An Integrated Digital Twin and Reinforcement Learning Solution for Ride-Hailing Repositioning

Xinkai Xu, Naga Sesha Sai, Zihan Ma, Chun Wang*
xinkai.xu@mail.concordia.ca; chun.wang@concordia.ca
Concordia Institute for Information Systems Engineering, Concordia University, Montreal
* Corresponding author

1. Idle vehicle repositioning in ride-hailing

Idle vehicle repositioning is crucial for the efficient operation of most ride-hailing systems, where rider demand and driver supply often do not align across different locations and times. This imbalance negatively affects both drivers and riders by reducing operational efficiency and service quality. Currently, when drivers become available after dropping off passengers, the platform does not provide specific guidance on where they should go next to anticipate future rider requests. As a result, drivers must rely on their own experience to predict demand and travel to areas where they believe new requests will emerge. If the actual demand does not meet their expectations or if too many drivers converge in the same area, they may end up waiting for long periods without receiving any ride assignments. Consequently, they must reassess the situation and move to other regions. This ad-hoc prediction-based approach to finding rides leads to high operational costs for drivers and creates negative social impacts, such as increased urban congestion and emissions[1].

To address this issue, ride-hailing systems have implemented indirect vehicle repositioning guidance strategies like surge pricing to draw drivers to areas where demand exceeds supply, potentially reducing wait times [2]. However, with surge pricing, drivers can only gauge demand based on surge price levels, and their relocation decisions are uncoordinated. This often leads to an oversupply of surge chasing drivers in some areas while worsening supply shortages in others, resulting in longer overall wait times for riders across regions and increased idle driving costs for the drivers [3], [4]. Optimal repositioning in ride-hailing is crucial for drivers, riders, and platforms. By strategically repositioning, drivers can minimize the idle time spent waiting for their next ride, which increases overall efficiency leading to more rides per hour, increasing the revenue for both drivers and the ride-hailing platform. Driver satisfaction and retention can also be improved. Optimal repositioning helps ensure that vehicles are available where and when they are needed most, reducing waiting times for riders, enhancing the overall customer experience, leading to higher satisfaction and potentially more repeat customers.

Existing approaches to repositioning problem in ride-hailing can be categorized into reactive and proactive strategies [7]. Reactive strategies attempt to rebalance vehicles within the system after rider demands in different regions are realized. Given the existing vehicle positions and rider demands, these methods optimize the vehicle-rider assignments to reduce idle detour miles and rider wait times [9]. Overall system performance can also be improved by strategically rejecting demand at low-demand locations and inducing driver repositioning to high-demand locations [10]. On the other hand, repositioning strategies that anticipate riders' demands before they are realized are proactive. These strategies relocate the vehicles to the regions where high demand is expected. For example, Model predictive controller (MPC) leverages short term rider demand forecast to optimize the control actions over a set of time horizons [5] [6]. Time-Series Model based demand prediction methods such as ARIMA [7], and Graph Neural Networks (GNNs) [8] are used for learning spatial-temporal rider demand patterns and guiding repositioning decisions. Most proactive repositioning methods consist of repeated optimization based on prediction on historical data repeatedly over a moving time horizon to choose the next control action.

In addition, reinforcement learning (RL) has emerged as a powerful tool for optimizing taxi repositioning and dispatching decisions for independent drivers who maximize their own reward functions [11] and for tackling the full fleet management and dispatching problems at the systems level [12]. Among other approaches, multiagent reinforcement learning are used to address heterogeneity of drivers in a region [13]. In [11], a practical deep RL method are designed for large-fleet repositioning. The authors recruited 1200 drivers from a large ride-hailing platform participating in their data collection, model design and validation processes, which provides first-hand experience on applying their proposed RL model to the repositioning problem in ride-hailing. Compared with commonly used prediction and optimization-based methods such as MPC, RL models can compute large-scale repositioning solutions faster. However, most proposed RL models for repositioning are off-line in the sense that they lack real-time adaptability and react only to historical data, without considering newly introduced changes in the environment, for example, ride demand and traffic pattern shifting and added transport infrastructures such as newly installed charging stations. Since these changes were not represented in the historical data, the RL models may compute repositioning decisions that are far from optimal given the changed environment and, in some settings such as EV taxi fleet, are infeasible to execute due to EV range limits and charging constraints.

We propose an integrated digital twin (DT) and reinforcement learning framework for making driver repositioning decisions. By leveraging the synthetic data generated by the digital twin of the ride-hailing environment, the RL model are trained on changes and scenarios which do not appear or are not sufficiently represented in the available historic data sets. The digital twin also allows virtual testing of repositioning strategies on various fleet configurations and future infrastructure expansions. With a modular scalable design, our system adapts to evolving urban mobility, including autonomous and electric vehicle strategies.

