Benchmarking Reinforcement Learning for Network-level Coordination of Autonomous Mobility-on-Demand Systems Across Scales

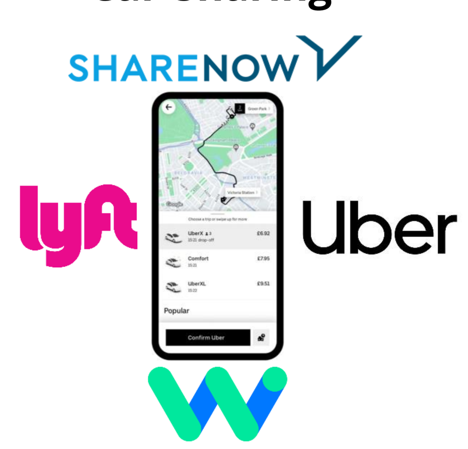
Luigi Tresca[1,2], Daniele Gammelli[2], James Harrison[3], Gioele Zardini[2,4], Marco Pavone[2]
[1]Politecnico di Torino, [2]Stanford University , [3]Google DeepMind, [4]MIT

Autonomous Mobility-on-Demand (AMoD)

As urbanization intensifies, relying on **private cars** for private mobility becomes **increasingly unsustainable**, necessitating the exploration of alternative transit solutions.
Technological advances in the field of **autonomous driving** together with **mobility-on-demand systems** offer a potential solution by enabling operators to **coordinate vehicles in an automated and centralized manner**

Challenge: AMoD systems potentially entail controlling **thousands** of AVs in complex and congested networks

Autonomy + **Car-Sharing**



Coordination Across Scales

Although previous approaches cover a wide range of algorithms, there lacks a discussion on

- how to **combine** the **benefits** of **learning-based** and **optimization-based** methods
- Define **neural network architectures** able to exploit the **graph structure** present in **urban transportation networks**

Objectives:

- Propose a novel hierarchical policy framework that **leverages the strength of direct optimization and graph network-based RL**
- Show that this approach is **highly performant, scalable and robust** to **changes in operating conditions** and **network topologies**
- Show that the desired features are still valid **across simulator of different fidelity**

Graph-Reinforcement Learning

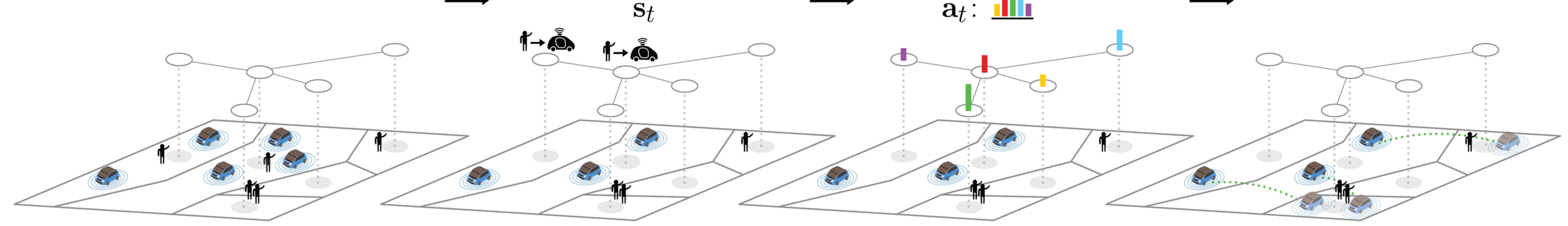


Fig 1. An illustration of the three-step framework determining the proposed AMoD control strategy. Given the current distribution of idle vehicles (cars) and user transportation requests (stick figures), the control strategy is defined by: (1) dispatching idle vehicles to specific trip requests by solving a matching problem (thus, characterizing the current state $\mathbf{s}_t$ of the system), (2) computing an action $\mathbf{a}_t$ (i.e. the desired distribution of idle vehicles) using some policy π_θ , and (3) translate into actionable rebalancing trips such that overall rebalancing cost is minimized.

