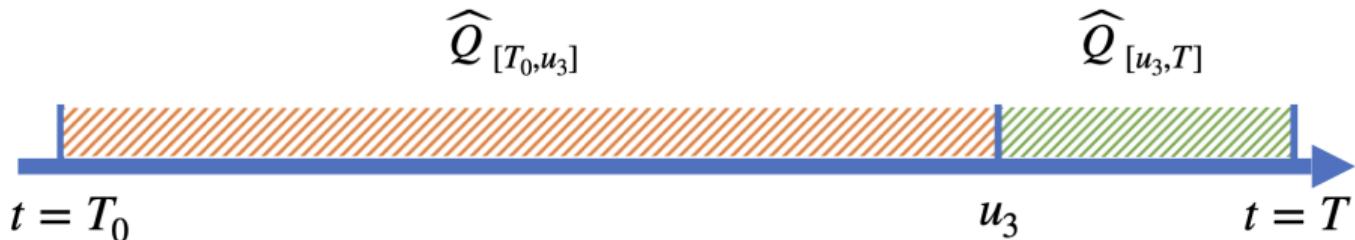
- Select a set of candidate change point locations $\boldsymbol{u} \in [T_0, T]$
- For each \boldsymbol{u} , estimate two Q-functions $\widehat{Q}_{[T_0, \boldsymbol{u}]}$ and $\widehat{Q}_{[\boldsymbol{u}, T]}$
- Construct the test based on their maximal difference



Method: Test Statistics (Cont'd)

- A CUSUM-type test statistic

- Select a set of candidate change point locations $\boldsymbol{u} \in [T_0, T]$
- For each \boldsymbol{u} , estimate two Q-functions $\widehat{Q}_{[T_0, \boldsymbol{u}]}$ and $\widehat{Q}_{[\boldsymbol{u}, T]}$
- Construct the test based on their maximal difference



Method: Test Statistics (Cont'd)

- Standard CUSUM-statistics that focuses on the difference in the **mean**
- We focus on the difference in \hat{Q} which is a **function** of the state-action pair
- Need to aggregate the maximal difference

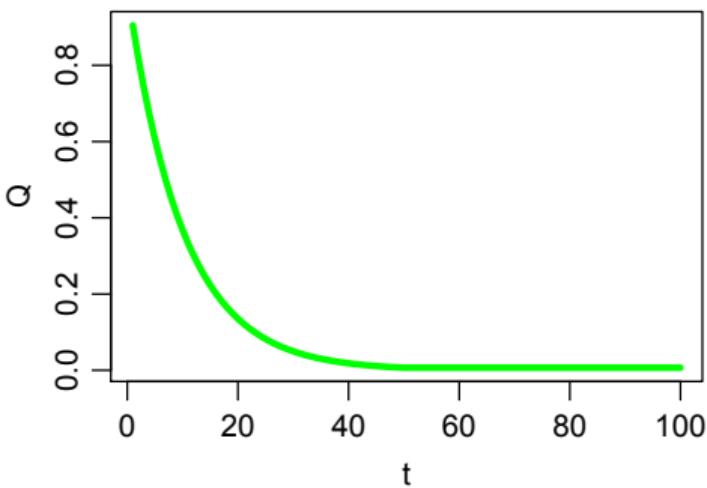
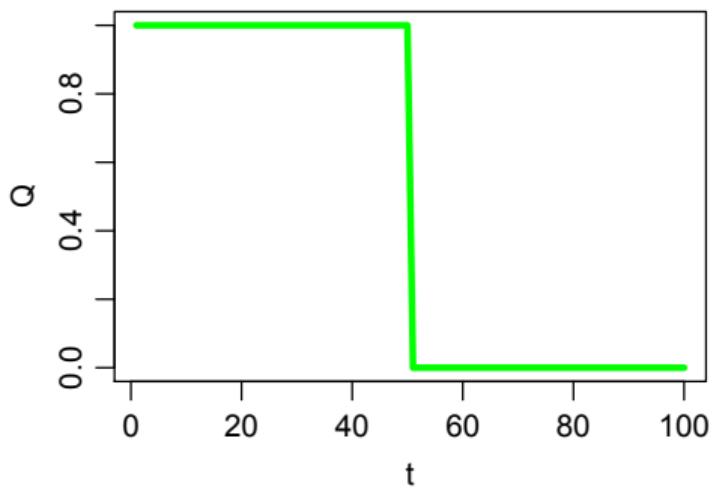
$$\Delta(\mathbf{a}, \mathbf{s}) = \max_{\mathbf{u}} \sqrt{\frac{(\mathbf{T} - \mathbf{u})(\mathbf{u} - \mathbf{T}_0)}{(\mathbf{T} - \mathbf{T}_0)}} |\hat{Q}_{[\mathbf{T}_0, \mathbf{u}]}(\mathbf{a}, \mathbf{s}) - \hat{Q}_{[\mathbf{u}, \mathbf{T}]}(\mathbf{a}, \mathbf{s})| \quad (1)$$

over different state-action pair

- Three proposed test statistics
 1. **ℓ_1 -type**: aggregate $\Delta(\mathbf{a}, \mathbf{s})$ over the empirical data distribution
 2. **maximum-type**: $\max_{\mathbf{a}, \mathbf{s}} \Delta(\mathbf{a}, \mathbf{s})$
 3. **normalized maximum** (widely used in econ): $\max_{\mathbf{a}, \mathbf{s}} \hat{\sigma}^{-1}(\mathbf{a}, \mathbf{s}) \Delta(\mathbf{a}, \mathbf{s})$
- Bootstrapped statistic: replace \hat{Q} in (1) with simulated normal errors

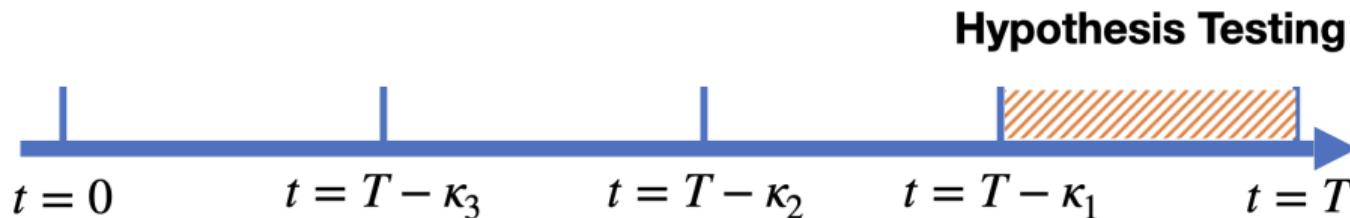
Method: Test Statistics (Cont'd)

The test is able to detect both **abrupt** and **smooth** changepoints



Method: Sequential Procedure

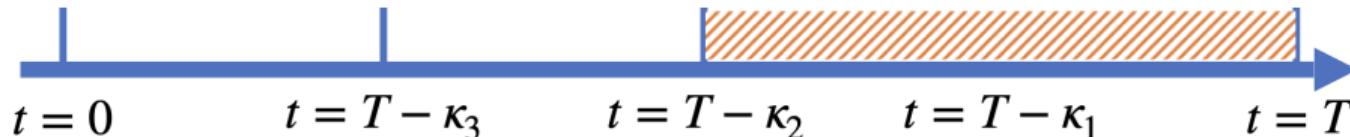
- Sequentially apply the test for selecting the most recent **best data “segment”**
 - Sequentially test whether \mathcal{H}_0 holds on the data interval $[T - \kappa, T]$ for $\kappa_1 < \kappa_2 < \kappa_3 < \dots$
 - Suppose \mathcal{H}_0 is first rejected at some $\kappa = \kappa_{j_0}$
 - Use the data subset within the interval $[T - \kappa_{j_0-1}, T]$ for policy optimisation



Method: Sequential Procedure (Cont'd)

- Sequentially apply the test for selecting the most recent **best data** “segment”
 - Sequentially test whether \mathcal{H}_0 holds on the data interval $[T - \kappa, T]$ for $\kappa_1 < \kappa_2 < \kappa_3 < \dots$
 - Suppose \mathcal{H}_0 is first rejected at some $\kappa = \kappa_{j_0}$
 - Use the data subset within the interval $[T - \kappa_{j_0-1}, T]$ for policy optimisation

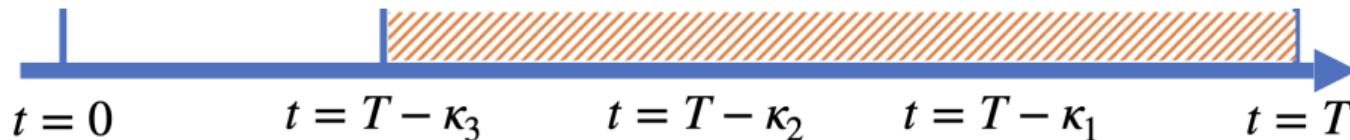
Not rejected. Combine more data



Method: Sequential Procedure (Cont'd)