2. Integrated digital twin and reinforcement learning framework

The digital twin architecture design We propose a digital twin (DT) architecture for implementing urban mobility digital twins in mobility-on-demand (MoD) settings, called MoD-DT. Using the MoD-DT architecture, designer can flexibly implement digital twins for mobility systems with various types of datasets, road networks, operation algorithms and strategies. MoD-DT provides structural support for integrating machine learning (ML) modules in digital twins. It serves as a test bed and playground for testing the performance of various ML algorithms in MoD settings.

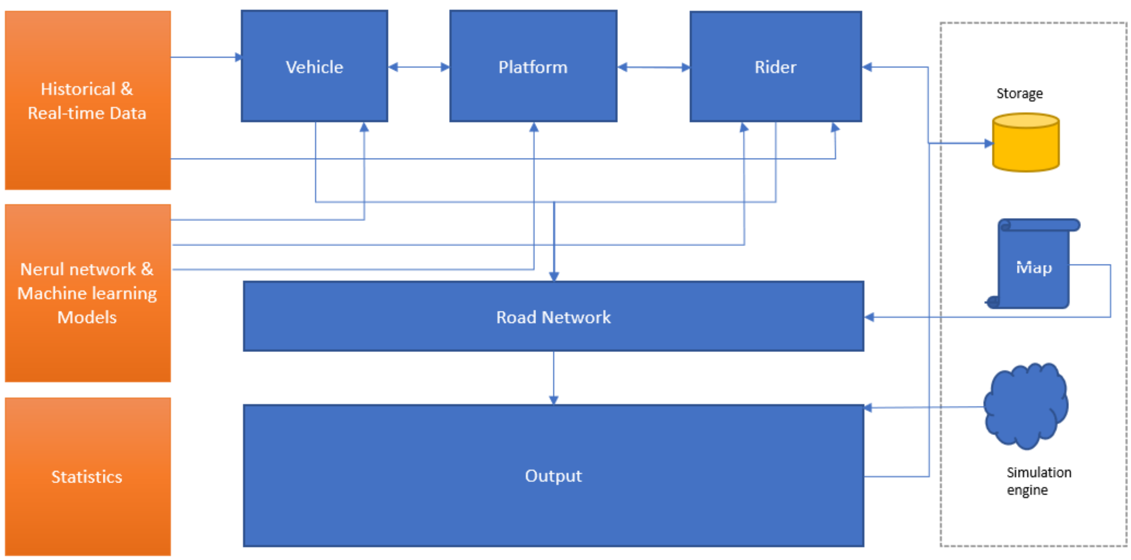

Figure 1: MoD-DT architecture design

Figure. 1 illustrates the structure and components of the proposed MoD-DT. The framework consists of five modules, namely (1) Vehicle; (2) Rider; (3) Platform; (4) Road network; (5) Outputs.

- Various agent structures, preferences and decision-making models are included in the Vehicle module for constructing driver agents representing the drivers and their vehicles in classical ride-hailing settings or robotaxi agents representing the autonomous driving vehicles and their owners. The vehicle can be either electric or internal combustion. Driver agents use neural networks and machine learning, such as reinforcement learning, for decision-making. Their internal states (location, battery level, ride status, assigned trips) are updated by the digital twin, while external states like traffic, demand, surge pricing, and charging station availability are also provided to the driver agent by the digital twin.
- Similarly, we design agent structures, preferences, and decision-making models in the Rider module to construct rider agents representing the riders in the ride-hailing systems. Rider module has the functionality of generating real-time rider requests based on current and projected ride demands. Neural networks and machine learning models are used for rider request generation and demand prediction.

- The Platform module incorporates optimization models, machine learning algorithms, to construct *central operation control* functions of the MoD-DT, such as matching ride requests with available drivers, minimizing wait times, considering factors such as distance, availability, and charging levels when dispatching vehicles. The module also continuously monitors the MoD environment, such as traffic conditions, rider demands and makes predictions in terms of traffic patterns.
- The Road Network Module provides location information for both drivers and riders. It integrates OpenStreetMap and AnyLogic simulation to calculate optimal routes, supplies traffic and congestion data, positions and configures charging stations for EVs. It also provides real-time locations of vehicles and riders to facilitate vehicle dispatching operations by the central operation control platform.
- The output module generates simulation visualizations and performance reports. It also evaluates repositioning efficiency in real time and posts key KPIs. As shown in Figure 2, the user interface consists of a map section which shows the real-time position and movement of vehicles and riders and four information windows. The black window on the left-hand side of the map shows the dynamic log of the state changes and actions of the vehicles, handling order, pickup, drop-off and charging. The upper three windows define map legend (top-left), the simulation setup and matching status (top-middle), and show the comparation of the repositioning efficiency (expected incomes in this case) of different repositioning strategies.

