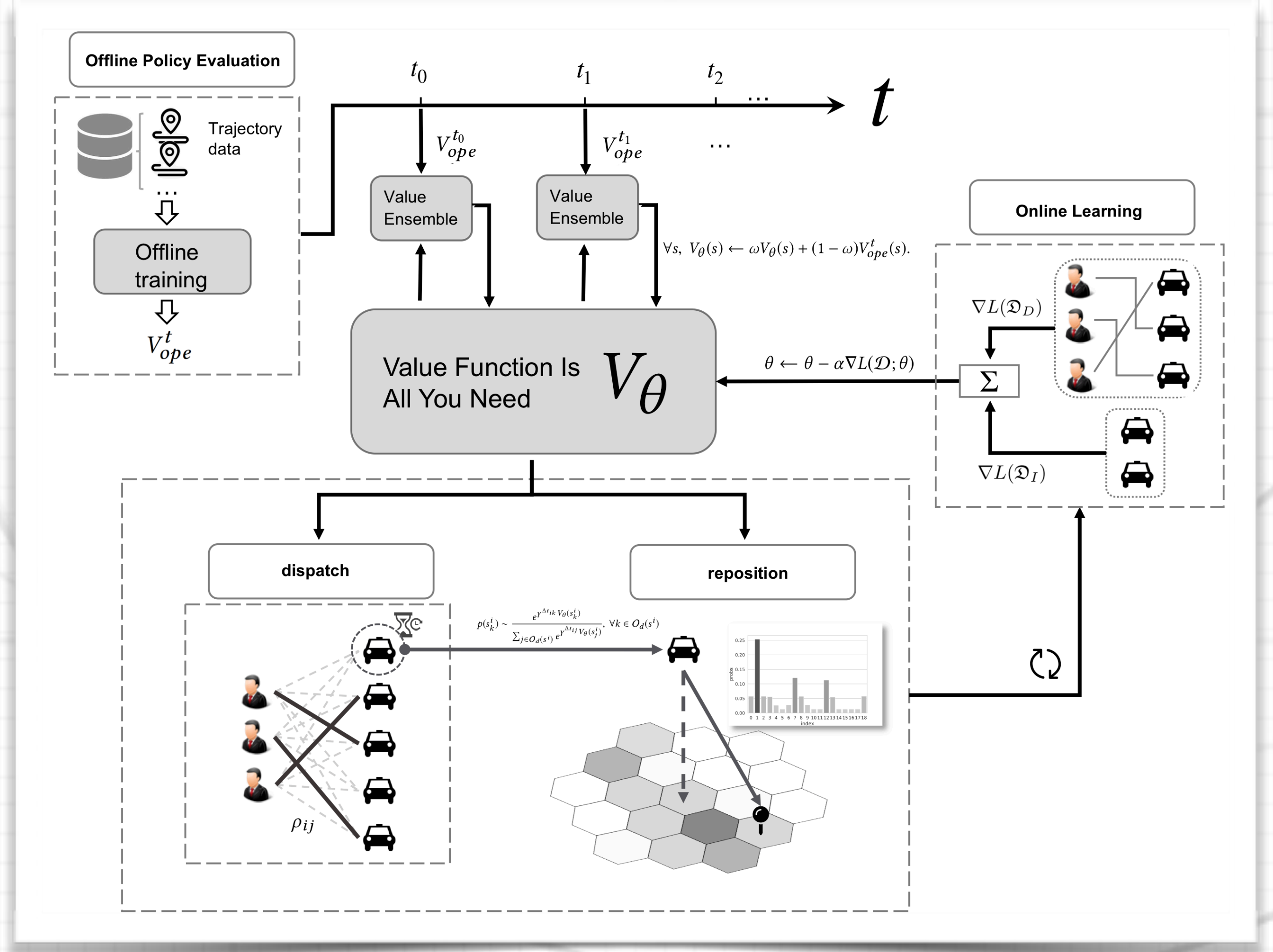
Value Function is All You Need: A Unified Learning Framework for Ride Hailing Platforms

Xiaocheng Tang

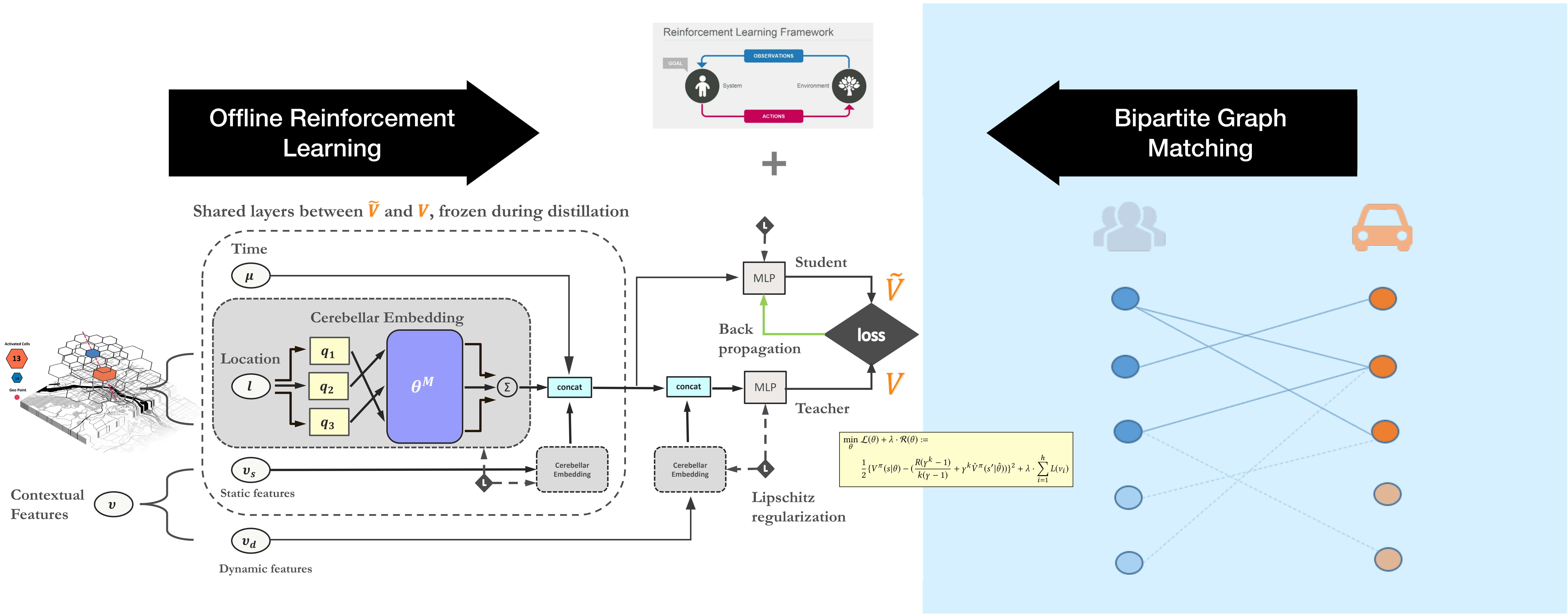
DiDi Labs, Mountain View, CA | http://mktal.github.io

Joint work with Fan Zhang, Zhiwei Qin, Yansheng Wang, Dingyuan Shi, Bingchen Song, Yongxin Tong, Hongtu Zhu, Jieping Ye



KDD '21, Aug. 14–18, 2021, Singapore, Singapore

Background



Objective | *X. Tang et al.,* ***KDD Oral*** *2019*

$$\max_{x \in C} \sum_{i=1}^{m} \sum_{j=1}^{n} \rho_{ij} x_{ij}$$

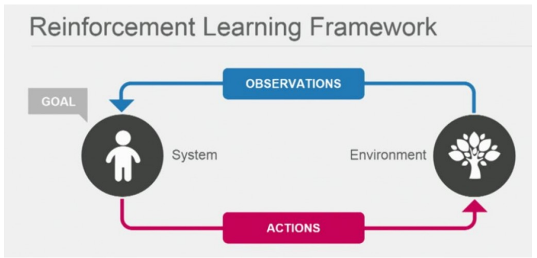
✓ maximize the total utilities of the assignments where the utility scores are computed as the Temporal Difference error between order's destination state and driver 's current state, e.g.,

Spatiotemporal optimality!

$$\rho_{ij} = R_{ij} \frac{(\gamma^{k_{ij}} - 1)}{k_{ij}(\gamma - 1)} + \gamma^{k_{ij}} V(s_j) - V(s_i) + \Omega \cdot U_{ij}$$

