Atomic Proximal Policy Optimization for Electric Robo-Taxi Dispatch and Charger Allocation

Jim Dai[a], Manxi Wu[b], Zhanhao Zhang[a,*]

[a] Operations Research and Information Engineering, Cornell University
[b] Department of Civil and Environmental Engineering, University of California, Berkeley
[*] Corresponding author

Extended abstract submitted for presentation at the 12th Triennial Symposium on Transportation Analysis conference (TRISTAN XII)
June 22-27, 2025, Okinawa, Japan

Keywords: Deep Reinforcement Learning, Electric Vehicles, Ride-Hailing Services.

1 Introduction

Companies like Waymo and Cruise are deploying electric robo-taxi services across the U.S., but these vehicles face operational challenges due to their relatively short range, driven by the high battery consumption needed for sensing and computation. Efficiently dispatching, repositioning, and charging these fleets in a dynamic, stochastic environment is a complex problem. In this article, we model robo-taxi fleet operations as a Markov Decision Process (MDP) and propose a deep reinforcement learning algorithm called Atomic-PPO to compute the optimal policy. Atomic-PPO builds on the classical PPO algorithm (Schulman *et al.*, 2017) but introduces a novel decomposition of the fleet routing policy by assigning atomic actions sequentially to individual vehicles. This approach reduces the action space from being exponential in fleet size to being a constant, significantly lowers the complexity of policy training.

We evaluate the effectiveness of Atomic-PPO using NYC taxi data and show that our method achieves a total reward that is a high percentage of the provable upper bound, derived from the system's fluid limit analysis (Theorem 1). Additionally, we provide insights into how system performance depends on factors such as charging speed, vehicle range, and charger allocation. Our model and results extend the reinforcement learning literature on non-EV ride-hailing dispatching (Feng *et al.*, 2021, Tang *et al.*, 2019) and matching (Azagirre *et al.*, 2024), and offer a new method for training RL algorithms in EV ride-hailing dispatching (Turan *et al.*, 2020, Kullman *et al.*, 2022, Luke *et al.*, 2021) that achieves fast training on a large problem scale.

2 Model

We consider a transportation network with V service regions. A fleet of N electric robo-taxi vehicles with battery size B are operated by a central planner to serve customer trip requests. For each pair of $(u, v) \in V \times V$, we assume that the battery consumption for traveling from u to v is a constant $b_{uv} \in \mathbb{R}_{\geq 0}$. A set of chargers with different per unit-time charging rates $\delta \in \Delta$ are installed in the network. The minimum charging time is J time steps. We model the operations of a ride-hailing system as a discrete-time MDP with finite time horizon T. At each time step $t = 1, \ldots, T$, the number of trip requests between each u-v pair follows a Poisson distribution with mean λ_{uv}^t, and trip duration being a constant τ_{uv}^t. A vehicle must be assigned to a rider within $L_c \geq 0$ time steps, and the rider will wait at most $L_p \geq 0$ time steps for the assigned vehicle to arrive at their origin. Otherwise, the rider will leave the system.

A vehicle is associated with type (v, η, b) if it is traveling to or charged at region v, with remaining time η, and battery level b upon finishing the task. We note that a vehicle may be

assigned to pick up a new rider before completing its current trip or charging period as long as $\eta \leq L_p$. Let $\mathcal{C}$ denote the set of all vehicle types. A trip order is associated with the type (u, v, ξ) if it originates from u, heads to v, and has been waiting for vehicle assignment in the system for ξ time steps. We use $\mathcal{O}$ to denote the set of all trip types. A charger is associated with type (v, δ, j) if it is located in region v with rate δ and is j time steps away from being available. We use $\mathcal{W}$ to denote the set of all charger types.

State. The state vector $s^t \in \mathcal{S}$ records the number of trip orders of each type, the number of vehicles of each type, and the number of chargers of each type at t.

Action. At each t, a central planner selects a fleet routing action $a^t := (f_c^t, e_c^t, q_c^t, p_c^t)_{c\in\mathcal{C}}$. The term $f_c^t := (f_{c,o}^t \in \mathbb{N})_{o\in\mathcal{O}}$ represents the number of vehicles of type $c := (v, \eta, b)$ assigned to each trip type o. Similarly, $e_c^t := (e_{c,v}^t \in \mathbb{N})_{v\in V}$ represents the number assigned to reposition to destination v, and $q_c^t := (q_{c,\delta}^t \in \mathbb{N})_{\delta\in\Delta}$ the number to charge at stations with rate δ. Finally, $p_c^t \in \mathbb{N}$ represents the number assigned to continue their current action (referred as the passing action). We denote the set of fleet routing actions that are feasible given state s as $\mathcal{A}^s$. Given s^t and a^t, the state transition to s^{t+1} at time step $t+1$ includes the change of fleet state (trip fulfillment, repositioning, and charging), order state (trip fulfillment and new trip arrival) and charger state update. We omit the expression of state transition due to space limit.