Integrating reinforcement learning for ride-hailing repositioning As mentioned in the previous subsection, MoD-DT provides structural support for integrating machine learning modules in the digital twin implementations. For example, for the purpose of providing drivers with repositioning decisions, RL models can be used by driver agents to compute optimal reposition locations after dropping off a rider. The RL model can be trained using the synthetic data generated from the MoD digital twin based on the scenarios generated based on historical ride request transaction data and added features such as charging networks to be installed. In the meantime, the digital twin also provides a safe virtual environment to train and validate RL models and to execute workflow simulations to test and analyze different operational strategies before real-world deployment.

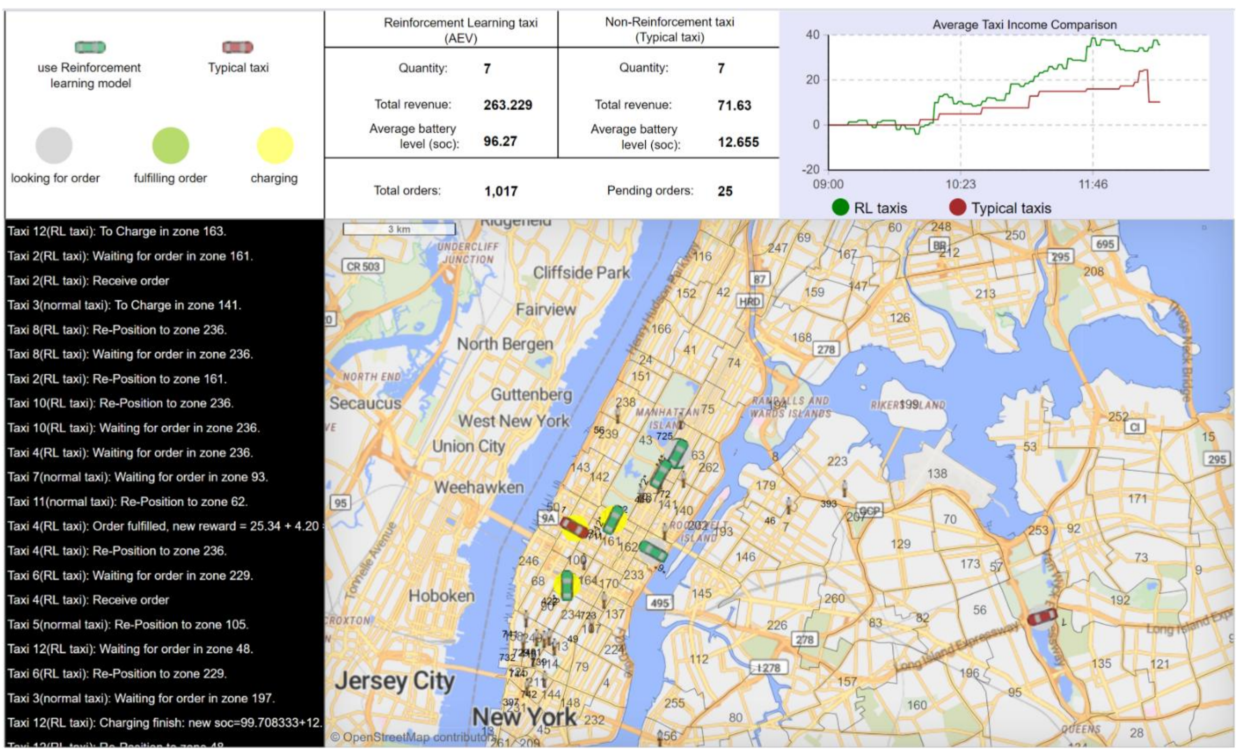

Figure 2: the output interface of MoD-DT with New York City Map and Taxi Data

3. Case Study

This case study demonstrates the implementation of the MoD-DT architecture in a ride-hailing setting and the integration of the DT with a RL model for generating idle vehicle repositioning strategies. The RL model makes repositioning decisions for an EV taxi driver based on their internal state including their current location, time of the day, and state of the charge (SOC). The RL model is trained using the New York City historic taxi dataset (https://databank.illinois.edu/datasets/IDB-9610843). In this dataset, New York city is divided into 263 districts. Taxi transaction records showing pick-up and drop-off districts and times are provided. Our research question here is to know what the best repositioning strategy is if current taxi fleets are replaced by EV fleets. Since the existing data set does not reflect EV fleets, we would generate synthetic transaction data with EVs and charging stations based on the rider demand represented in the existing dataset. The policy of the driver agent is to maximize long term (one day) income by optimizing their repositioning decisions, meaning that they reposition themselves to the positions with high likelihood to be matched with high-dollar-value orders. Figure 2 shows a small case study for ride-hailing EV repositioning. We deployed a group of 7 Green EV taxis and a group of 7 Red taxis. Green taxis use the RL model to make repositioning decisions, while red taxis cruise randomly after dropping off a customer in the hope of being matched to a ride request along their cruising route. Ride requests are dynamically generated across the city to reflect real-world demand patterns. The simulation runs for a full day in simulation time, during which key performance indicators are continuously recorded and displayed in the information windows. As shown in Figure 2, Green taxis, which use RL for repositioning, outperform red taxis in terms of average income. The purpose of this small case study is to demonstrate the function of the digital twin. We will report on larger scale digital twin simulations for MoD applications in the near future.