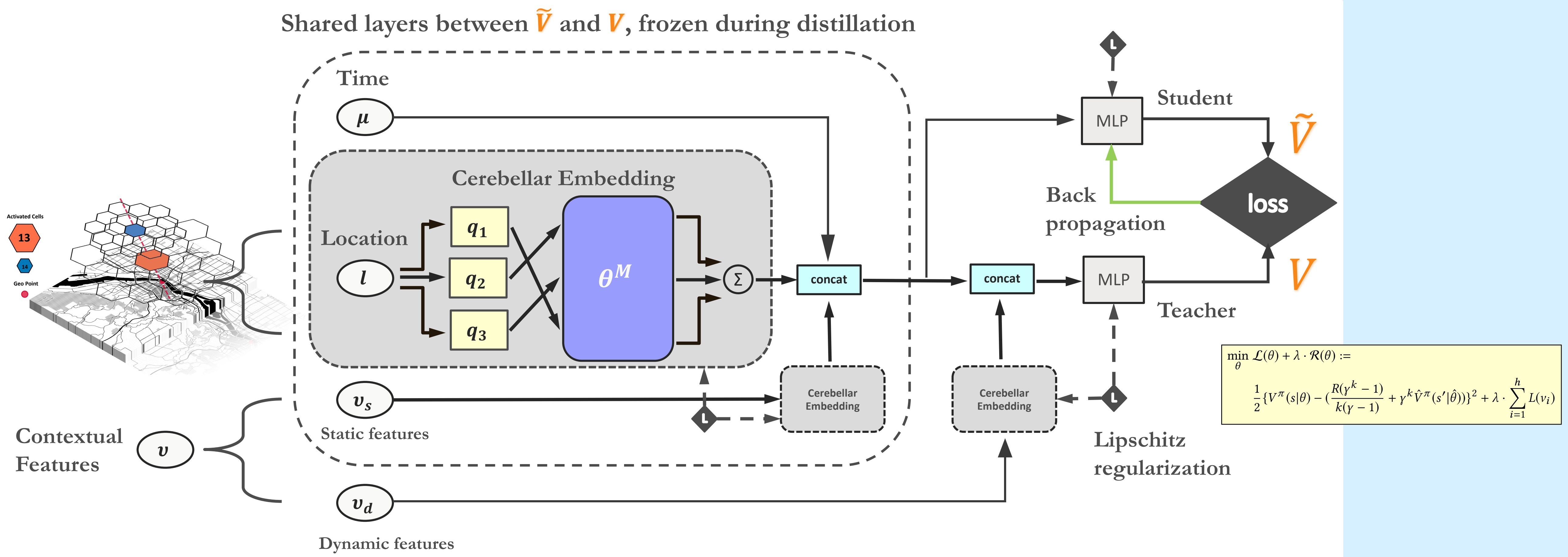
Background

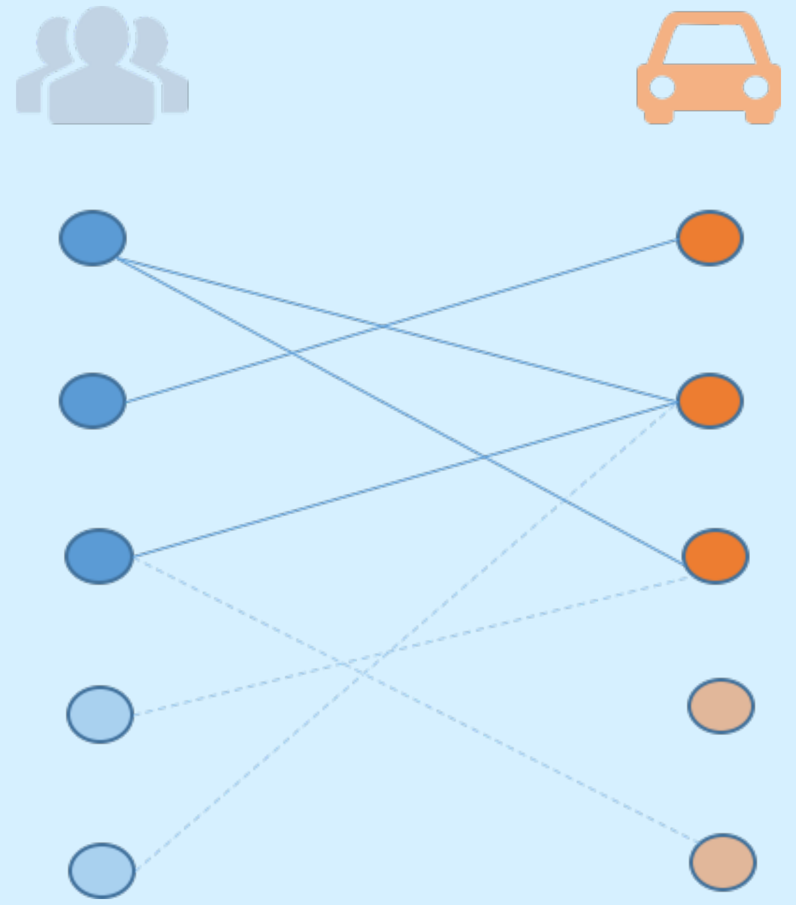
Offline Reinforcement Learning



+

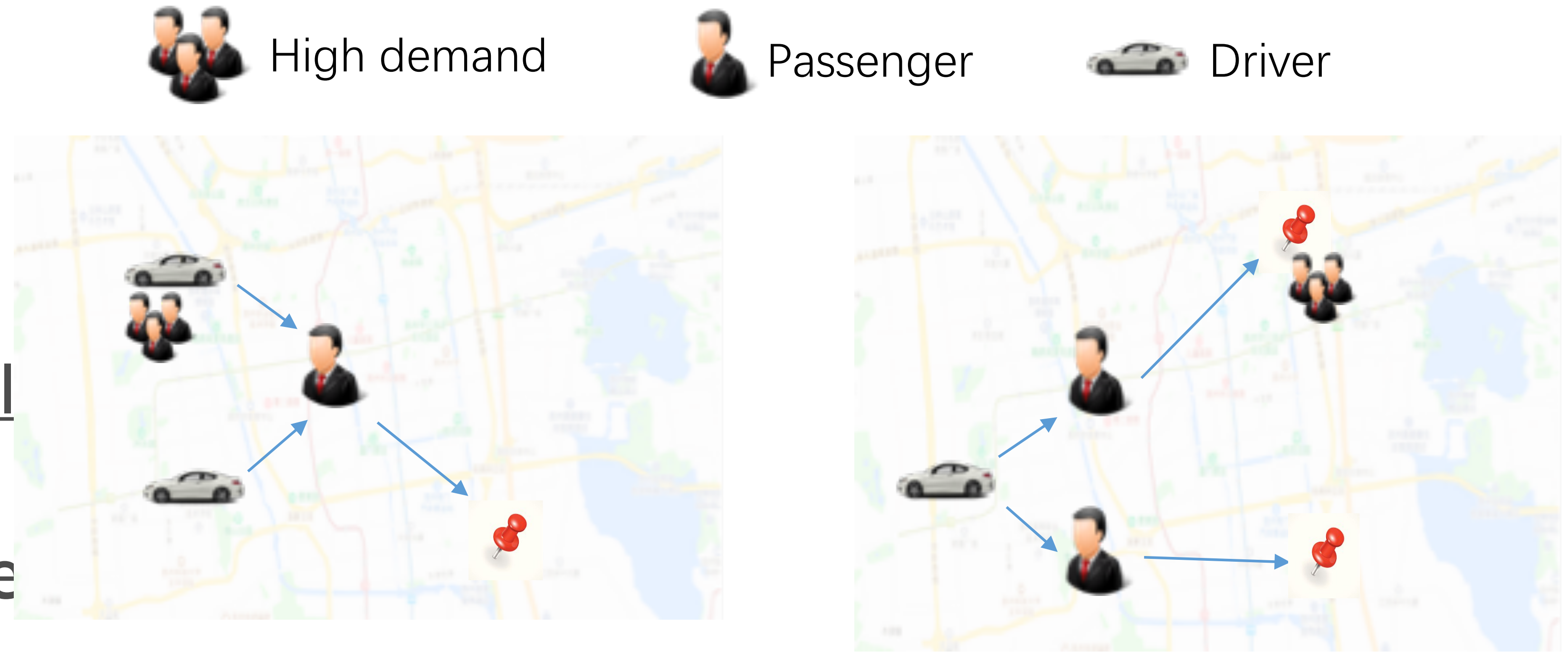
Bipartite Graph Matching





Spatiotemporal optimality!

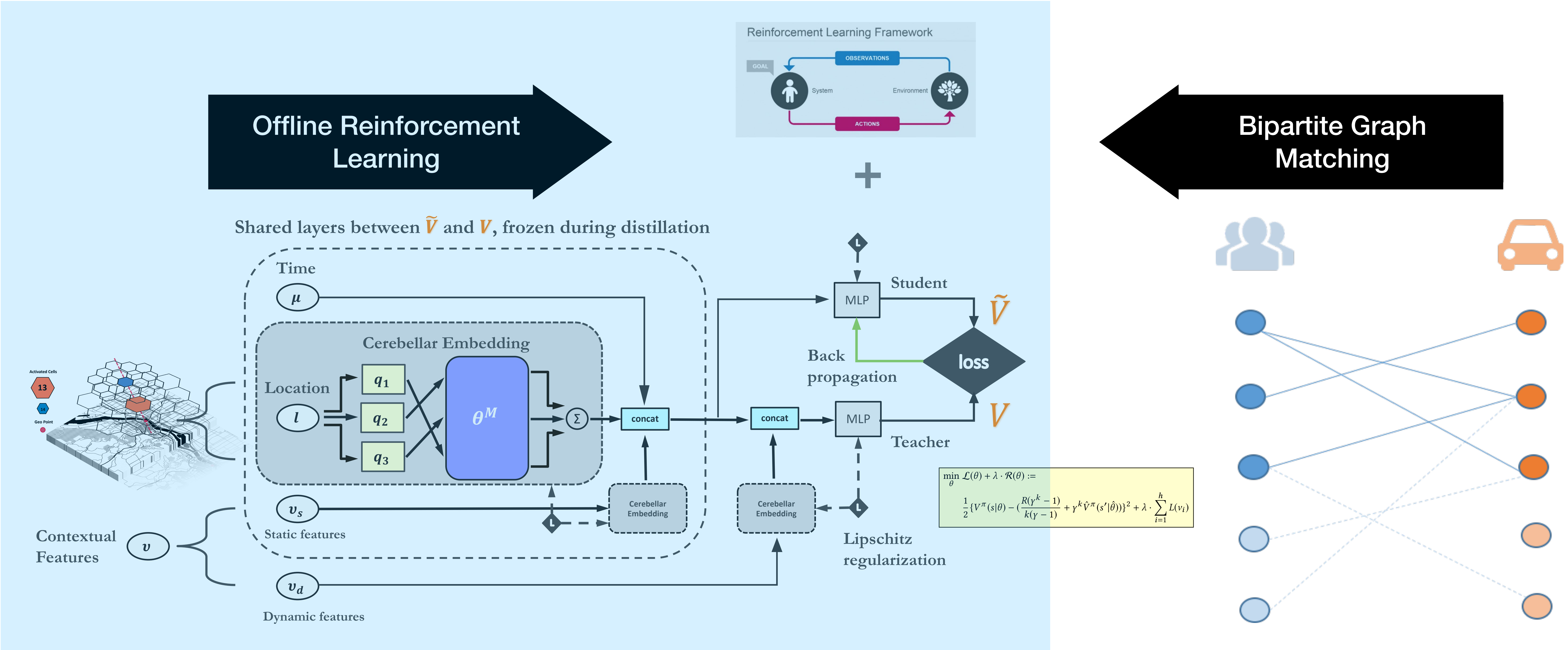
$$\rho_{ij} = R_{ij}\frac{(\gamma^{k_{ij}}-1)}{k_{ij}(\gamma-1)} + \underline{\gamma^{k_{ij}}V(s_j)} - \underline{V(s_i)} + \Omega \cdot U_{ij}$$



- Case study
  - **Left**: same pickup distance, driver features, etc. <u>Which one to dispatch</u>?
  - **Right**: same trip fee, pickup distance, passenger features, etc. <u>Which one to fulfill</u>
- The final <u>matching weight</u> captures both cases balancing between the value of <u>passenger's destination</u> and that of the <u>driver's current state</u>

DiDi

Background



Offline RL

*X. Tang et al., **KDD Oral** 2019*

$$\min_{\rho} L_{ope}(\mathcal{H};\rho) :=$$

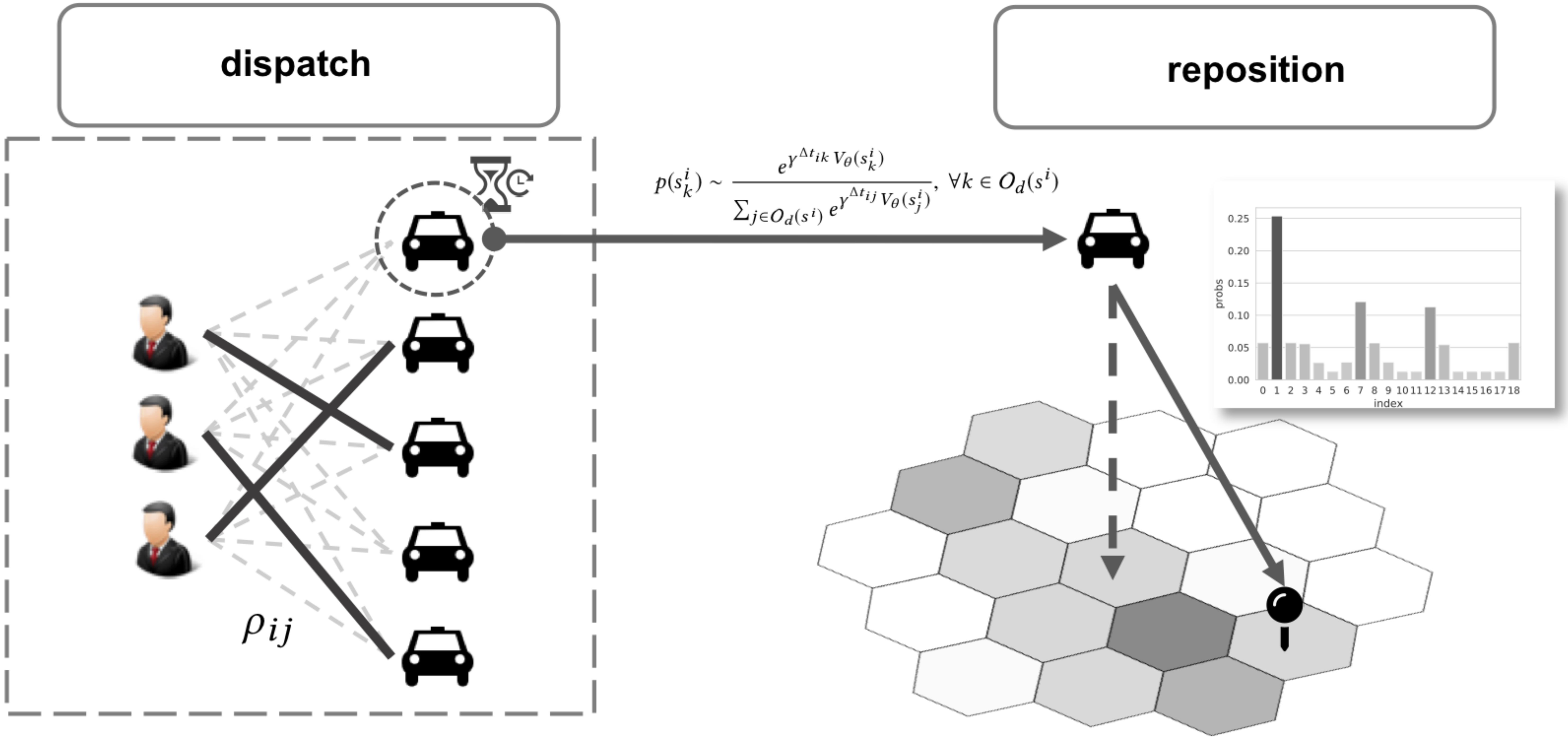
$$E_{(s,R,s')\sim\mathcal{H}}\left[\left(R+\gamma^{\Delta t}\hat{V}_{ope}(s',t'|\rho)-V_{ope}(s,t|\rho)\right)^2\right]+\lambda\cdot L_{reg}(\rho)$$