- Sequentially apply the test for selecting the most recent **best data** “segment”
 - Sequentially test whether \mathcal{H}_0 holds on the data interval $[T - \kappa, T]$ for $\kappa_1 < \kappa_2 < \kappa_3 < \dots$
 - Suppose \mathcal{H}_0 is first rejected at some $\kappa = \kappa_{j_0}$
 - Use the data subset within the interval $[T - \kappa_{j_0-1}, T]$ for policy optimisation

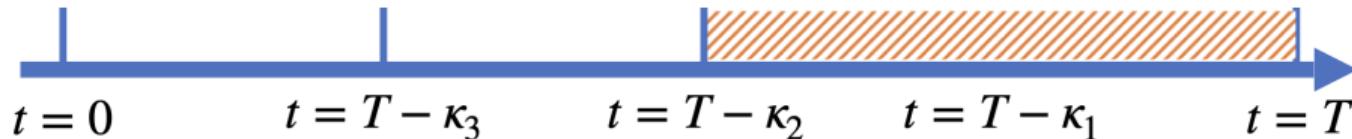
Not rejected. Combine more data



Method: Sequential Procedure (Cont'd)

- Sequentially apply the test for selecting the most recent **best data** “segment”
 - Sequentially test whether \mathcal{H}_0 holds on the data interval $[T - \kappa, T]$ for $\kappa_1 < \kappa_2 < \kappa_3 < \dots$
 - Suppose \mathcal{H}_0 is first rejected at some $\kappa = \kappa_{j_0}$
 - Use the data subset within the interval $[T - \kappa_{j_0-1}, T]$ for policy optimisation

Rejected. Use the last data interval



Simulation

- Settings:

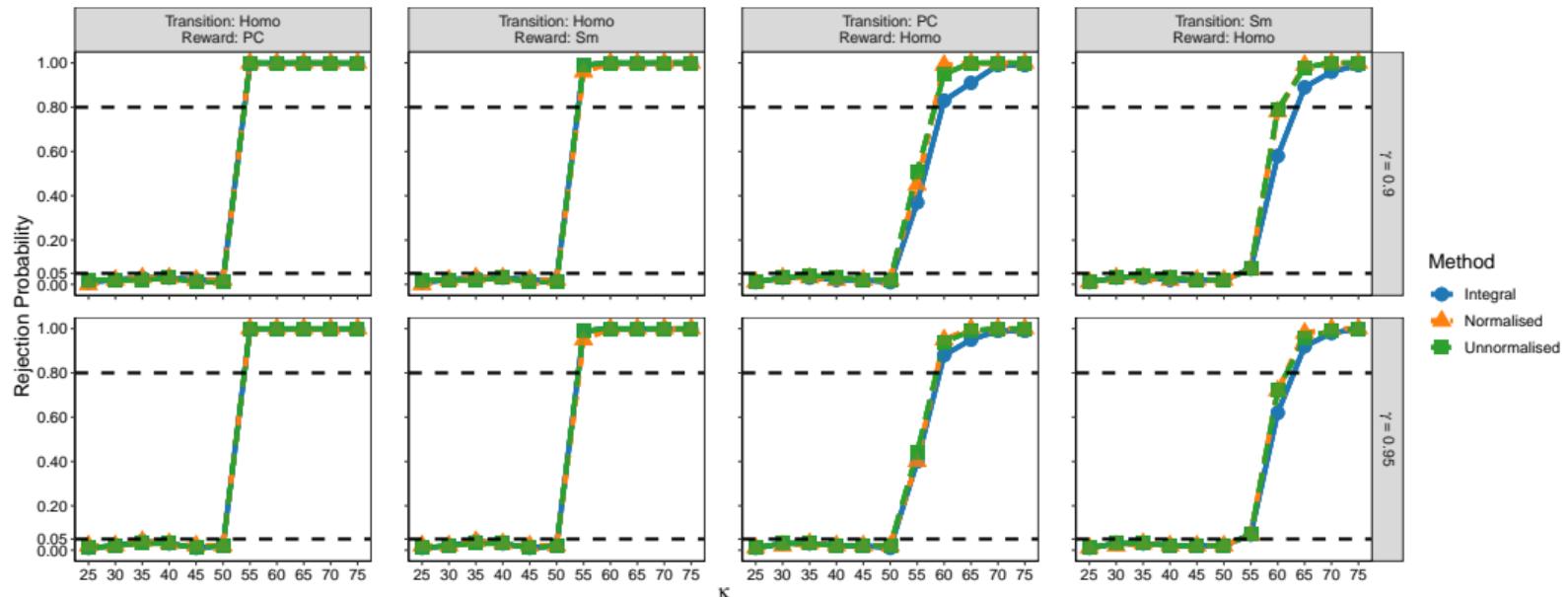
	State transition function	Reward function
(1)	Time-homogeneous	Piecewise constant
(2)	Time-homogeneous	Smooth
(3)	Piecewise constant	Time-homogeneous
(4)	Smooth	Time-homogeneous

- Analysis:

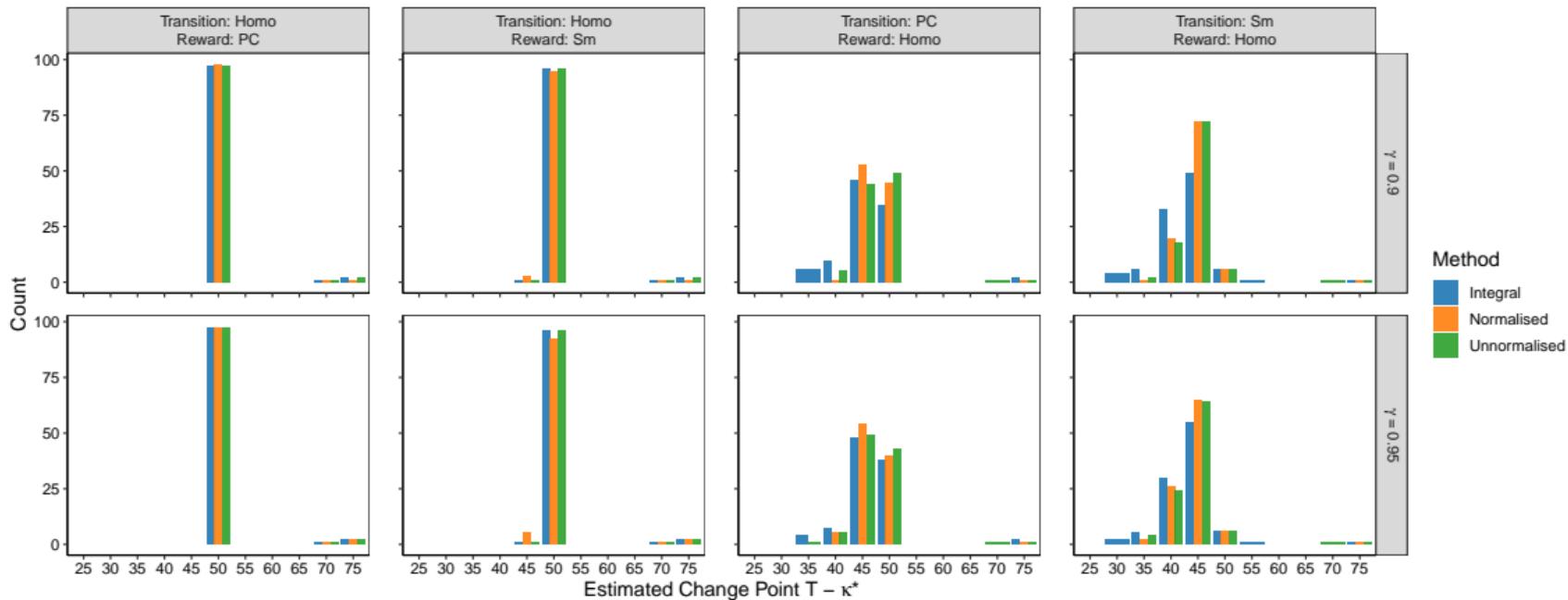
- Testing stationarity
- Change point detection
- Policy learning