1. Matching

We solve the following matching problem to derive **passenger flows** $\{x_{ij}^t\}_{i,j\in\mathcal{V}}$

$$\max_{\{x_{ij}^t\}_{i,j\in\mathcal{V}}} \sum_{i,j\in\mathcal{V}} x_{ij}^t(p_{ij}^t - c_{ij}^t)$$
$$\text{s.t.} \quad 0 \le x_{ij}^t \le d_{ij}^t, \; i,j\in\mathcal{V},$$

$\{p_{ij}^t\}_{i,j\in\mathcal{V}}$: price

$\{c_{ij}^t\}_{i,j\in\mathcal{V}}$: cost

$\{d_{ij}^t\}_{i,j\in\mathcal{V}}$: demand

2. GNN-RL $\mathbf{a}_{\text{reb}}^t$:

A learned behavior policy $\pi_\theta(\mathbf{a}_t|\mathbf{s}_t)$ is used to determine the desired idle vehicle distribution $\mathbf{a}_{\text{reb}}^t = \{a_{\text{reb},i}^t\}_{i\in\mathcal{V}}$

3. Minimal Rebalancing Cost

The third step entails rebalancing, wherein a minimal rebalancing-cost problem is solved to derive **rebalancing flows** $\{y_{ij}^t\}_{i,j\in\mathcal{V}}$

$$\min_{\{y_{ij}^t\}_{(i,j)\in\mathcal{E}}\in\mathbb{Z}_+^{|\mathcal{E}|}} \sum_{(i,j)\in\mathcal{E}} c_{ij}y_{ij}^t$$
$$\text{s.t.} \quad \sum_{j\neq i}(y_{ji}^t - y_{ij}^t) + m_i^t \ge \hat{m}_i^t, \; i\in\mathcal{V},$$
$$\sum_{j\neq i} y_{ij}^t \le m_i^t, \; i\in\mathcal{V},$$

$\{m_i^t\}_{i\in\mathcal{V}}$: idle vehicles

$\{\hat{m}_i^t\}_{i\in\mathcal{V}}$: desired idle vehicles

Advantages

- **Decomposes problem in time**, since we can combine the long-term capabilities of RL methods with the performance guarantees of optimization methods
- **Reduction of the action space** from N_v^2 to N_v since the learned policy **defines an action at each node**

Results

TABLE I
SYSTEM PERFORMANCE ON NEW YORK MACROSCOPIC SIMULATION

	Reward (%Dev. MPC-oracle)	Served Demand	Rebalancing Cost ($)
ED	30,746 (-13.4%)	8,770	7,990
RL (ours)	33,886 (**-4.3%**)	8,772	5,038
MPC-oracle	35,356 (0%)	8,968	4,296
RL-0Shot	33,397 (**-5,7%**)	8,628	4,743

TABLE II
SYSTEM PERFORMANCE ON CHENGDU MACROSCOPIC SIMULATION

	Reward (%Dev. MPC-oracle)	Served Demand	Rebalancing Cost ($)
ED	12,538 (-26,8%)	41,189	3,397
RL (ours)	15,167 (**-9,8%**)	40,578	1,063
MPC-oracle	16,702 (0.0%)	44,662	1,162
RL-0Shot	14,791 (**-12,3%**)	40,646	1,467

TABLE III
SYSTEM PERFORMANCE ON LUXEMBOURG MESOSCOPIC SIMULATION

	Reward (%Dev. MPC-oracle)	Served Demand	Rebalancing Cost ($)
ED	20,17 (-20,1%)	100%	8,85
P1	29,48 (-22,7%)	79%	3,38
RL (ours)	24,25 (**-3,9%**)	100%	5,00
MPC-oracle	25,24 (0%)	95%	2,40