- Evaluated the value network on the hundreds of millions of historical driver trajectories based on a semi-MDP formulation
- Proposed the use of **Lipschitz regularization** on the value function for better offline RL performance
  - Kumar et al., 2020 makes the case that for TD-learning with function approximation the neural network is being implicitly under-parametrized with a drop in the rank of learned features
  - Gogianu et al., 2021 improves the performance of DQN by simply constraining the Lipschitz constant of a single layer, which also help preserve the rank of the features
- Context randomization, hierarchical coarse-coded embedding and multi-city progressive transfer for better generalization in the real world

DiDi

Challenges

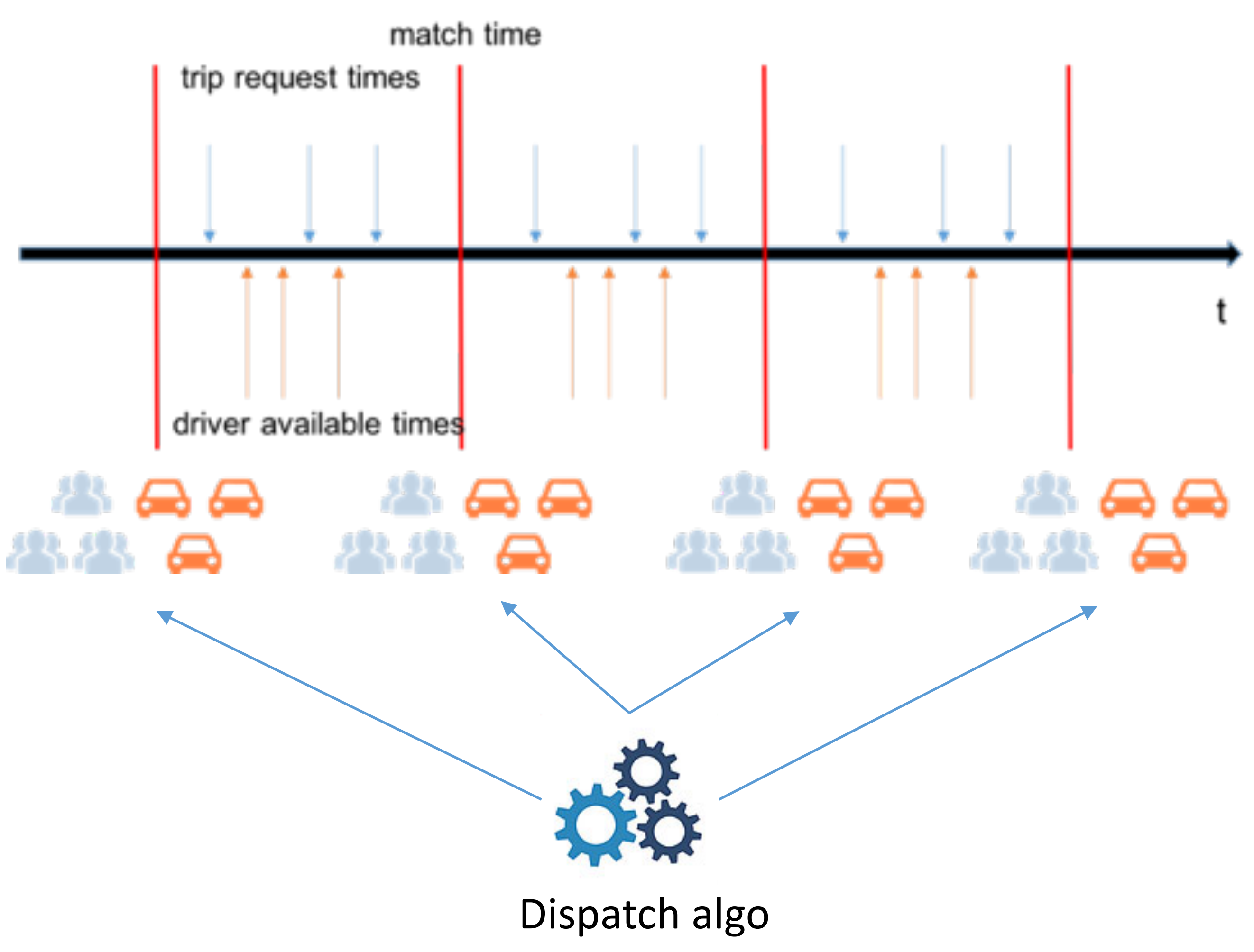
- Ride-hailing marketplace — **multi-task sequential decision problem**
  - Order dispatching and vehicle repositioning (autonomous fleet management)
  - Hundreds of thousands of decisions are made per day with extended temporal effects
  - Connecting tens of thousands of vehicles in a city to millions of ride demands continuously throughout the day