Analysis I: Testing Stationarity

- $N = 25$, $T = 100$, true change occurs at $\kappa = 50$



Analysis II: Change Point Detection

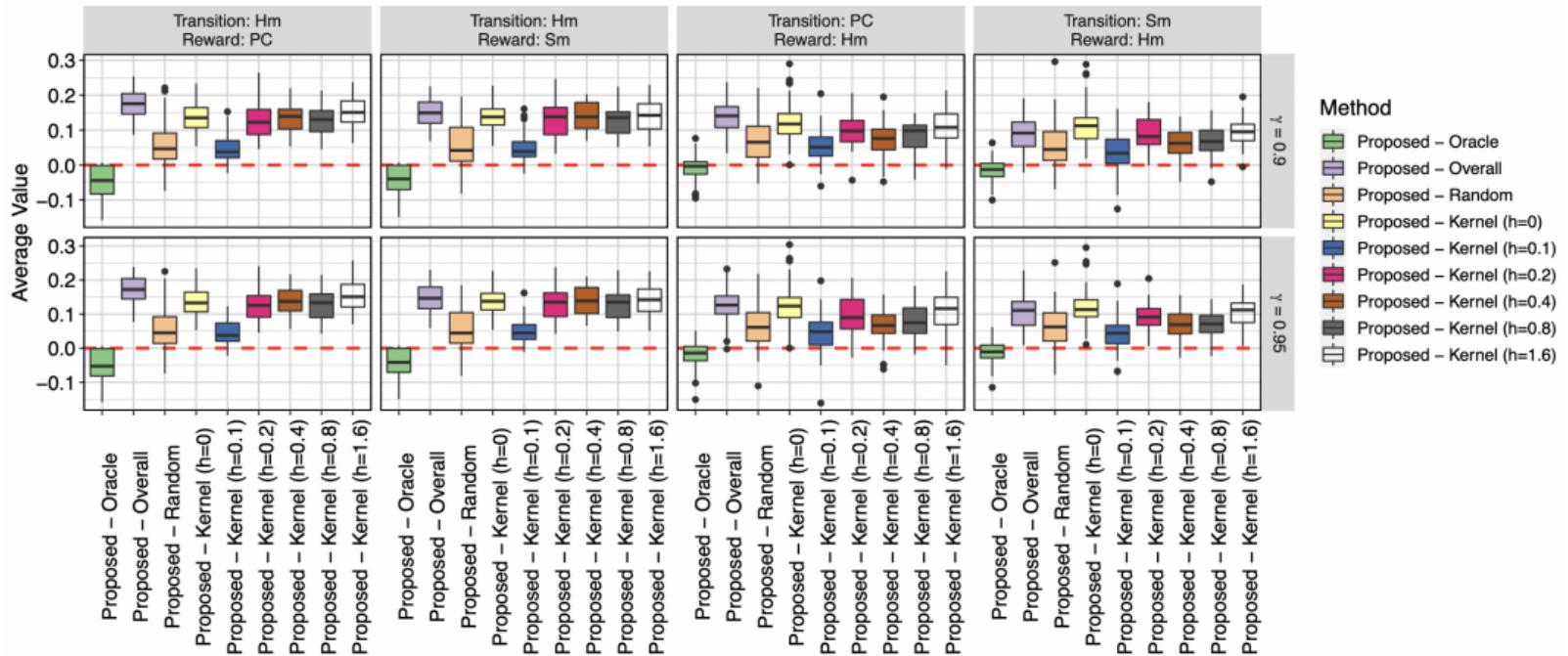


Analysis III: Policy Learning

- **Offline** data with $T = 100$
- Apply our proposal for identifying the **recent change point** \hat{T}
- Apply RL algorithm to data interval $[\hat{T}, T]$ to learn a **warm-up** policy
- Use the warm-up policy (combined with ϵ -greedy) to generate **online** data
- Online data come in **batches** regularly at every 25 time points
- Number of change points follow a **Poisson process** with rate $1/50$
- Update the change point and the policy after each data batch arrives
- Compute the **average reward**

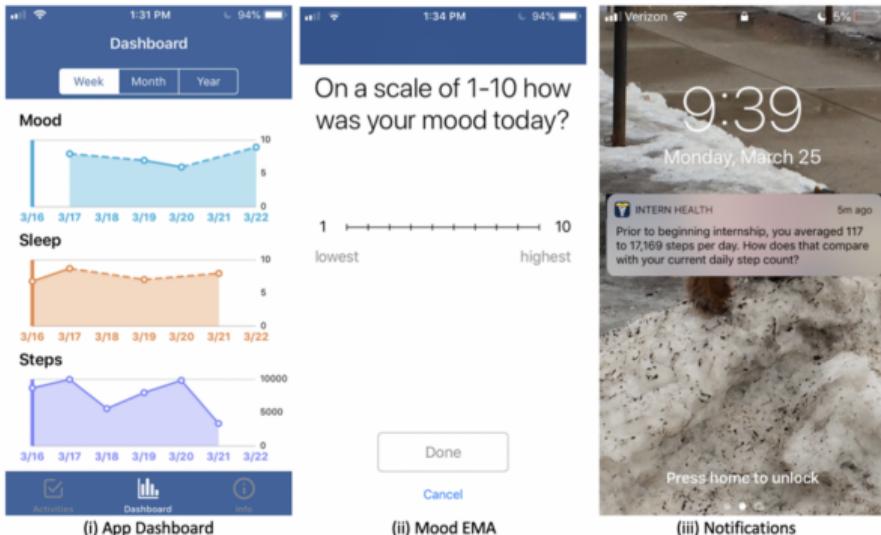
Analysis III: Policy Learning (Cont'd)

- Competing methods: Oracle, Overall, Random, Kernel

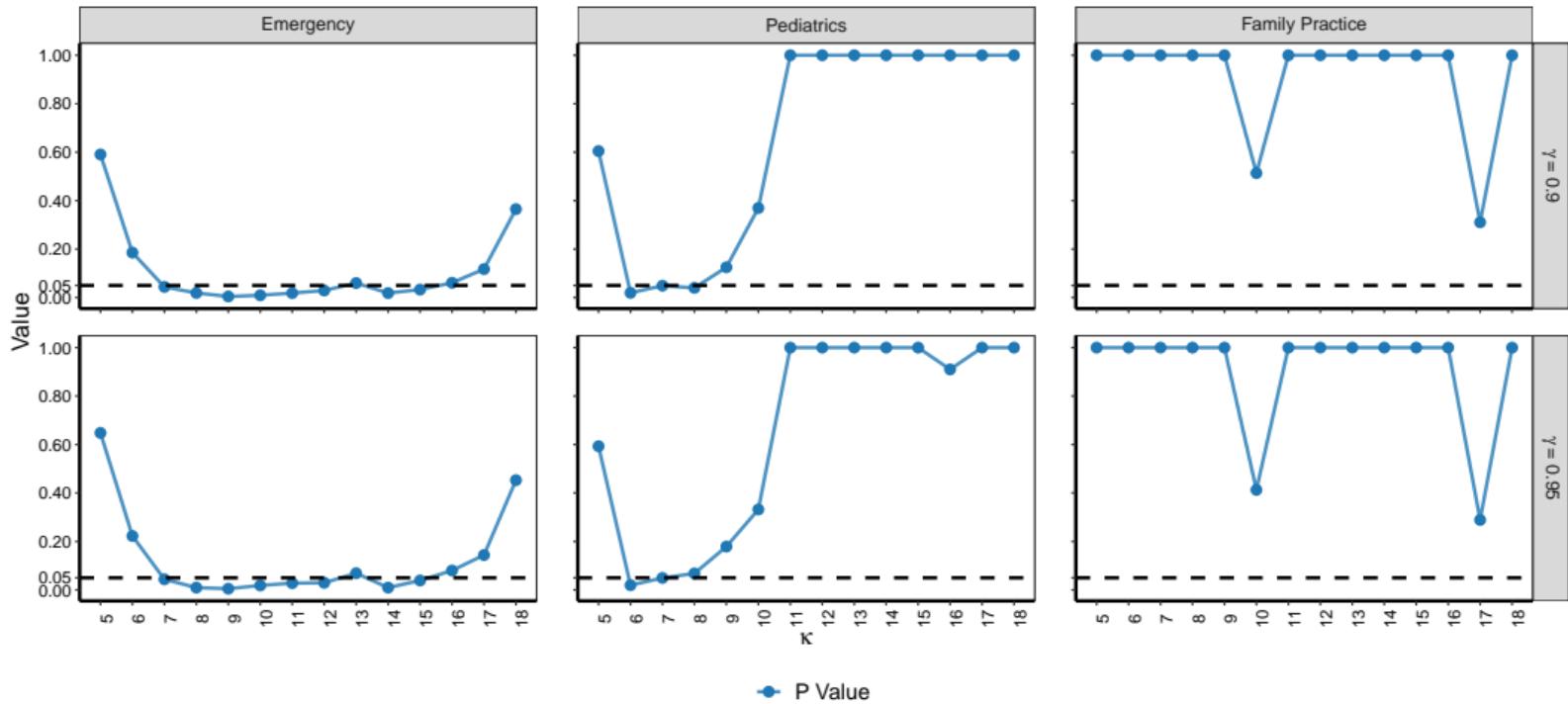


Application: Intern Health Study

- **Subject:** First-year medical interns
- **Objective:** Develop treatment policy to determine whether to send certain text messages to interns to improve their health
- **S_t :** Interns' mood scores, sleep hours and step counts
- **A_t :** Send text notifications or not
- **R_t :** Step counts



Application: Intern Health Study (Cont'd)



Application: Intern Health Study (Cont'd)

Number of Change Points	Specialty	Method	$\gamma = 0.9$	$\gamma = 0.95$
≥ 1	Emergency	Proposed	8237.16	8295.99
		Overall	8108.13	8127.55
		Behavior	7823.75	7777.32
		Random	8114.78	8080.27
≥ 2	Pediatrics	Proposed	7883.08	7848.57
		Overall	7925.44	7960.12
		Behavior	7730.98	7721.29
		Random	7807.52	7815.30
0	Family Practice	Proposed	8062.50	7983.69
		Overall	8062.50	7983.69
		Behavior	7967.67	7957.24
		Random	7983.52	7969.31

TABLE 3

Mean value estimates using decision tree in analysis of IHS. Values are normalised by multiplying $1 - \gamma$. All values are evaluated over 10 splits of data.