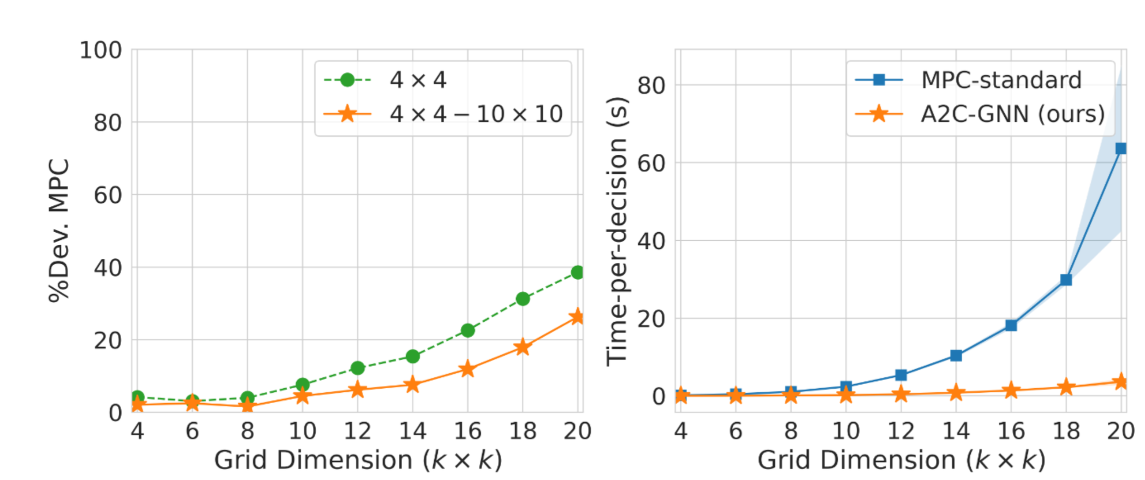
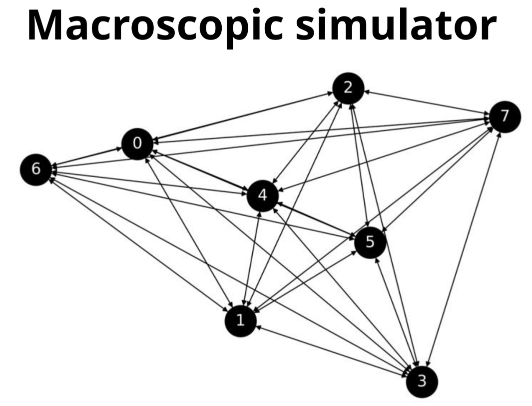
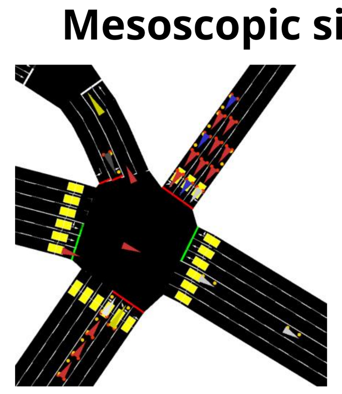


Fig. 2. Left: System performance (Percentage Deviation from MPC-standard) for agents trained either on a single granularity (4×4) or across granularities (4×4-10×10), Right: Comparison of computation times between A2C-GNN and MPC-standard.

Towards Open Sourcing a MobilityGym

Macroscopic simulator **Mesoscopic simulator**





SUMO

With this work we aim at **democratizing algorithmic development** for the AMoD system control problem

- releasing **benchmarks**, **simulators across different scales** and **calibrated scenarios** from real-world data
- **engaging diverse communities** by inviting contributions from a broad spectrum of fields
- **expand the portfolio of applications** that could benefit from the presented approach, including **large-scale network-based problems**.

SCAN ME

Conclusion

This work paves the way for future investigations evaluating the framework's potential using higher fidelity levels, including microscopic traffic simulators. Future efforts will focus on analysing generalizability across different fidelity levels and assessing performance versus computational trade-offs.