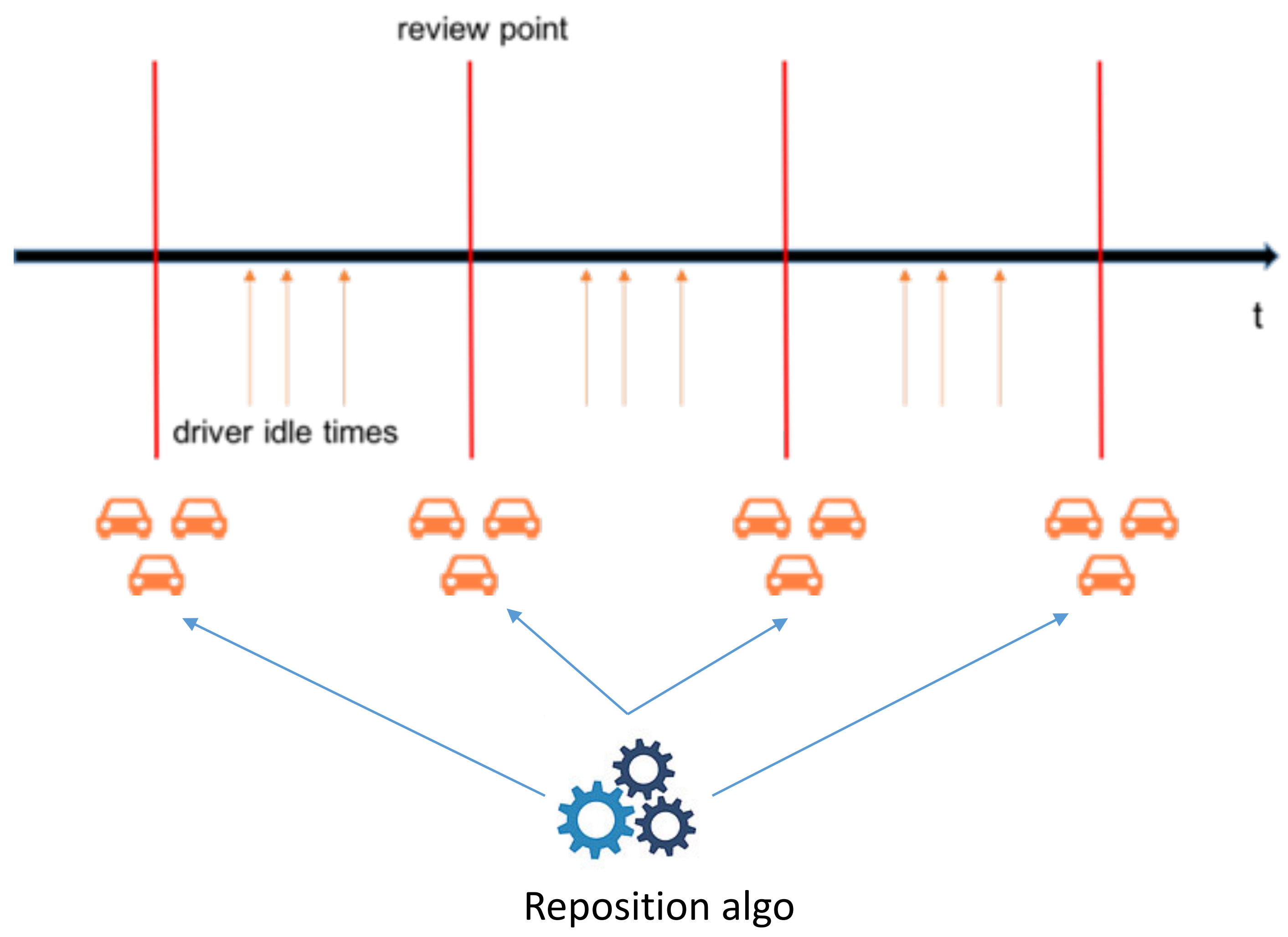


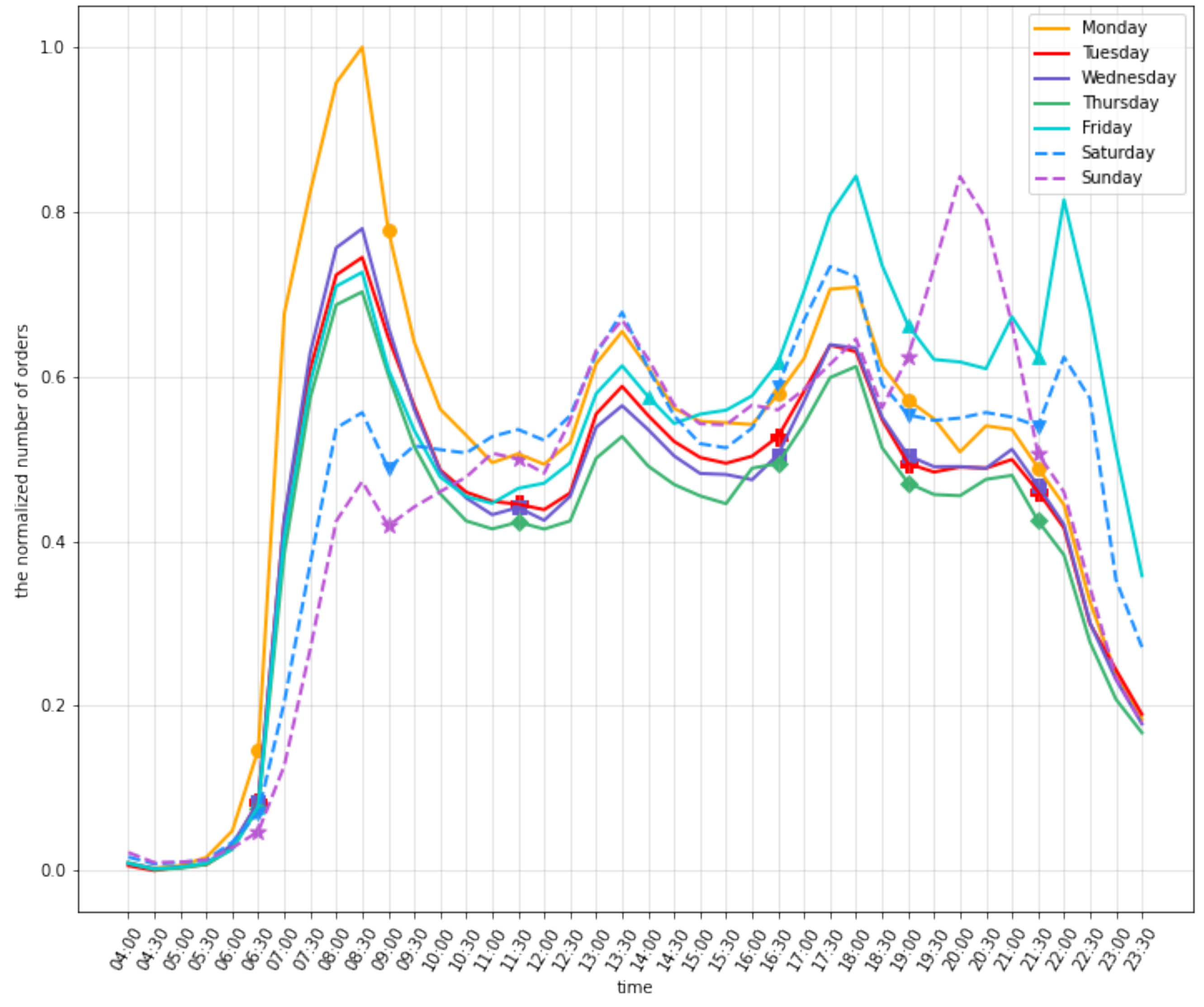
DiDi

Challenges

- **Real-time dynamics** between supply and demand in a stochastic and time-varying environment.
  - Daily recurrent variations usually have good representations in large historical datasets (offline RL)
  - Occurrences of irregular (long-tail) events some may never occur in the training data (online learning)
  - Additional contextual features are NOT good enough
- **Coordinations** among vehicles (multi-agent)
  - Resolve dispatching constraints and avoid undesirable competitions among managed vehicles
- **Interactions** between tasks (multi-task)
  - Both tasks modify the system state, e.g., supply/demand distributions, as well as the state transition dynamics, e.g., traffic on the road and the estimated arrival time.







DiDi

V1D3: Next Generation Decision Engine

Offline RL

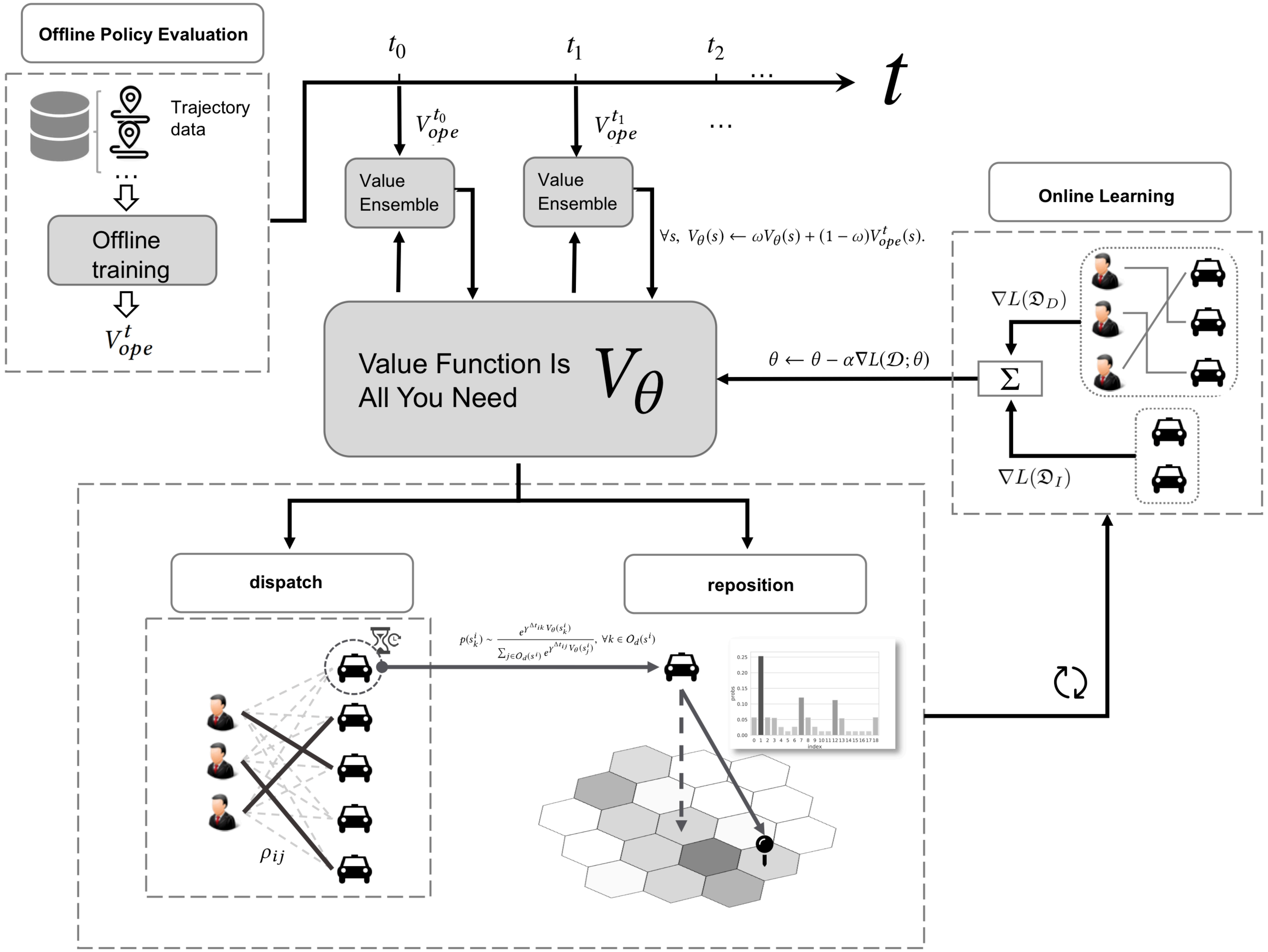
Ride-hailing Marketplace

Multi-task

Multi-agent

Online Adaptations

A unified value-based dynamic learning framework (V1D3) for both dispatching and repositioning

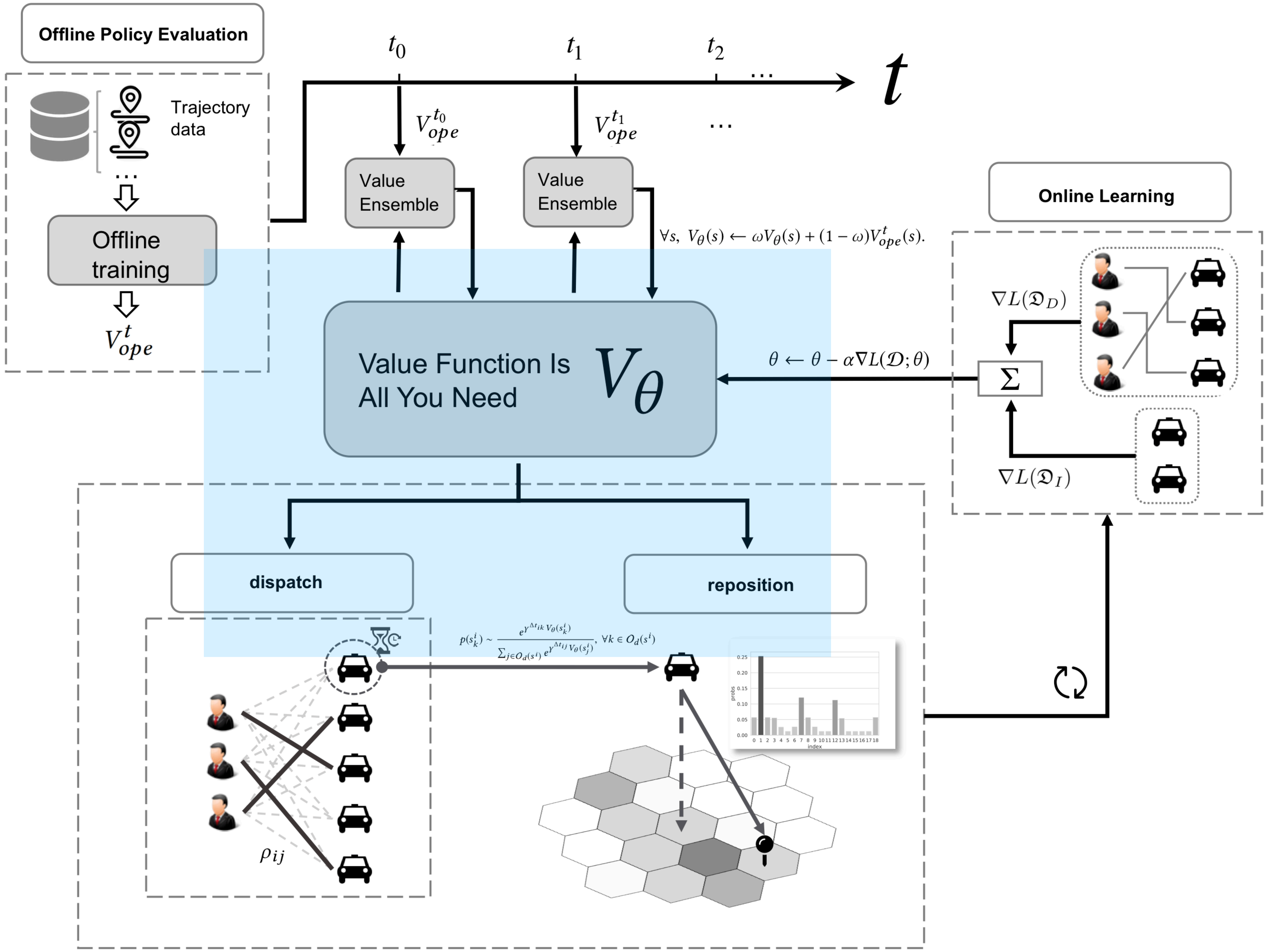


V1D3: Next Generation Decision Engine

√ At the center of the framework is a **globally shared value function** that is updated continuously to reflect in real time the platform transactions

- Both tasks rely on the shared value function for decision making
- Any changes on the global state made by dispatching and repositioning are communicated in real-time through the value function
- A "**feedback loop**" to reach equilibrium of supply and demand as an implicit form of coordinations

A unified value-based dynamic learning framework (V1D3) for both dispatching and repositioning



V1D3: Next Generation Decision Engine

A unified value-based dynamic learning framework (V1D3) for both dispatching and repositioning

✓ **Online adaptations** with the population-based TD learning objective obtained for each round of dispatch