- Mean value is the weekly average step counts per day
- The proposed method improves mean value by 50 – 150 steps, compared to the behavior policy

Bidirectional Theory

- N the number of trajectories
- T the number of decision points per trajectory
- **bidirectional asymptotics:** a framework allows either N or $T \rightarrow \infty$
- large N , small T (Intern Health Study)



- small N , large T (OhioT1DM dataset)



- large N , large T (games)

Bidirectional Theory (Cont'd)

Theorem (Informal Statement)

Under certain conditions, as either \mathbf{N} or \mathbf{T} diverges to infinity

1. Our test controls the type-I error under \mathcal{H}_0
2. Its power approaches 1 under \mathcal{H}_1

- The number of sieves shall grow to infinity → reduce the model misspecification error (classical weak convergence theorem is **not** directly applicable)
- Develop a **matrix concentration inequality** under nonstationarity (sharper than naively applying concentration inequalities for scalar random variables)
- **Undersmoothing** is not needed to guarantee the test has good **size** property
- **Cross-validation** can be employed to select the number of sieves
- ℓ_1 and normalized maximum type tests require **weaker** conditions than the maximum-type test

Project II

Doubly Inhomogeneous Reinforcement Learning

Joint work with Liyuan Hu, Mengbing Li, Zhenke Wu and Piotr Fryzlewicz

Motivation

- Most existing RL algorithms require two fundamental assumptions:
 1. **Global (temporal) stationarity assumption** (GSA): the system dynamics within each data trajectory does not experience temporal changes
 2. **Global (subject) homogeneity assumption** (GHA): all data trajectories share the same system dynamics
- Both assumptions are likely to be violated in a number of applications (e.g., healthcare, ridesharing), challenging high-quality sequential decision making

Table: Forms of the Optimal Policy in Different Environments.

GSA ✓ GHA ✓	GSA ✓ GHA ✗	GSA ✗ GHA ✓	GSA ✗ GHA ✗
doubly homogeneous	stationary	homogeneous	subject-specific history-dependent

- In this project, we study RL in **doubly inhomogeneous** environments (e.g., dynamics change over time and population)

Configurations of Double Inhomogeneity

- To illustrate double inhomogeneity, consider two subjects with a single change point

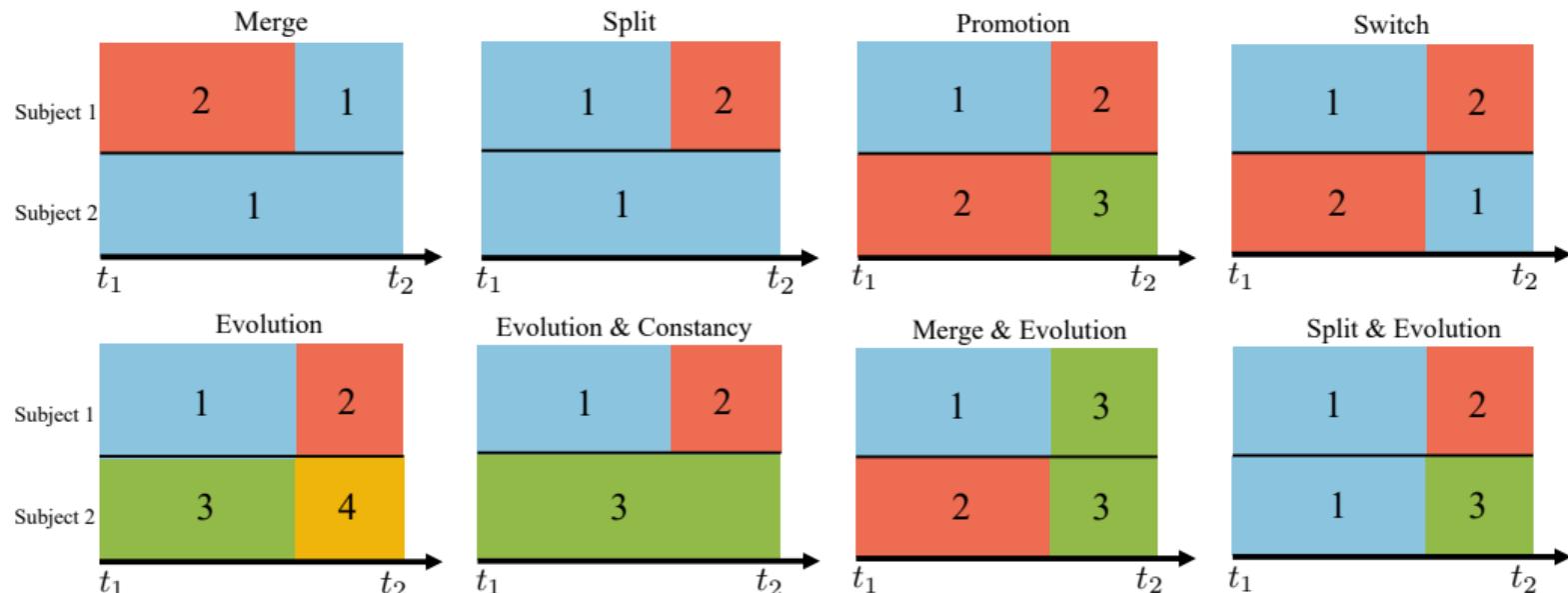


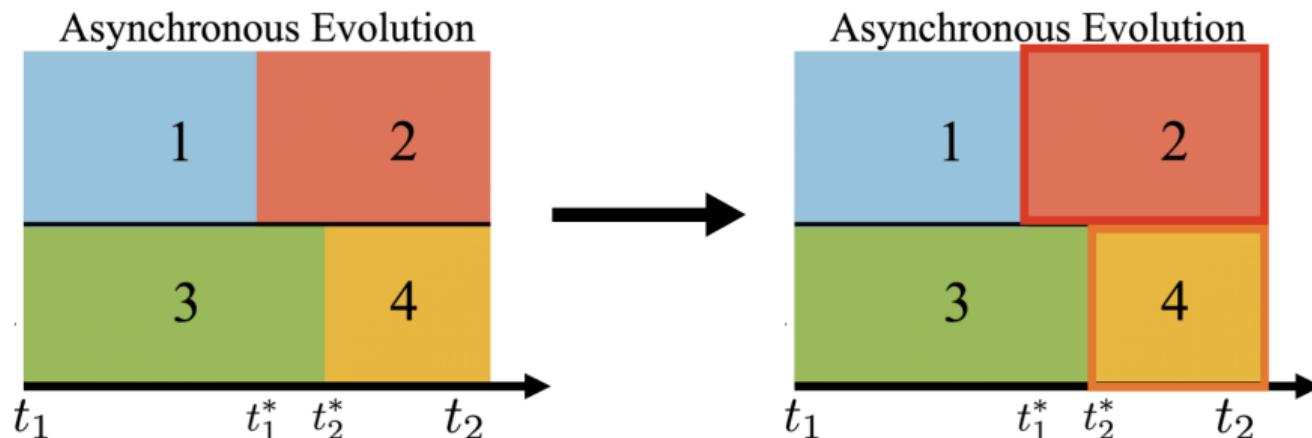
Figure: Basic building blocks with two subjects (one in each row) and a single change point. Different dynamics are represented by distinct colors.