Policy. The central planner determines a fleet routing policy $\pi := \{\pi^t : \mathcal{S} \to \Delta(\mathcal{A})\}_{t\in[T]}$ that maps the system state to a distribution of feasible fleet routing actions at each time step t, where $\pi^t(a|s)$ is the probability of routing the fleet according to a given state s at time t.

Reward. At each time step t, the reward of fulfilling a trip request between u and v is $r_{f,uv}^t \in \mathbb{R}_{\geq 0}$, and the reward of repositioning between u and v is $r_{e,uv}^t \in \mathbb{R}_{\leq 0}$. Additionally, the reward of charging a vehicle at time t is $r_{q,\delta}^t \in \mathbb{R}_{\leq 0}$. We can compute the total reward at t given the fleet routing action a^t, denoted as $r^t(a^t)$. Given any initial state s, the goal of the central planner is to compute the optimal policy π^* that maximize the expected total reward $R(\pi|s) := \mathbb{E}_\pi\left[\sum_{t=1}^T r^t(a^t)|s\right]$. We study the fluid limit of the MDP and construct a fluid upper bound of the optimal total reward.

Theorem 1 (Fluid upper bound (informal)) We construct a fluid-based linear program with optimal value $\bar{R}$ that is an upper bound of $R(\pi^*|s)$ for all initial state $s \in \mathcal{S}$.

3 Atomic Proximal Policy Optimization

3.1 Action space reduction. One critical challenge of computing the optimal fleet routing policy π^* is that the dimension of fleet routing action a and $|\mathcal{A}|$ scales exponentially with the fleet size N and the number of vehicle types $|\mathcal{C}|$. To address this challenge, we propose an *atomic action policy*: Instead of determining the entire fleet routing policy, we assign atomic action $\tilde{\mathcal{A}} := \{(\hat{\mathrm{f}}_o)_{o\in\mathcal{O}}, (\hat{\mathrm{e}}_v)_{v\in V}, (\hat{\mathrm{q}}_\delta)_{\delta\in\Delta}, \tilde{\mathrm{p}}\}$ to each vehicle sequentially, where $\hat{\mathrm{f}}_o$ is to fulfill a trip of type o, $\hat{\mathrm{e}}_v$ is to reposition to region v, $\hat{\mathrm{q}}_\delta$ is to charge with rate δ, and $\tilde{\mathrm{p}}$ is to pass.

We now present the procedure of atomic action assignment. In each time step t, vehicles are arbitrarily indexed from 1 to N, and are sequentially selected. For a selected vehicle n, the atomic policy $\tilde{\pi}^t : \mathcal{S}\times\mathcal{C} \to \Delta(\tilde{\mathcal{A}})$ maps from the tuple of system state s_n^t before n-th assignment and the selected vehicle's type c_n to a distribution of atomic actions. The system state s_n^t transitions after every single vehicle assignment with $s_1^t = s^t$, and s_N^t transitions to s^{t+1} after assigning the last vehicle and trip arrival at $t+1$ is realized. For any atomic action $\tilde{a}$ and vehicle type c, the generated reward $r^t(\tilde{a}, c)$ is the same as that defined in Sec. 2, and the total reward of each time step t is the sum of all rewards generated from each atomic action assignment in t. Our atomic action policy can be viewed as a reduction of the original fleet routing policy in that any realized sequence of atomic actions corresponds to a feasible fleet routing action with the same reward of the time step. The atomic action policy significantly reduces the dimension of the policy function since $\tilde{\mathcal{A}}$ does not scale with the fleet size or the number of vehicle types.

3.2 Atomic PPO with state reduction. We develop the Atomic proximal policy optimization (Atomic-PPO as in Algorithm 1) to compute the optimal atomic policy $\tilde{\pi}^*$. This algorithm builds on our atomic policy reduction and the classical PPO method. In our Atomic-PPO, we use neural networks to approximate the value function $\tilde{V}_\psi^t : \tilde{\mathcal{S}} \to \mathbb{R}$ (resp. policy function $\tilde{\pi}_\theta^t : \tilde{\mathcal{S}} \times \mathcal{C} \to \Delta(\tilde{A})$), where ψ (resp. θ) denotes the network parameters. Here, we further reduce the state space by clustering battery levels into three categories: low, medium, and high, with flexibility to refine this granularity if computational resources allow. Trip orders are aggregated by recording only the number of requests originating from or arriving at each region, instead of tracking origin-destination pairs, further reducing the state space.