- **Positive updates** from drivers successfully matched with passengers

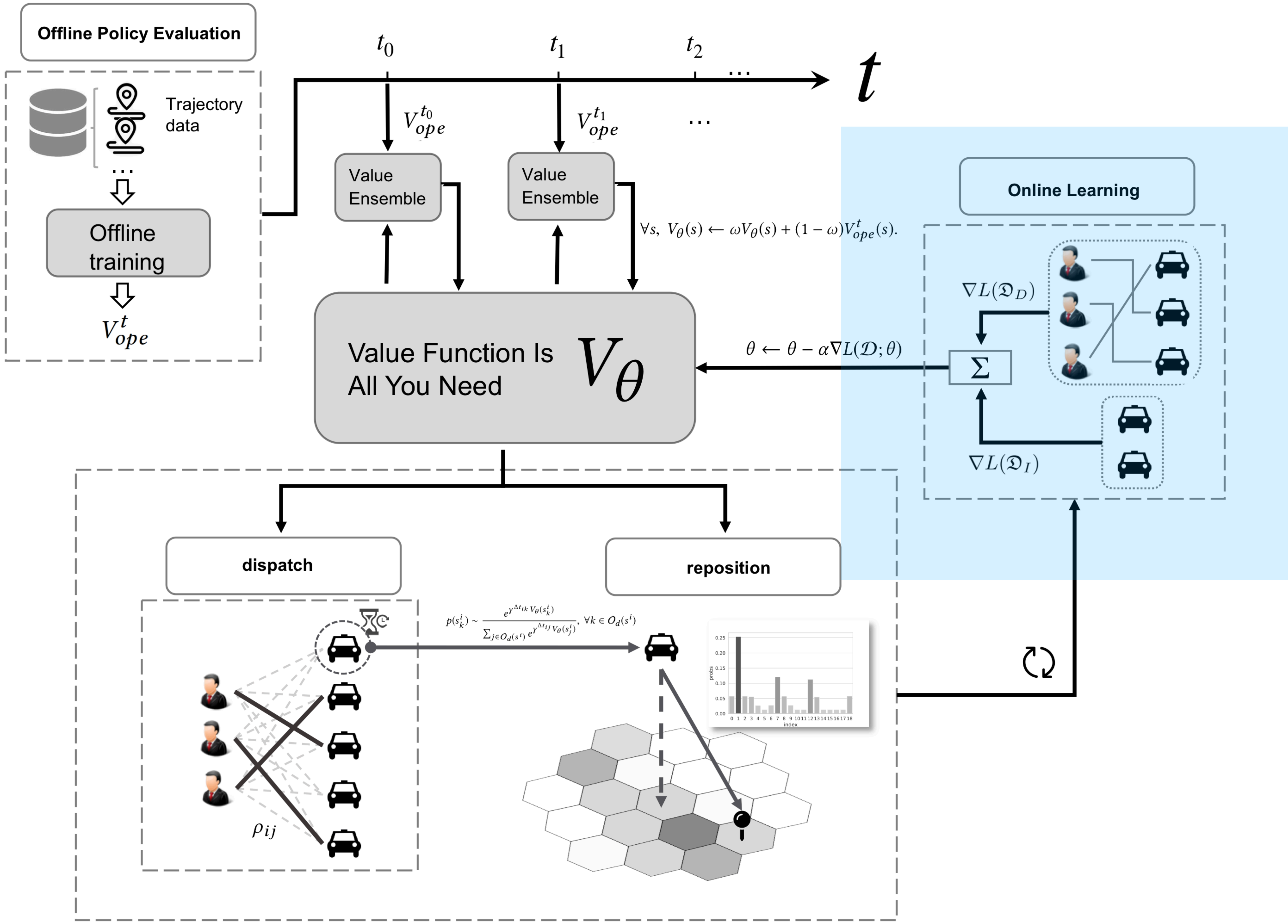
$$V(s^i_{driver}) \leftarrow r^i_{order} + \gamma^{\Delta t_{order}} V(s^i_{order})$$

- **Negative updates** from idling drivers

$$V(s^i_{driver}) \leftarrow 0 + \gamma^{\Delta t_{idle}} V(s^i_{idle})$$

- Intuitively positive updates increase the state value while negative updates decrease the corresponding ones. Together the objective is to minimize the population-based mean-squared TD error

$$\min_\theta L(\mathcal{D};\theta) := \sum_{i\in\mathcal{D}_D} (V_\theta(s^i_{driver}) - r^i_{order} - \gamma^{\Delta t_{order}} \bar{V}_\theta(s^i_{order}))^2 + \sum_{i\in\mathcal{D}_I} (V_\theta(s^i_{driver}) - \gamma^{\Delta t_{idle}} \bar{V}_\theta(s^i_{idle}))^2 = \sum_{i\in\mathcal{D}} (\delta^i_\theta)^2$$



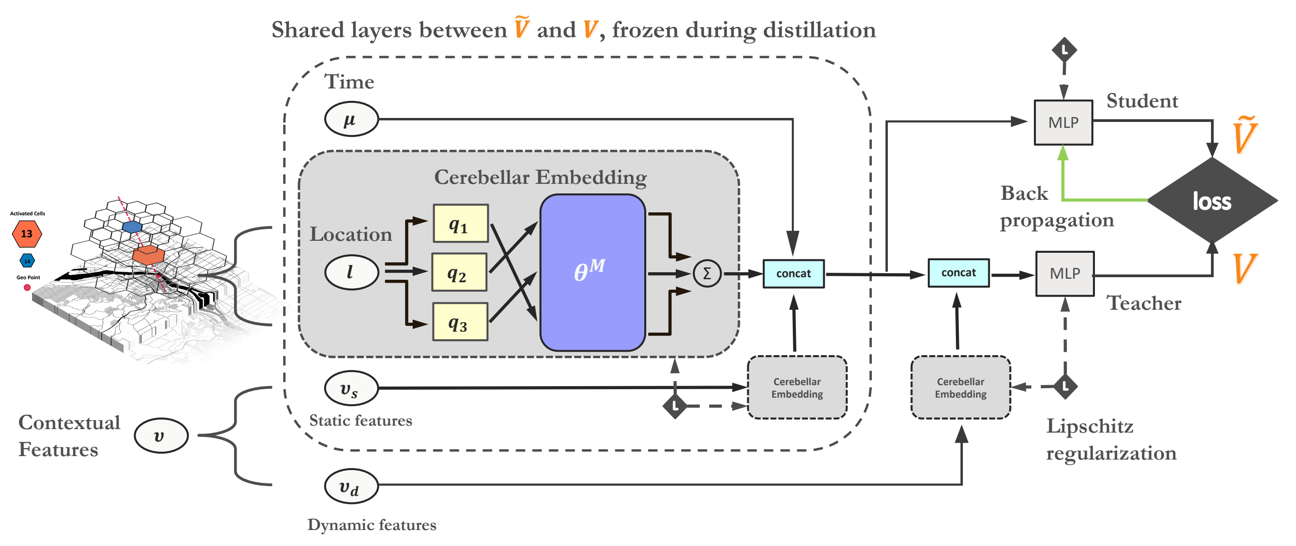
DiDi

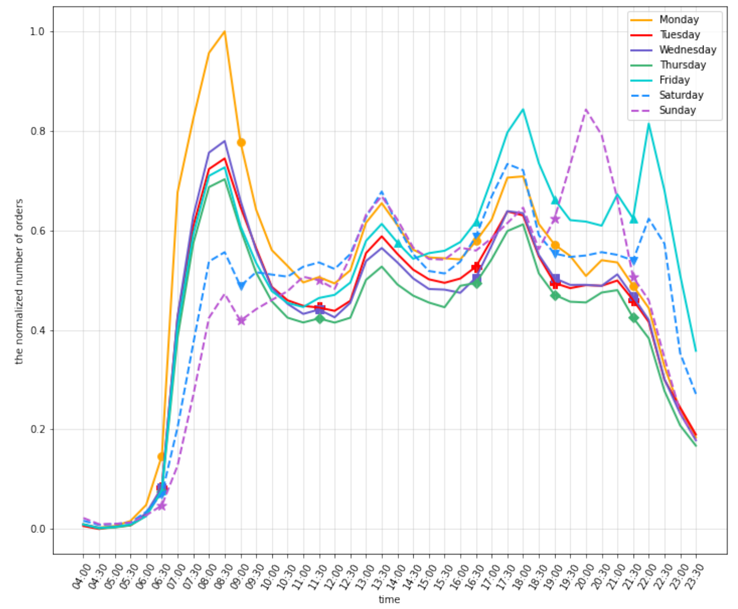
V1D3: Next Generation Decision Engine

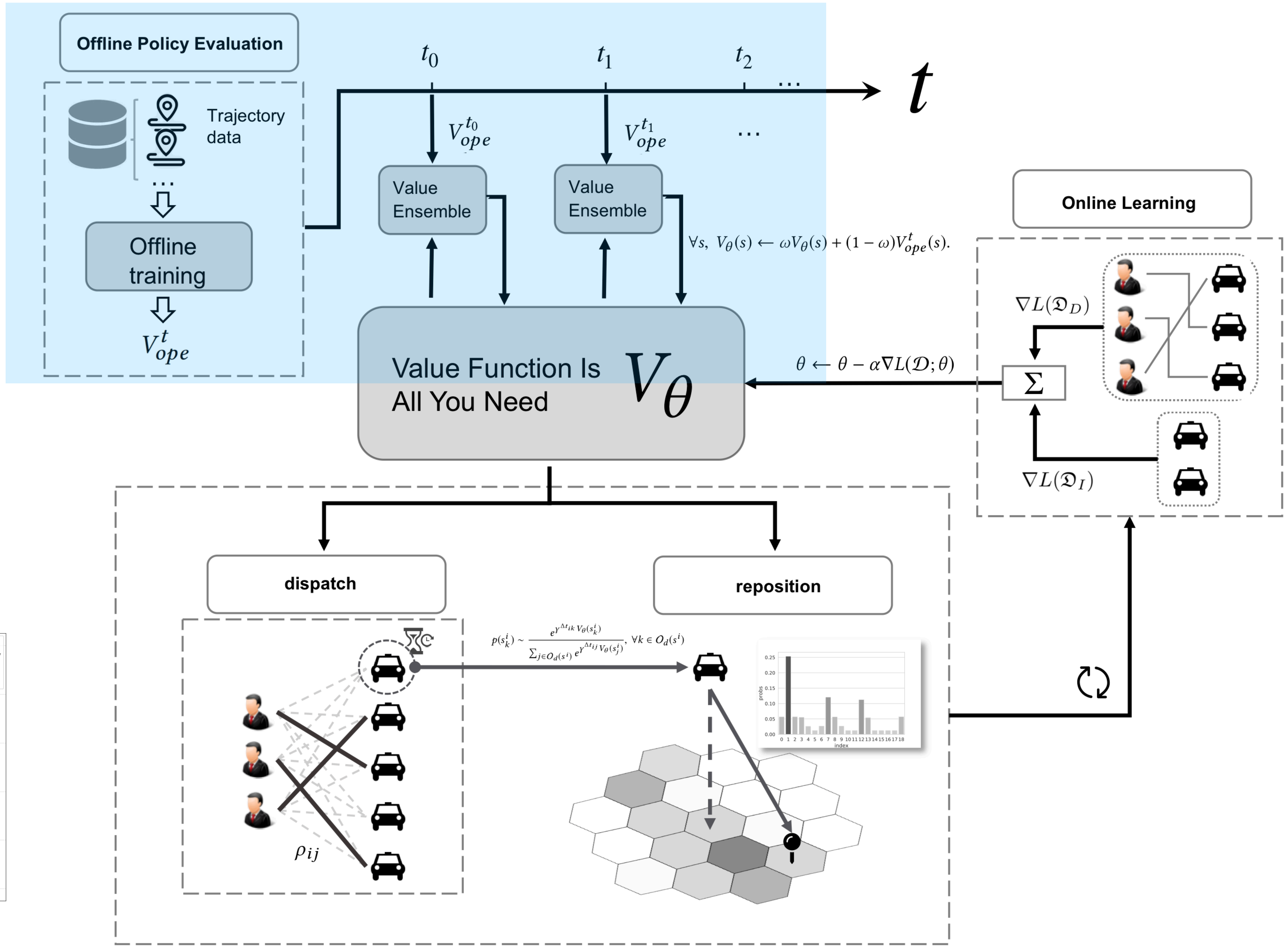
A unified value-based dynamic learning framework (V1D3) for both dispatching and repositioning