Data, Assumptions and Objective

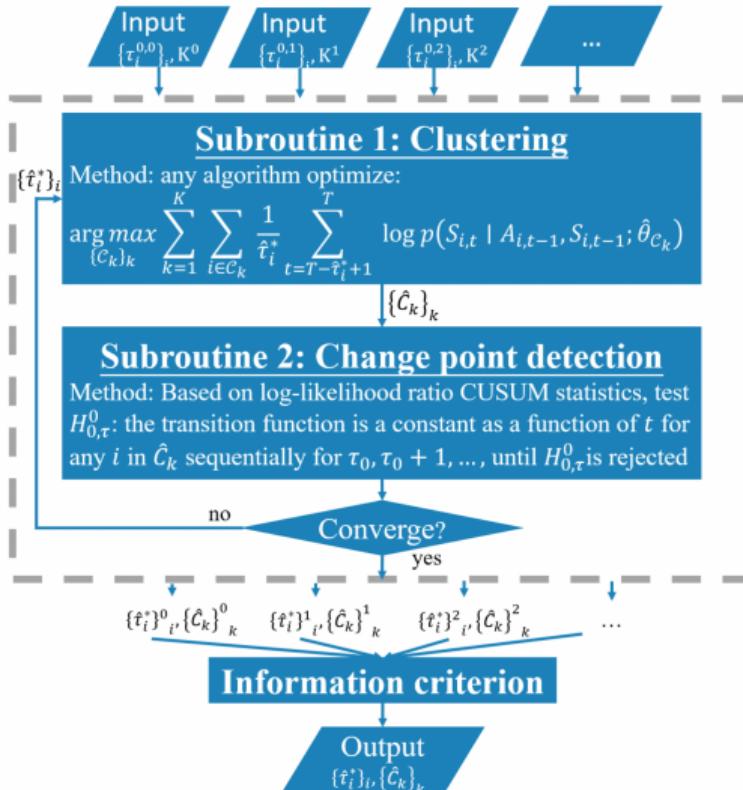
- Data: N trajectories, T time points per trajectory.
- Our assumptions:
 1. **Local Stationarity at the Endpoint (LSE)**: For each subject i , there exists some $\tau_i > 0$ such that the transition function is a constant function of t for any $T - \tau_i \leq t \leq T$.
 2. **Local Homogeneity at the Endpoint (LHE)**: There exists a finite number K of disjoint subject clusters $\cup_{k=1}^K \mathcal{C}_k$, where $\mathcal{C}_k \subseteq \{1, \dots, N\}$, such that within each cluster \mathcal{C}_k , the transition function at time T is constant over different subjects
- Objective: determine the **best data rectangle** that display similar dynamics over time and subjects, to borrow information for effectively policy learning

Objective (Cont'd)

As an example ...



Method



Theory

Table 2: Rate of convergence when N and T have different divergence properties. The “CP error” refers to the change point detection error and “non-negligible” means that the error does not decay to zero as $N \rightarrow \infty$.

Iteration		$T \rightarrow \infty$	$T \rightarrow \infty$	T fixed
		$N \rightarrow \infty$	N fixed	$N \rightarrow \infty$
1 st	clustering error	0	0	non-negligible
	CP error	0	$O_p\left(\frac{\log^2(NT)}{NTs_{cp}^2}\right)$	non-negligible
2 nd	clustering error	0	0	non-negligible
	CP error	0	$O_p\left(\frac{\log^2(NT)}{NTs_{cp}^2}\right)$	non-negligible
...	

- Only require the **overestimation** error of each initial τ_i to satisfy certain rate. No assumption is imposed on their **underestimation** error.
- Detect **weaker signals** and have **faster convergence rates** compared to applying the clustering algorithm per time or the CP detection algorithm per subject

Simulation

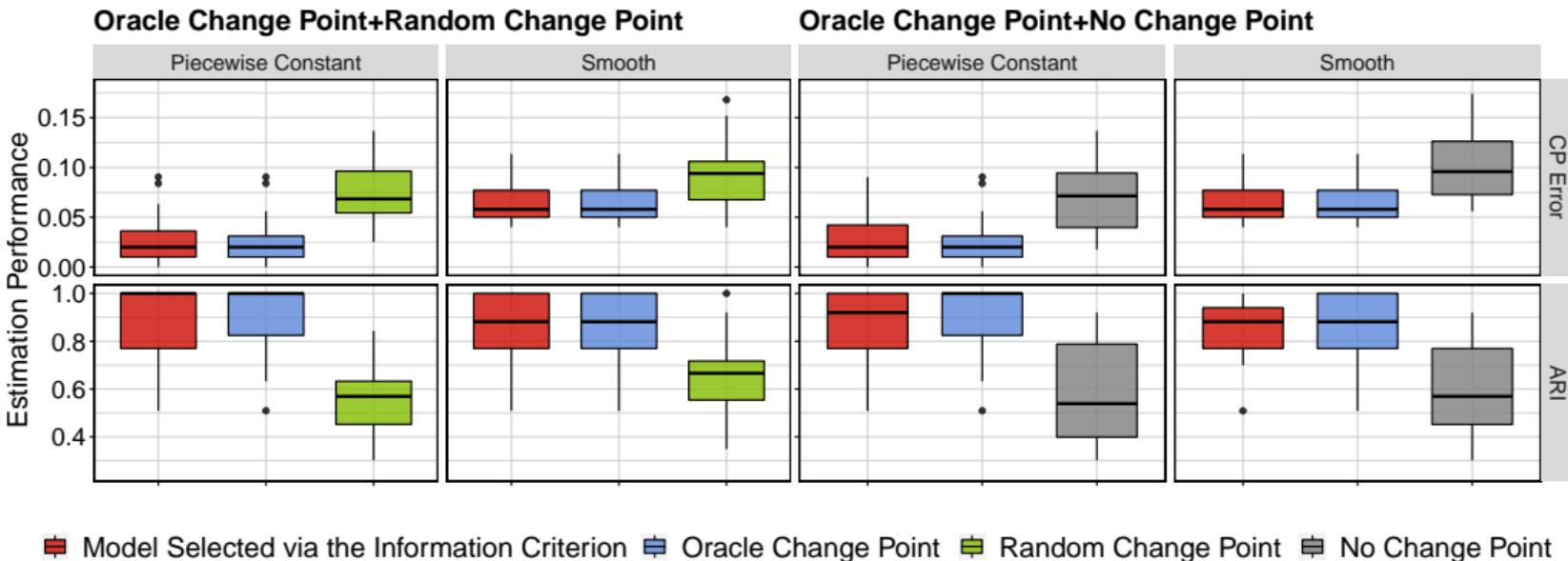


Figure: Average CP error and ARI with different initial change point locations are chosen by the information criterion.

Simulation (Cont'd)

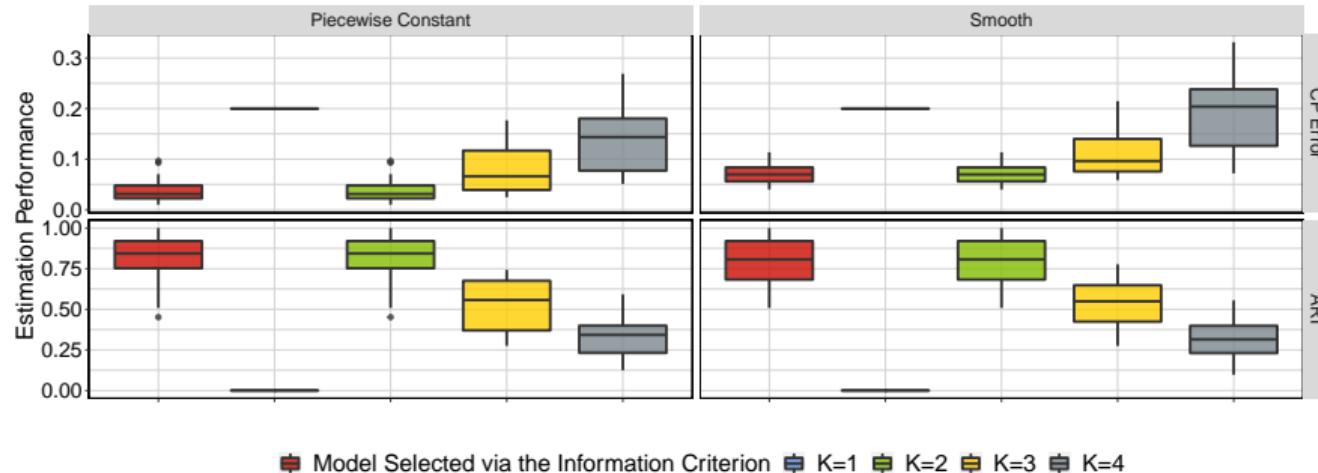


Figure: Average performance in offline estimation with different number of clusters ($K = 1, 2, 3, 4$) and the results chosen by the information criterion.

Simulation (Cont'd)

- **Online value evaluation:** recursively apply the proposed algorithm to update the estimated optimal policy and use this policy for action generation
- **Competing policies:** oracle, doubly homogeneous (DH), homogeneous, stationary

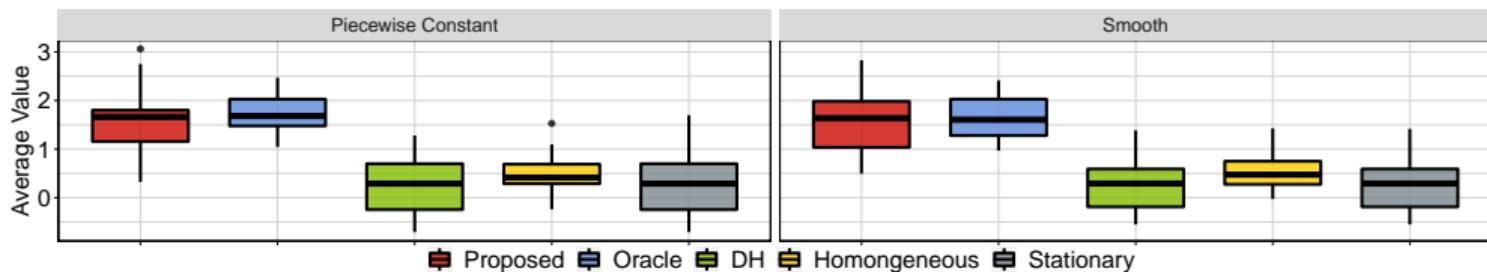


Figure: Boxplot of the expected returns under the proposed policy and other baseline policies that either ignore non-stationarity or heterogeneity.