Algorithm 1: The Atomic-PPO Algorithm

Inputs: Number of policy iterations J, number of episodes K, initial policy $\tilde{\pi}_{\theta_0}$.

for *policy iteration* $j = 1, \ldots, J$ **do**

Run policy $\tilde{\pi}_{\theta_{j-1}}$ for K episodes each of which is a single day operations.

Store trajectory of system state $s_n^{t,(k)}$, vehicle type $c_n^{t,(k)}$, atomic action $\tilde{a}_n^{t,(k)}$, and reward $r^t(\tilde{a}_n^{t,(k)}, c_n^{t,(k)})$ for each assignment n, time step $t \in T$ and episode $k \in K$.

Compute empirical estimate of value function $\hat{V}_n^{t,(k)}(\tilde{s}_n^{t,(k)})$ as the accumulated reward of realized trajectory starting from n-th assignment at time step t in k-th episode. Update value network by minimizing $\sum_{k,t,n}(\tilde{V}_\psi^t(\tilde{s}_n^{t,(k)}) - \hat{V}_n^{t,(k)}(\tilde{s}_n^{t,(k)}))^2$.

Estimate advantage functions by

$$\hat{A}_{\theta_{j-1}}(\tilde{s}_n^{t,(k)}, \tilde{a}_n^{t,(k)}, c_n^{t,(k)}) := \begin{cases} r^t(\tilde{a}_n^{t,(k)}, c_n^{t,(k)}) + \tilde{V}_\psi(\tilde{s}_{n+1}^{t,(k)}) - \tilde{V}_\psi(\tilde{s}_n^{t,(k)}), & \text{If } n < N, \\ r^t(\tilde{a}_n^{t,(k)}, c_n^{t,(k)}) + \tilde{V}_\psi(\tilde{s}_1^{t+1,(k)}) - \tilde{V}_\psi(\tilde{s}_n^{t,(k)}), & \text{If } n = N. \end{cases}$$

Obtain the updated policy network $\tilde{\pi}_{\theta_j}$ by maximizing surrogate objective function

$$\hat{L}(\theta_j, \theta_{j-1}) := \frac{1}{K} \sum_{k,t,n} \min \left(\frac{\tilde{\pi}_{\theta_j}(\tilde{a}_n^{t,(k)} | \tilde{s}_n^{t,(k)}, c_n^{t,(k)})}{\tilde{\pi}_{\theta_{j-1}}(\tilde{a}_n^{t,(k)} | \tilde{s}_n^{t,(k)}, c_n^{t,(k)})} \hat{A}_{\theta_{j-1}}(\tilde{s}_n^{t,(k)}, \tilde{a}_n^{t,(k)}, c_n^{t,(k)}), \right.$$
$$\left. \text{clip}\left(\frac{\tilde{\pi}_{\theta_j}(\tilde{a}_n^{t,(k)} | \tilde{s}_n^{t,(k)}, c_n^{t,(k)})}{\tilde{\pi}_{\theta_{j-1}}(\tilde{a}_n^{t,(k)} | \tilde{s}_n^{t,(k)}, c_n^{t,(k)})}, 1-\epsilon, 1+\epsilon \right) \hat{A}_{\theta_{j-1}}(\tilde{s}_n^{t,(k)}, \tilde{a}_n^{t,(k)}, c_n^{t,(k)}) \right).$$

end

return *policy* $\tilde{\pi}_{\theta_J}$

4 Numerical experiments

We conduct numerical experiments of routing electric robo-taxi in New York City (NYC) Manhattan area. Using the NYC ride-hailing dataset in 2022, we calibrate the trip demand distribution of every 5 min time interval (time step) from 0:00 to 24:00 Mondays to Thursdays. We cluster all taxi zones in Manhattan into 10 regions, and the trip demand distribution is aggregated for each pair of origin and destination regions. We consider fleet size $N = 300$ and scale down the trip demand based on the actual fleet size during rush hours. Our baseline setting sets the full vehicle battery range as 130 miles following the Nissan Leaf E 2023 model (215 miles if driven by humans) and the fact that robo-taxi on average spends 40% of energy on sensing, computation and communication. Our baseline setting assumes that each region has abundant number of fast chargers (75kW). Furthermore, our simulation incorporates the nonlinear charging rate, time-varying charging and repositioning cost and trip reward. With 30 CPUs, it takes less than 20 minutes to finish a policy iteration of Atomic-PPO and the training for each case can be completed within 3 hours. Using Microsoft Azure, the training of one case costs \$15.