✓ **Periodic value ensemble** with **offline evaluated** time-sensitive policy for handling distributional shift in a time-varying non-stationary environment

- Lipschitz-regularized offline policy evaluation with time stamp inputs to obtain a time series of state value functions
- Periodically 'reinitialize' with a weighted ensemble scheme and a pre-determined set of ensemble time points from learning a segmentation on the historical aggregated order time series







DiDi

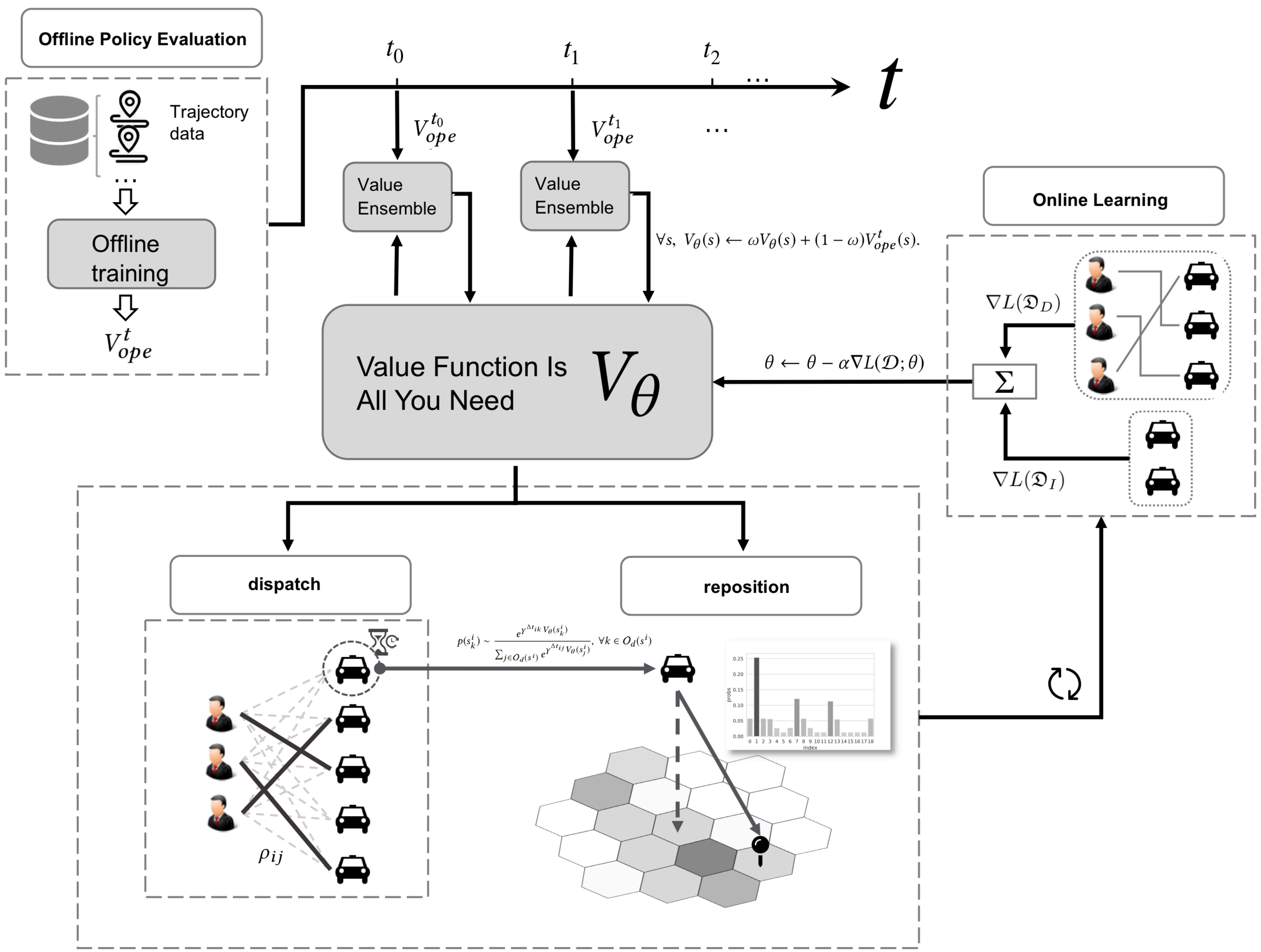
V1D3: Next Generation Decision Engine

✓ **Sample-efficiency** and **robustness**: the novel periodic ensemble method combining the fast online learning with a large-scale offline training scheme that leverages the abundant historical driver trajectory data

- **Adapt** quickly to the highly dynamic environment,
- **Generalize** robustly to recurrent patterns
- **Drive** implicit coordinations among the population of managed vehicles

✓ **V1D3** outperforms both first prize winners of dispatching and repositioning tracks in the KDD Cup 2020 RL competition, achieving state-of-the-art results on improving both **total driver income** and **user experience** related metrics

A unified value-based dynamic learning framework (V1D3) for both dispatching and repositioning

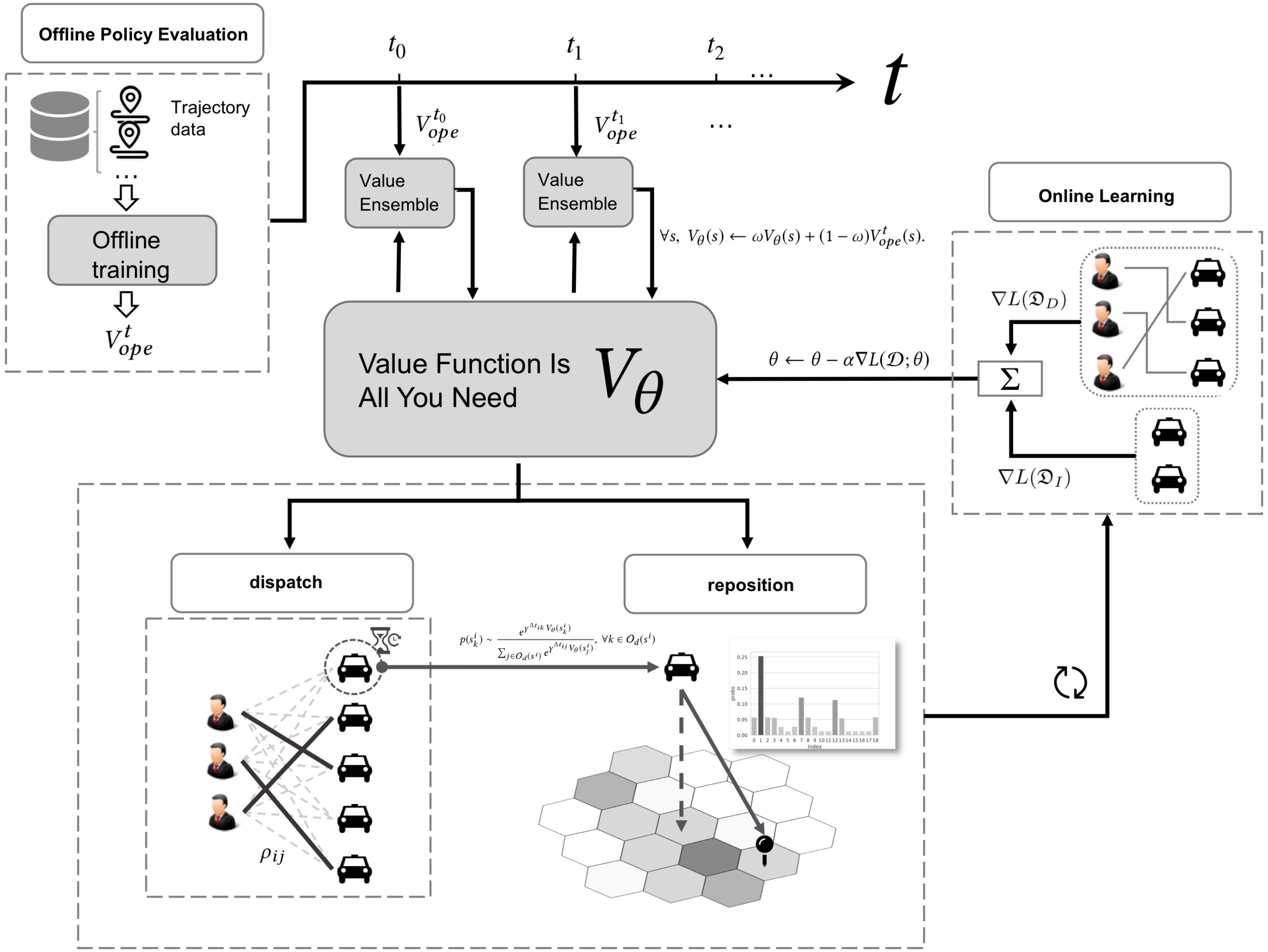


V1D3: Next Generation Decision Engine

Algorithm 5.1 Unified **V**alue Learning Framework for **D**ynamic Order **D**ispatching and **D**river Repositioning (**V1D3**)

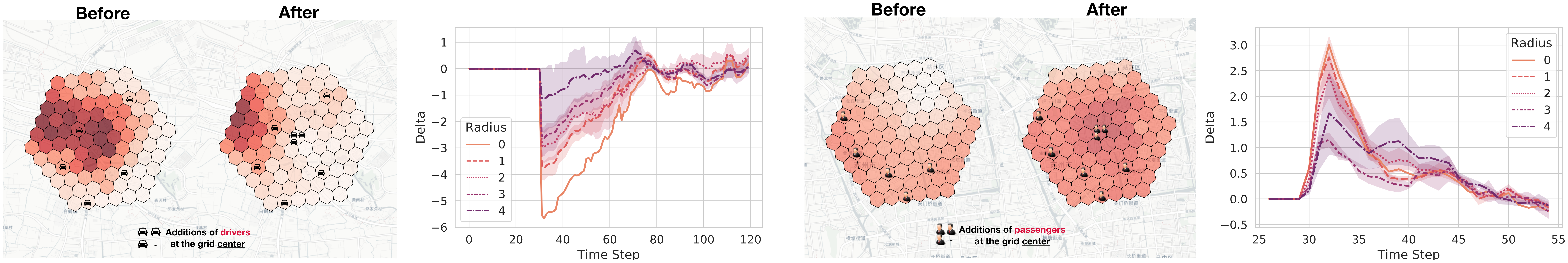
Given: the ensemble weight $1 > \omega > 0$, the reposition threshold $C > 0$ (usually chosen between 150 and 300).
Given: the offline evaluated value function V_{ope}.
Compute the set $\mathcal{E}$ containing the changing time points to re-ensemble.
Initialize the state value network V with random weights θ.
for the dispatch round $t = 1, 2, \cdots, N$ **do**
    if $t \in \mathcal{E}$ **then**
        $\forall s,\ V_\theta(s) \leftarrow \omega V_\theta(s) + (1-\omega)V^t_{ope}(s)$.
    end if
    Solve the dispatch problem (7) given the current value V_θ.
    if $t \mod C = 0$ **then**
        Collect all drivers with idle time exceeding C time steps.
        Compute the destination distribution (8) for each driver given the current value V_θ.
        Reposition each driver stochastically according to the distribution.
    end if
    Obtain the system state $\mathcal{D}_D$, $\mathcal{D}_I$ and $\mathcal{D} = \mathcal{D}_D \cup \mathcal{D}_I$.
    Construct the gradient of the learning objective (4), i.e., $\nabla L(\mathcal{D};\theta)$ based on the current system state $\mathcal{D}$.
    Update the state value network by performing a gradient descent step on θ, e.g., $\theta \leftarrow \theta - \alpha\nabla L(\mathcal{D};\theta)$
end for
return V