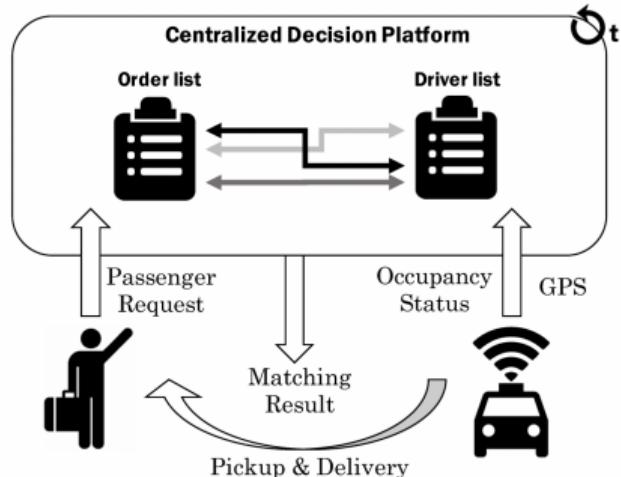
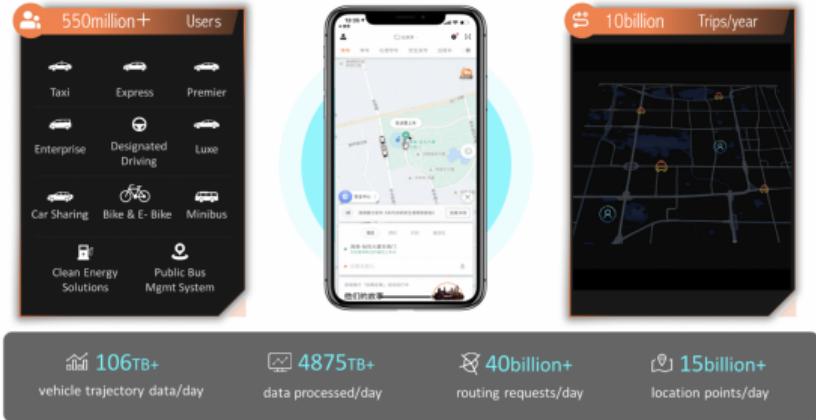
Project III

Pattern Transfer Learning for Reinforcement Learning in Order Dispatching

—Best Paper in IJCAI RL4ITS Workshop

Joint work with Runzhe Wan, Sheng Zhang, Shikai Luo and Rui Song

Ridesharing: Order-Dispatching



Objective: learn an optimal policy to maximize

- answer rate (proportions of call orders being answered)
- completion rate (proportions of call orders being completed)
- drivers' income

Closest Driver Policy

Assign the call order to the closest available driver

$$\arg \min_{\mathbf{a}_{i,j}} \sum_{i=1}^m \sum_{j=1}^n d(i,j) a_{i,j} \quad \text{Minimize driver-passenger total distance}$$

$$s.t. \sum_{i=1}^m a_{i,j} \leq 1, j = 1, \dots, n \quad \text{Order assigned to at most one driver}$$

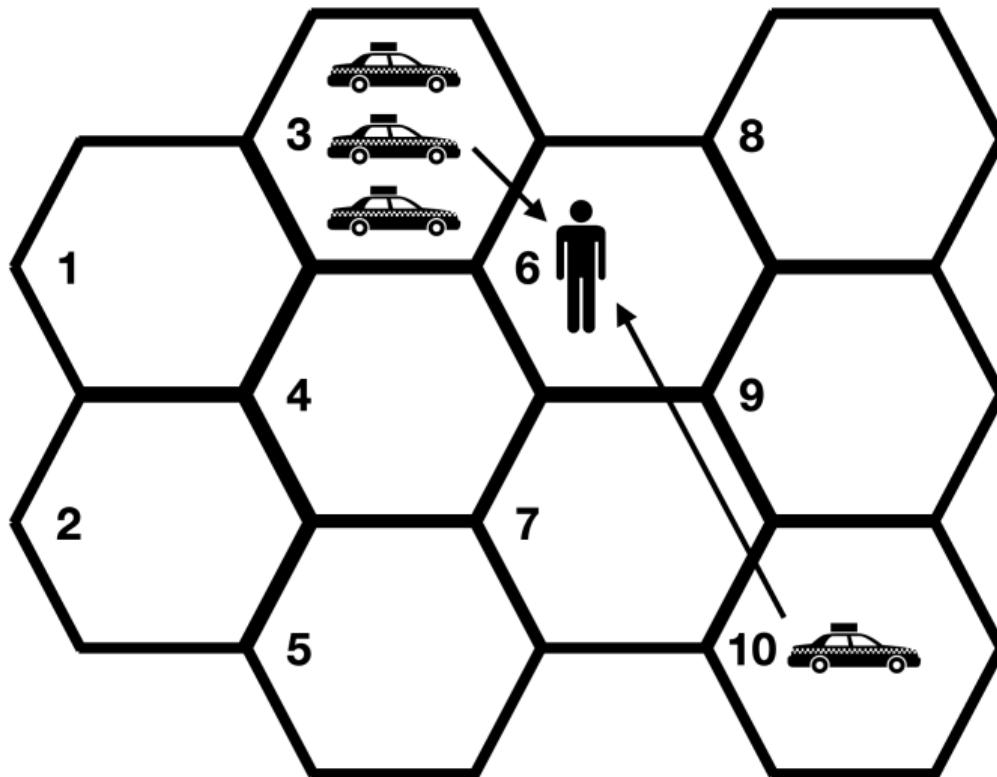
$$\sum_{j=1}^n a_{i,j} \leq 1, i = 1, \dots, m \quad \text{Driver assigned to at most one order}$$

- i indexes the i th driver
- $d(i,j)$ = distance between i and j
- One of the two equalities shall hold
- j indexes the j th order
- $a_{i,j} = 1 \Leftrightarrow$ order j is assigned to i

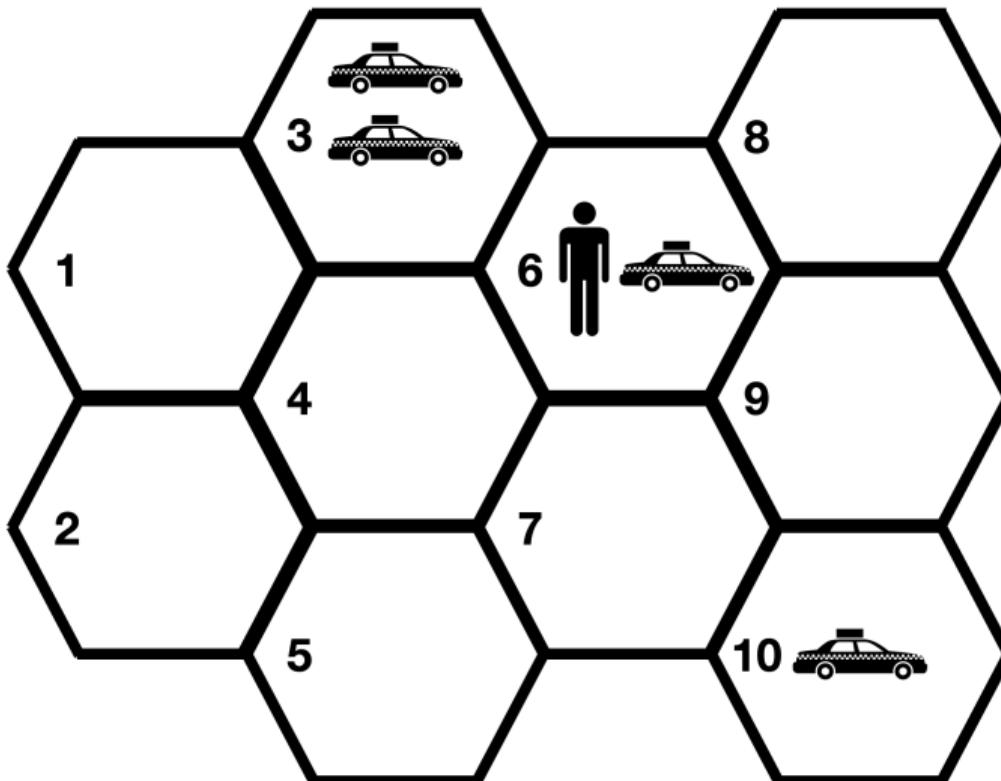
Closest Driver Policy: Limitations

- The company implements the policy every 2 seconds
- **Myopic** policy (e.g., maximize immediate rewards)
- No guarantee it will maximize long-term rewards
- Example given later