We find that our atomic PPO algorithm achieves total reward of \$390K, which is 91% of the fluid upper bound $\bar{R} = \$428K$ (recall Theorem 1). We also compare our Atomic-PPO against

	Revenue/$\bar{R}$	Revenue
Atomic-PPO	91%	\$390K
Power of d	71%	\$310K
Fluid policy	43%	\$185K

Table 1 – *Revenue comparison.*

	Setup	Revenue
Benchmark	75 kW & 130 miles	\$390K
Slow Chargers	15 kW & 130 miles	\$335K
Double Range	75 kW & 260 miles	\$390K

Table 2 – *Change of charging rates and range.*

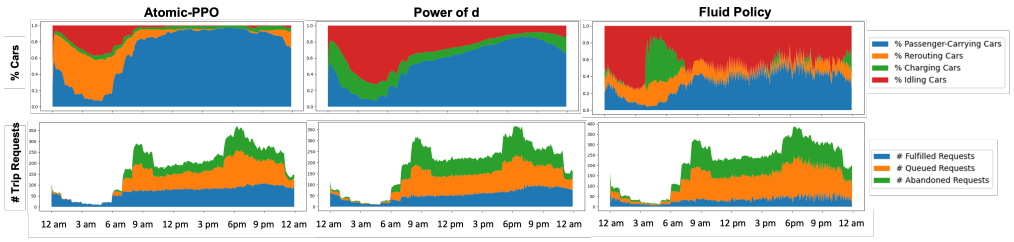

Figure 1 – *Policy evaluation. (First row) Fleet status; (Second row) Trip fulfillment status.*

two benchmark algorithms: (i) the power-of-d dispatching policy that fulfills a trip with a vehicle with highest battery level among d closest vehicles within dispatch range (Varma *et al.*, 2023), and (ii) the fluid policy derived from randomized rounding of optimal solution of the fluid-based LP. Table 1 shows that the performance of Atomic-PPO beats the benchmark algorithms in terms of total revenue by a large margin. Figure 1 demonstrates that Atomic-PPO has a uniformly higher percentage of fleet used for trip fulfilling, and higher trip fulfillment rate.

Apart from the baseline setting, we conduct experiments with other parameter settings and find that (i) fast chargers can effectively increases revenue, while doubling the vehicle range has negligible impact (Table. 2), and (ii) deploying a small number of chargers according to ridership patterns (which concentrates in midtown Manhattan) can achieve comparable revenue as that with abundant chargers, while uniform charger deployment is inefficient.

Allocation	Uniform (# chargers)				Concentrated (15 chargers)			
	10	20	30	40	Midtown	Lower Manh.	Upper Manh.	Abundant
Revenue (\$)	250K	375K	390K	390K	380K	225K	335K	390K

Table 3 – *Impact of charger distribution in Manhattan on revenue.*

References

Azagirre, Xabi, Balwally, Akshay, Candeli, Guillaume, Chamandy, Nicholas, Han, Benjamin, King, Alona, Lee, Hyungjun, Loncaric, Martin, Martin, Sébastien, Narasiman, Vijay, *et al.* 2024. A better match for drivers and riders: Reinforcement learning at lyft. *INFORMS Journal on Applied Analytics*, **54**(1), 71–83.

Feng, Jiekun, Gluzman, Mark, & Dai, Jim G. 2021. Scalable deep reinforcement learning for ride-hailing. *Pages 3743–3748 of: 2021 American Control Conference (ACC)*. IEEE.

Kullman, Nicholas D., Cousineau, Martin, Goodson, Justin C., & Mendoza, Jorge E. 2022. Dynamic Ride-Hailing with Electric Vehicles. *Transportation Science*, **56**(3), 775794.

Luke, Justin, Salazar, Mauro, Rajagopal, Ram, & Pavone, Marco. 2021. Joint optimization of autonomous electric vehicle fleet operations and charging station siting. *Pages 3340–3347 of: 2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. IEEE.

Schulman, John, Wolski, Filip, Dhariwal, Prafulla, Radford, Alec, & Klimov, Oleg. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Tang, Xiaocheng, Qin, Zhiwei, Zhang, Fan, Wang, Zhaodong, Xu, Zhe, Ma, Yintai, Zhu, Hongtu, & Ye, Jieping. 2019. A deep value-network based approach for multi-driver order dispatching. *Pages 1780–1790 of: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*.

Turan, Berkay, Pedarsani, Ramtin, & Alizadeh, Mahnoosh. 2020. Dynamic pricing and fleet management for electric autonomous mobility on demand systems. *Transportation Research Part C: Emerging Technologies*, **121**, 102829.

Varma, Sushil Mahavir, Castro, Francisco, & Maguluri, Siva Theja. 2023. Electric vehicle fleet and charging infrastructure planning. *arXiv preprint arXiv:2306.10178*.