A unified value-based dynamic learning framework (V1D3) for both dispatching and repositioning



DiDi

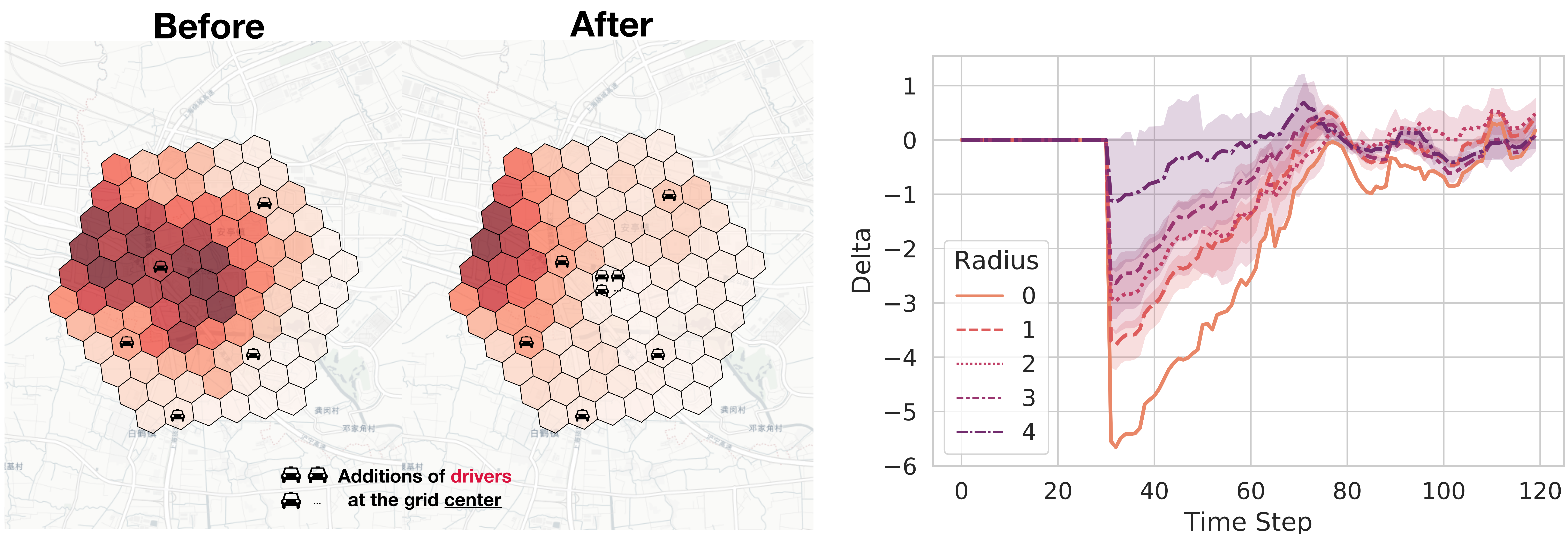
V1D3: Next Generation Decision Engine



Simulate the response curve of **V1D3**'s value function according to the change of **supply** and **demand**.

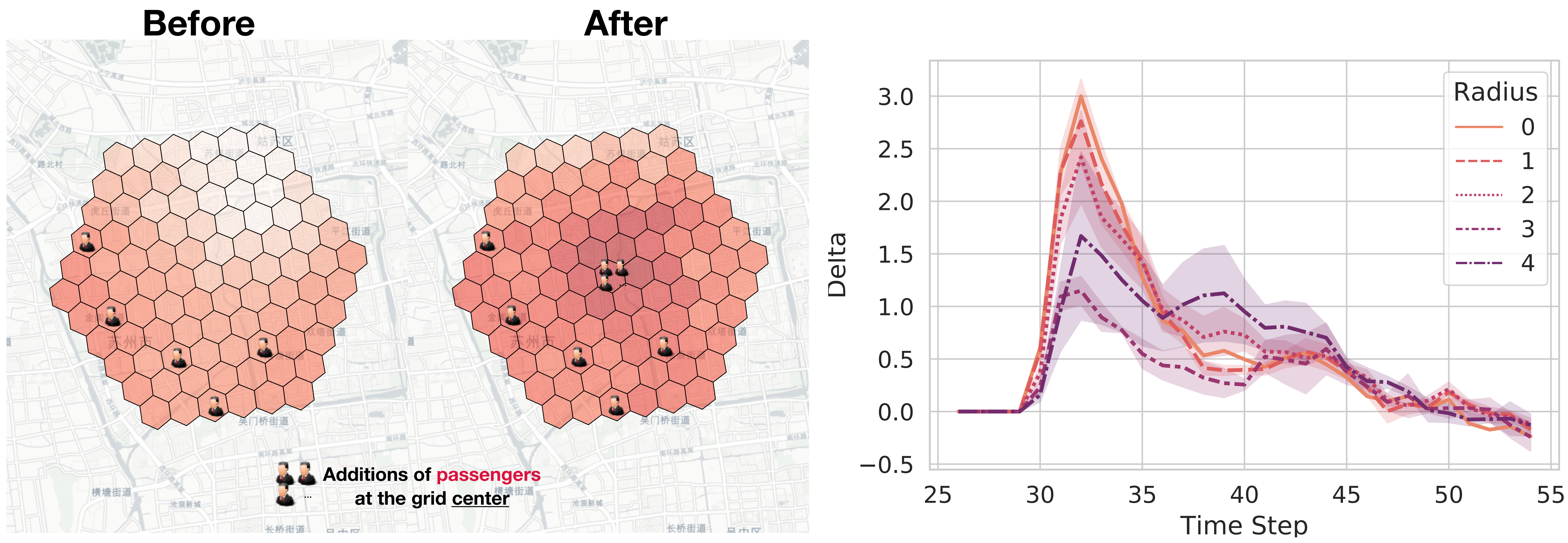
DiDi

V1D3: Next Generation Decision Engine



- The presence of **additional drivers** quickly brings **down** the value
- The values gradually return to stable state after the additional **supply** is consumed (feedback loop)

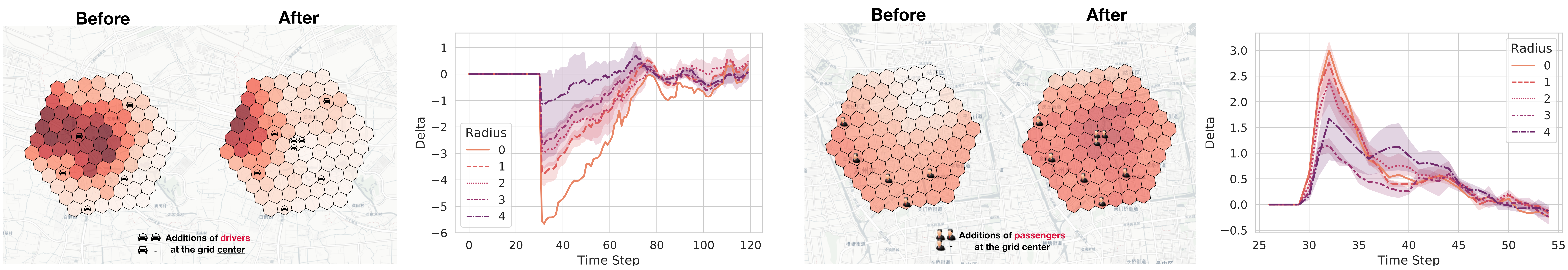
V1D3: Next Generation Decision Engine



- The presence of **additional orders** quickly brings **up** the value
- The values gradually return to stable state after the additional **demand** is consumed (feedback loop)