References

[1] N. J. Yuan, Y. Zheng, L. Zhang, and X. Xie, “T-Finder: A Recommender System for Finding Passengers and Vacant Taxis,” *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 10, pp. 2390–2403, Oct. 2013, doi: 10.1109/TKDE.2012.153.

[2] M. K. Chen and M. Sheldon, “Dynamic Pricing in a Labor Market: Surge Pricing and Flexible Work on the Uber Platform”.

[3] S. He and K. G. Shin, “Spatio-temporal Adaptive Pricing for Balancing Mobility-on-Demand Networks,” ACM Trans. Intell. Syst. Technol., vol. 10, no. 4, pp. 1–28, Jul. 2019, doi: 10.1145/3331450.

[4] A. Rosenblat, “Algorithmic Labor and Information Asymmetries: A Case Study of Uber’s Drivers,” 2016.

[5] M. Schwenzer, M. Ay, T. Bergs, and D. Abel, “Review on model predictive control: an engineering perspective,” Int J Adv Manuf Technol, vol. 117, no. 5–6, pp. 1327–1349, Nov. 2021, doi: 10.1007/s00170-021-07682-3.

[6] R. Iglesias, F. Rossi, K. Wang, D. Hallac, J. Leskovec, and M. Pavone, “Data-Driven Model Predictive Control of Autonomous Mobility-on-Demand Systems,” in 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD: IEEE, May 2018, pp. 1–7. doi: 10.1109/ICRA.2018.8460966.

[7] J. Gao, X. Li, C. Wang, and X. Huang, “BM-DDPG: An Integrated Dispatching Framework for Ride-Hailing Systems,” IEEE Trans. Intell. Transport. Syst., vol. 23, no. 8, pp. 11666–11676, Aug. 2022, doi: 10.1109/TITS.2021.3106243.

[8] L. Liu, Z. Qiu, G. Li, Q. Wang, W. Ouyang, and L. Lin, “Contextualized Spatial–Temporal Network for Taxi Origin-Destination Demand Prediction,” IEEE Trans. Intell. Transport. Syst., vol. 20, no. 10, pp. 3875–3887, Oct. 2019, doi: 10.1109/TITS.2019.2915525.

[9] M. Hyland and H. S. Mahmassani, “Dynamic autonomous vehicle fleet operations: Optimization-based strategies to assign AVs to immediate traveler demand requests,” Transportation Research Part C: Emerging Technologies, vol. 92, pp. 278–297, Jul. 2018, doi: 10.1016/j.trc.2018.05.003.

[10] P. Afèche, Z. Liu, and C. Maglaras, “Ride-Hailing Networks with Strategic Drivers: The Impact of Platform Control Capabilities on Performance”.

[11] Y. Jiao et al., “Real-world Ride-hailing Vehicle Repositioning using Deep Reinforcement Learning,” Jul. 08, 2021, arXiv: arXiv:2103.04555. doi: 10.48550/arXiv.2103.04555.

[12] J. Holler et al., “Deep Reinforcement Learning for Multi-driver Vehicle Dispatching and Repositioning Problem,” in 2019 IEEE International Conference on Data Mining (ICDM), Beijing, China: IEEE, Nov. 2019, pp. 1090–1095. doi: 10.1109/ICDM.2019.00129.

[13] C. Liu, C.-X. Chen, and C. Chen, “META: A City-Wide Taxi Repositioning Framework Based on Multi-Agent Reinforcement Learning,” IEEE Trans. Intell. Transport. Syst., vol. 23, no. 8, pp. 13890–13895, Aug. 2022, doi: 10.1109/TITS.2021.3096226.