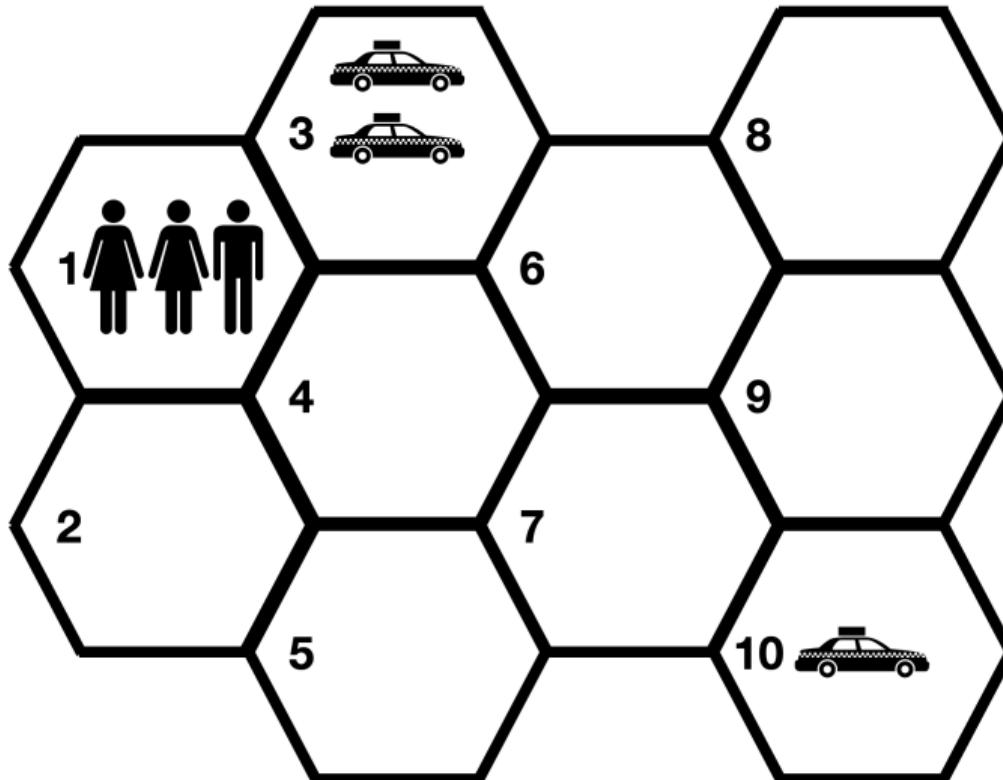
Illustration of Limitations of Closest Driver Policy



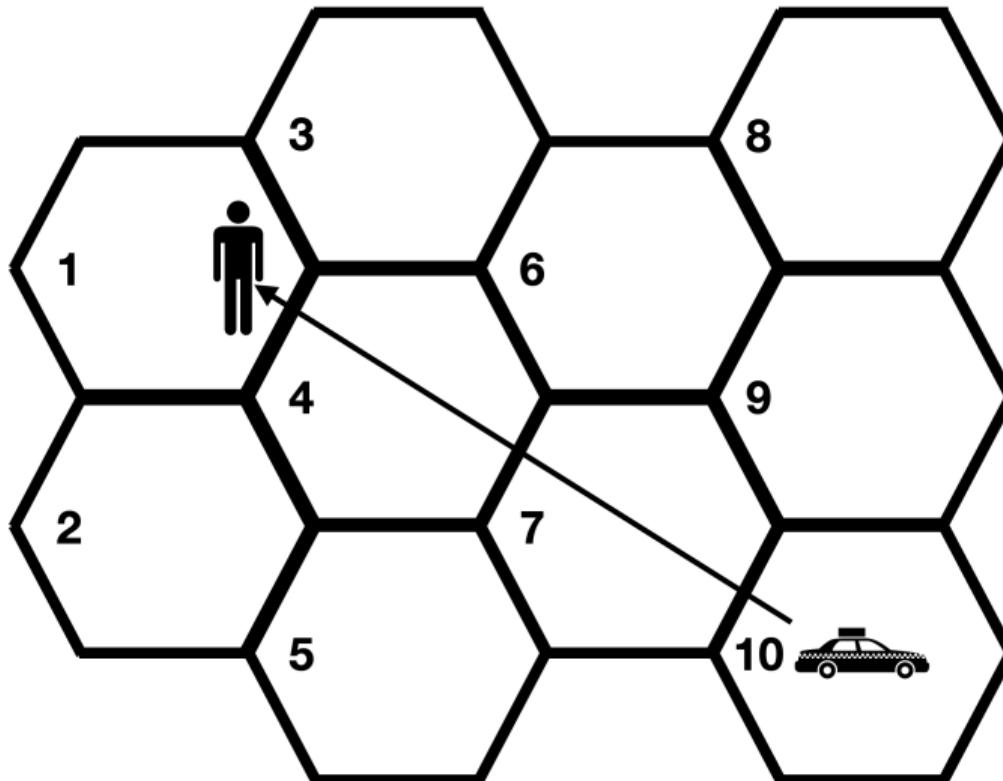
Adopting the Closest Driver Policy



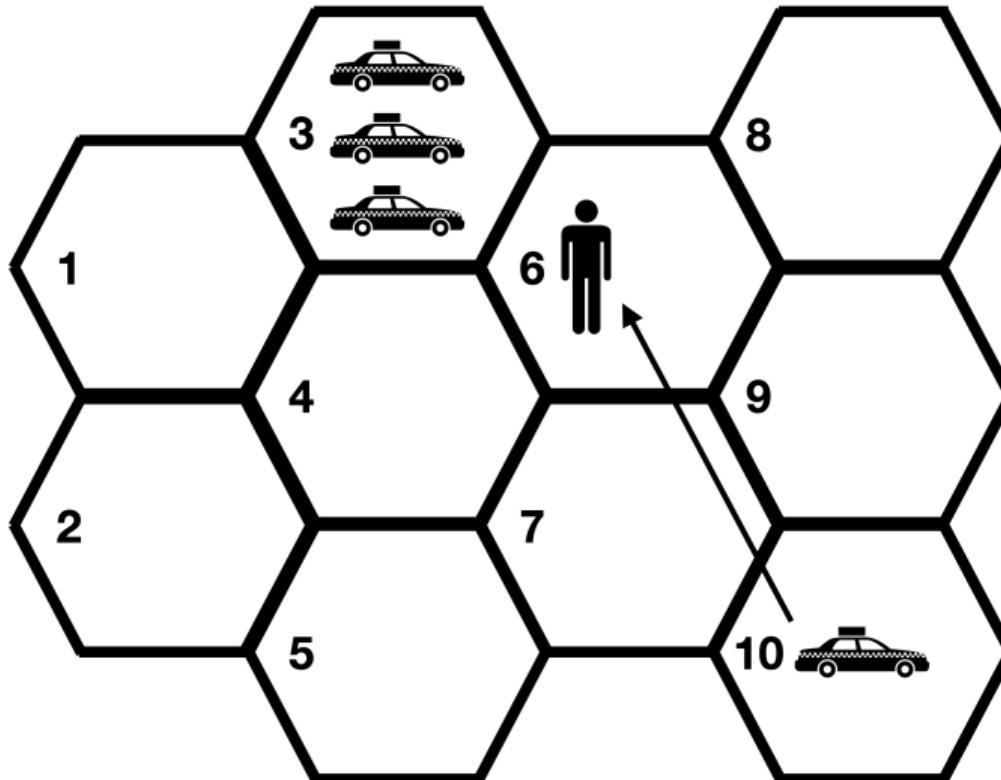
Some Time Later . . .