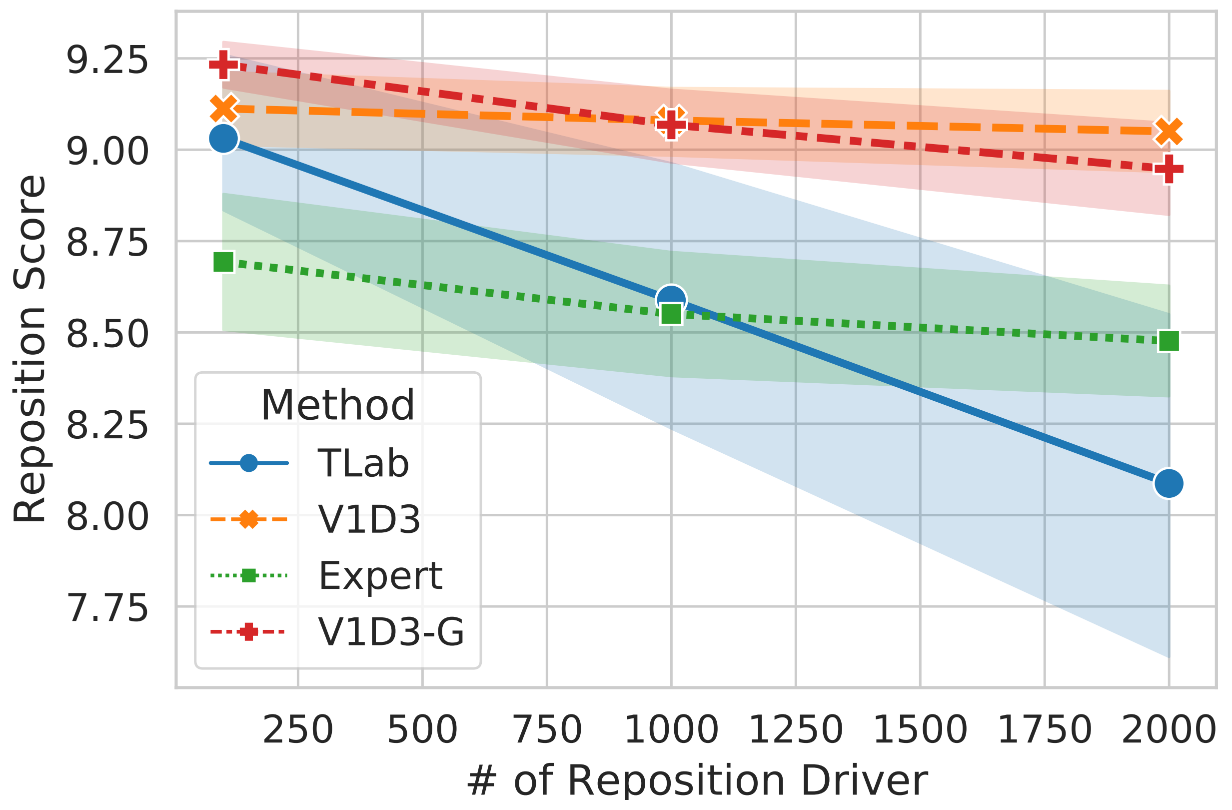
DiDi

V1D3: Next Generation Decision Engine

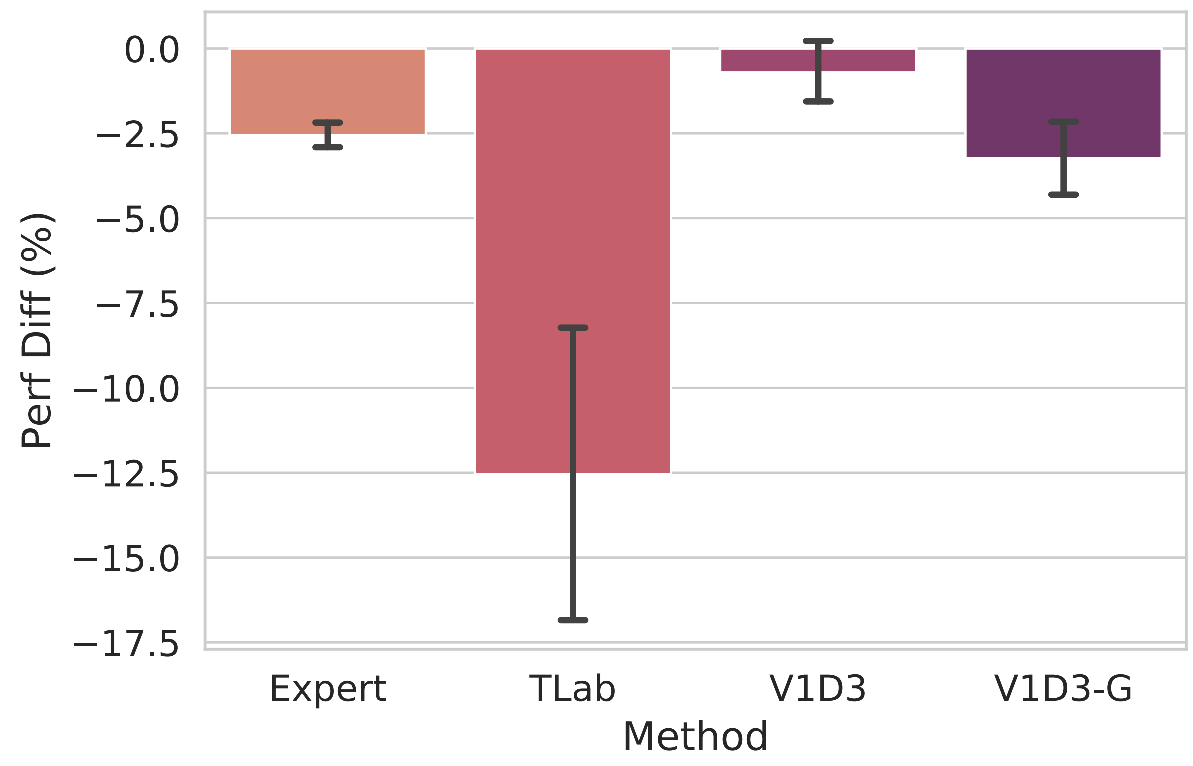


- In both cases the **smoothness** property of the value function allows the magnitude of the response to **gradually decrease** as we move away from the center of the event

V1D3: Next Generation Decision Engine



(a)



(b)

Table 1: Comparison with state-of-the-art dispatching algorithms in simulating environments using real-world data from DiDi's ride-hailing platform during both weekdays and weekends in three different cities. The results are averaged from multiple days and the means and variances across days are reported.

City	Environment	Method	Dispatch score	Answer rate (%) †	Completion rate (%) †
City A	Weekday	PolarB	2498023.82 ± 12517.26	+2.8398 ± 0.3638	+1.8177 ± 0.3192
		Baseline	2387008.73 ± 5429.38	+0.0000 ± 0.0000	+0.0000 ± 0.0000
		CVNet	2398814.43 ± 12839.90	**+3.7166 ± 0.3602**	+0.6548 ± 0.3540
		Greedy	2350685.21 ± 5567.51	-1.2964 ± 0.0603	-3.6622 ± 0.0008
		V1D3	**2509547.65 ± 8794.37**	+3.0823 ± 0.0653	**+2.0828 ± 0.0338**
	Weekend	PolarB	2577002.60 ± 91071.56	+2.0634 ± 0.4399	+0.9494 ± 0.4347
		Baseline	2487915.88 ± 77111.26	+0.0000 ± 0.0000	+0.0000 ± 0.0000
		CVNet	2534253.10 ± 84285.72	**+4.9861 ± 0.1908**	**+1.6428 ± 0.2126**
		Greedy	2430412.20 ± 77133.57	-1.5470 ± 0.4394	-4.2193 ± 0.3719
		V1D3	**2590333.62 ± 99474.20**	+2.5222 ± 0.1956	+1.3679 ± 0.1300
City B	Weekday	PolarB	1575231.41 ± 29200.11	+2.5077 ± 2.0896	+1.1372 ± 1.9432
		Baseline	1498126.49 ± 12037.66	+0.0000 ± 0.0000	+0.0000 ± 0.0000
		CVNet	1511983.792 ± 12331.36	+2.6405 ± 0.3073	+0.2856 ± 0.2215
		Greedy	1498385.19 ± 30811.10	+1.2401 ± 1.4075	-1.3727 ± 1.3386
		V1D3	**1589252.82 ± 20981.18**	**+3.7677 ± 0.7358**	**+2.4352 ± 0.5846**
	Weekend	PolarB	1436435.90 ± 52206.43	+1.3003 ± 1.4210	-0.2523 ± 1.5487
		Baseline	1402633.35 ± 33007.10	+0.0000 ± 0.0000	+0.0000 ± 0.0000
		CVNet	1407527.12 ± 38468.35	**+2.5140 ± 1.4626**	-0.8369 ± 1.5392
		Greedy	1388862.54 ± 46301.08	+0.6618 ± 0.6337	-2.3576 ± 0.9062
		V1D3	**1453191.10 ± 40822.98**	+2.4246 ± 0.2247	**+0.8618 ± 0.2460**
City C	Weekday	PolarB	767201.73 ± 33299.30	-3.0291 ± 3.6575	-3.8274 ± 3.4695
		Baseline	738083.83 ± 44261.91	+0.0000 ± 0.0000	+0.0000 ± 0.0000
		CVNet	744578.48 ± 42294.09	**+6.3528 ± 0.1955**	+2.7810 ± 0.6404
		Greedy	724491.04 ± 46843.13	-3.1926 ± 0.8896	-5.6701 ± 0.4511
		V1D3	**778687.02 ± 48186.72**	+4.8733 ± 0.0938	**+2.9925 ± 0.0934**
	Weekend	PolarB	804656.13 ± 15354.59	-1.9825 ± 2.9749	-2.8981 ± 2.9205
		Baseline	764460.73 ± 4893.10	+0.0000 ± 0.0000	+0.0000 ± 0.0000
		CVNet	780972.50 ± 18303.07	**+7.0296 ± 2.4580**	**+4.3322 ± 2.4390**
		Greedy	746729.07 ± 3357.45	-4.1320 ± 0.8392	-5.8998 ± 0.5004
		V1D3	**825870.31 ± 7756.72**	+1.6107 ± 1.1763	+0.5496 ± 0.8569

† The reported numbers are relative improvement computed against the Baseline.

Dispatch

- ✓ Experiments include both weekdays and weekends in three different cities
- ✓ Outperform methods including KDD Cup winner PolarB and published algorithms such as CVNet and strong baselines
- ✓ **V1D3** combines the advantages of both **PolarB** (pure online) and **CVNet**[1] (pure offline)
  - increases total driver income by as much as **+8%** against the Baseline, **+6%** against previous SOTA CVNet and **+3%** against PolarB
  - Increases user experience by as much as **+8%** against PolarB

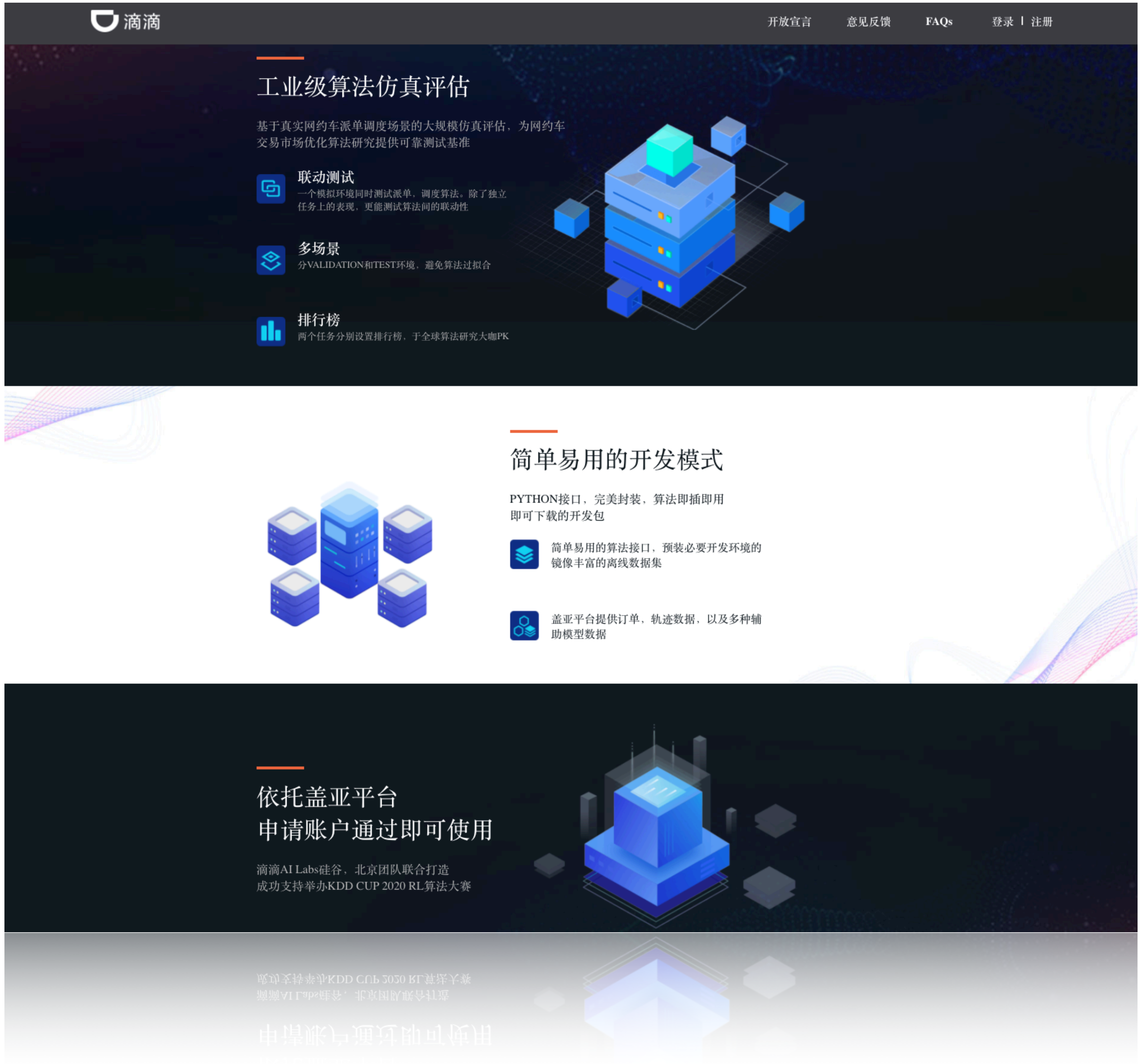
Reposition

- ✓ Experiments include varying the size of the managed fleet, for each fleet size averaging over five different days
- ✓ Outperform KDD Cup winner **TLab**[2] and **human expert policy**
- ✓ **V1D3** achieves more than **+6%** improvement in driver income rate over the human expert policy
- ✓ **V1D3** outperforms TLab by **15x** in robustness as the fleet size increases **20x**

1. X. Tang et al, A Deep Value-network Based Approach for Multi-Driver Order Dispatching, *Oral, acceptance rate 6%, SIGKDD* 2019
2. Y. Liu et al, Learning to reposition on an online taxi-hailing platform. preprint, 2021

DiDi

V1D3: Next Generation Decision Engine



Open ride-hailing marketplace simulation platform

- https://outreach.didichuxing.com/Simulation/

Link to full paper

- https://arxiv.org/abs/2105.08791

THANK YOU

滴滴
滴滴一下 美好出行

北京嘀嘀无限科技发展有限公司
北京市海淀区东北旺路8号院尚东·数字山谷B1号楼