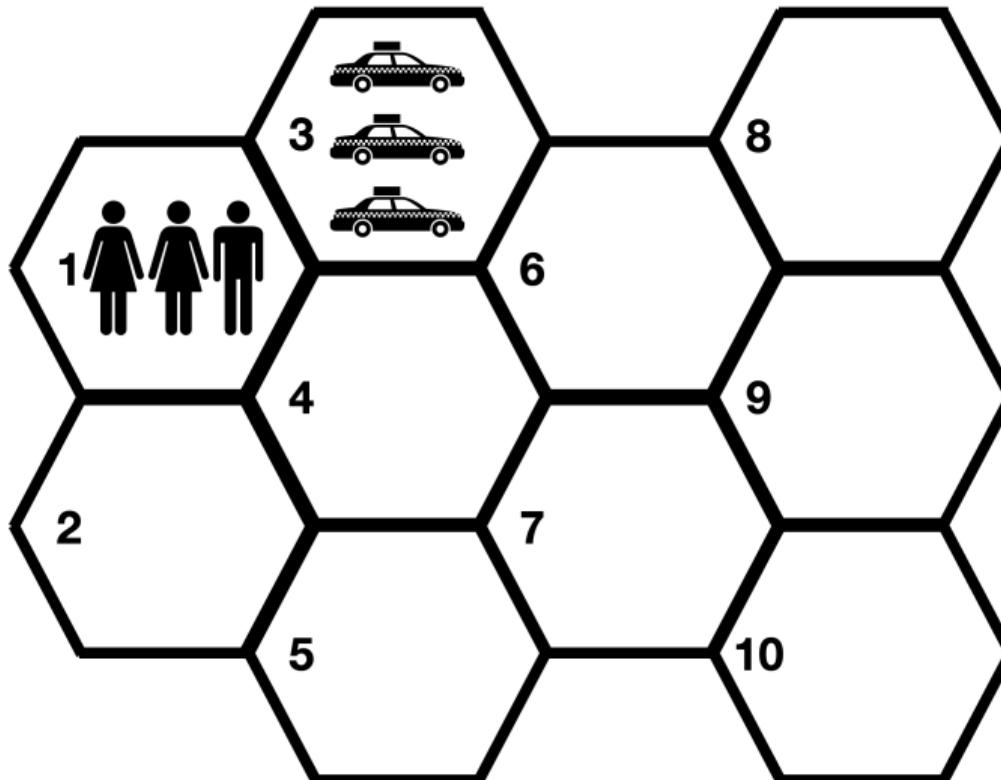
Miss One Order



Consider a Different Action



Able to Match All Orders



MDP Order Dispatch Policy (Xu et al., 2018)

- Adopts a **reinforcement learning** framework to optimize long-term rewards
- Delivers **remarkable improvement** on the platform's efficiency
- **Main idea:**
 - Closest driver is myopic because its objective function (e.g., total distance) only considers **immediate rewards**
 - Use an objective function that involves **long-term rewards** (e.g., value)
- A learning and planning approach:
 - **Learning:** policy evaluation based on historical data
 - **Planning:** order dispatch by maximizing total value

Pattern Transfer Learning

- **Motivation:** violation of **stationarity** assumption in data collected from ridesharing platforms, leading to nonstationary MDPs
 - The system dynamics is likely to vary over time
- **Naive solution:**
 - Use more **recent** data for policy evaluation (learning)
 - Use value function trick for order dispatching (planning)
 - **Disadvantage:** discard a lot of data
- **Research question:** how should we efficiently utilize historical dataset to improve the efficiency of value function estimation

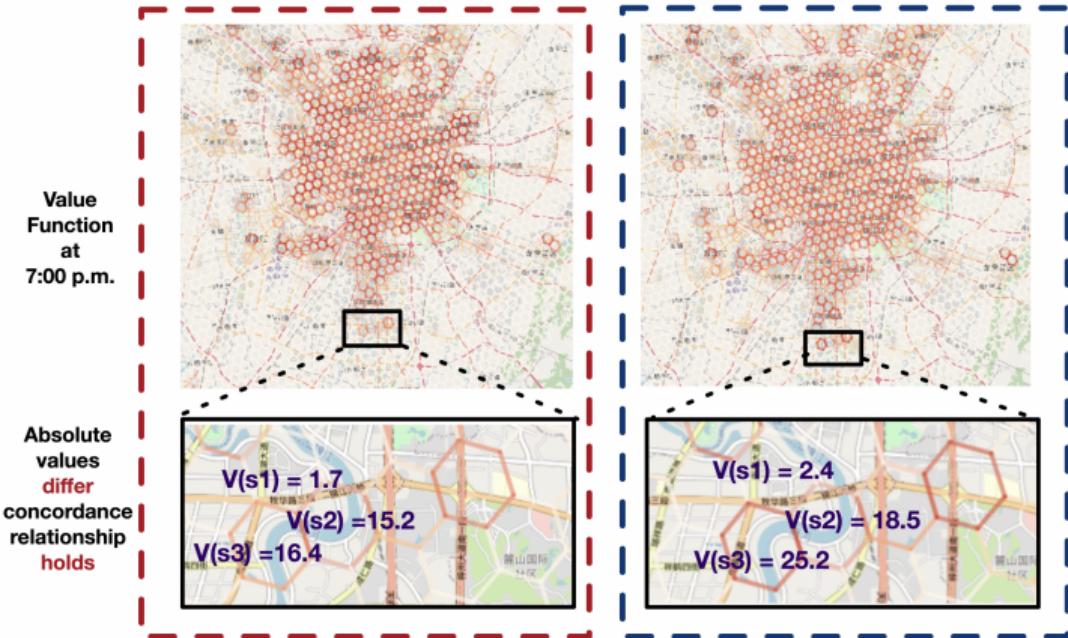
Nonstationarity

- Value function estimated based on data from [KDD CUP 2020](#)
- 30-day's data collected from Didi Chuxing
- Left plot: value based on first 15-day's data
- Right plot: value based on last 15-day's data
- Absolute values differ



Main Idea

- Magnitude of value is nonstationary
- **Concordance** relationship of value remains stationary
- Values of **hot** zones (e.g., centers) are consistently larger than those of **cold** zones (e.g., suburbs)
- Overall, concordance relationship holds on more than 80% state pairs



Concordance

- Widely used in the statistics and economics literature
 - Maximum rank correlation estimator for regression (Han, 1987)
 - Concordance-assisted estimator for learning optimal dynamic treatment regimes (Fan et al., 2017; Shi et al., 2021)
- For two states s_1 and s_2 and two value functions V_1 and V_2
 - Concordance is $\mathbf{1}$ if $\{V_1(s_1) - V_1(s_2)\}\{V_2(s_1) - V_2(s_2)\} \geq \mathbf{0}$ and $\mathbf{0}$ otherwise
- Concordance penalty:

$$c(V_1, V_2) = \frac{1}{n(n-1)} \sum_{i < j} \#[(\{V_1(s_i) - V_1(s_j)\}\{V_2(s_i) - V_2(s_j)\}) < 0]$$

- Constrained policy evaluation: compute the value function subject to the concordance constraint,

$$c(V^{old}, V^{new}) \leq \epsilon.$$

Simulation

- Build dispatch simulator using the KDD dataset

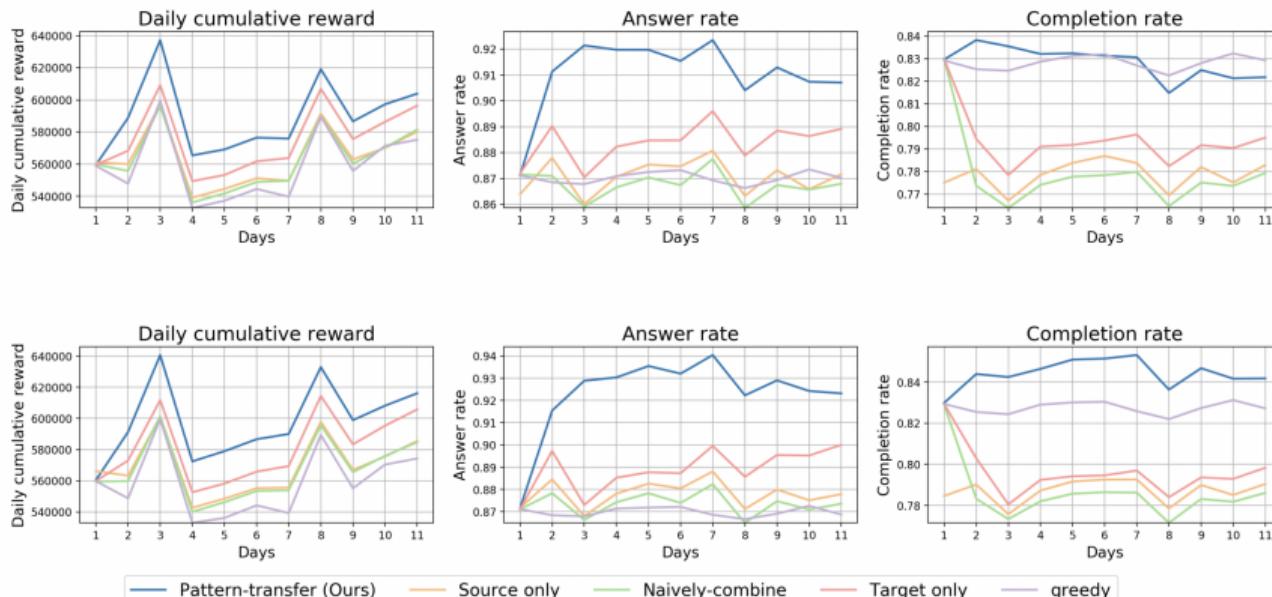


Figure 2: Performance of different methods when $\gamma = 0.9$ (upper) and $\gamma = 0.95$ (lower). The x-axis represents consecutive weekdays in the target environment. Our method outperforms the baseline methods under different metrics.

Thank You!

😊 Papers and softwares can be found on my personal website

callmespring.github.io