Markov Decision Processes

Modeling sequential decision problems

Cathy Wu

1.041/1.200 Transportation: Foundations and Methods

Readings

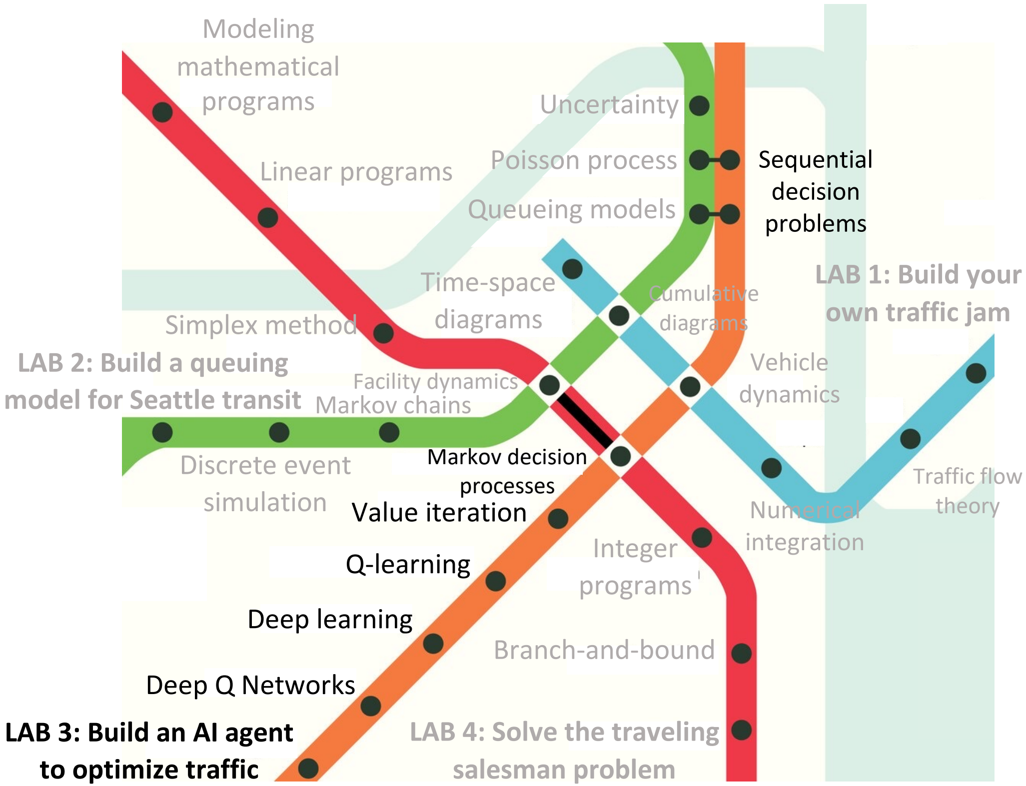
1. Morales, Miguel. Grokking deep reinforcement learning. 2020. Chapter 2: Mathematical Foundations of Reinforcement Learning. [URL]
2. (Optional) Morales, Miguel. Grokking deep reinforcement learning. 2020. Chapter 1: Introduction to Deep Reinforcement Learning. [URL]

Unit 3: Machine learning for traffic control

Unit 3

Optimizing

Multi-stage



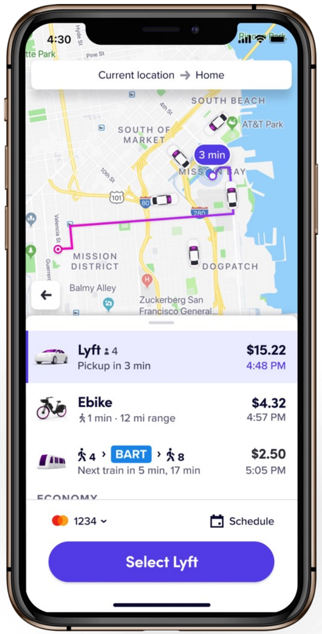
Outline

1. Reinforcement learning for transportation
2. Markov Decision Process (MDP)
3. The optimization problem
4. Emergency medical service (EMS) vehicle problem

Outline

1. **Reinforcement learning for transportation**
   a. For real-time decision making (online)
   b. For modeling system behavior (offline)
2. Markov Decision Process (MDP)
3. The optimization problem
4. Emergency medical service (EMS) vehicle problem

RL for real-time transportation decision making



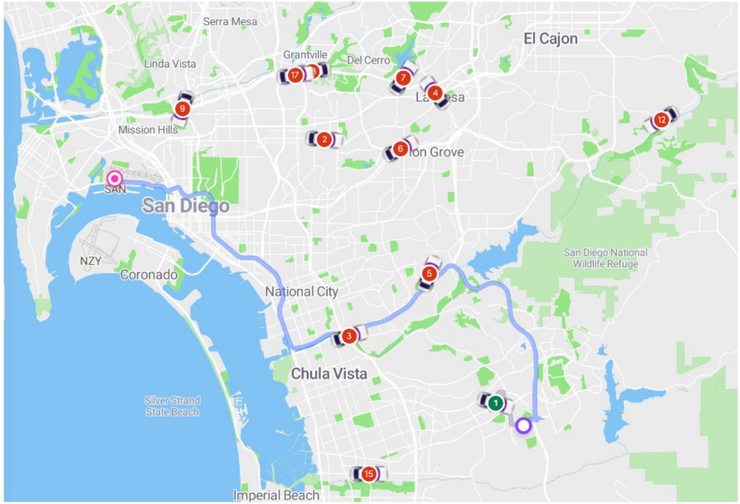
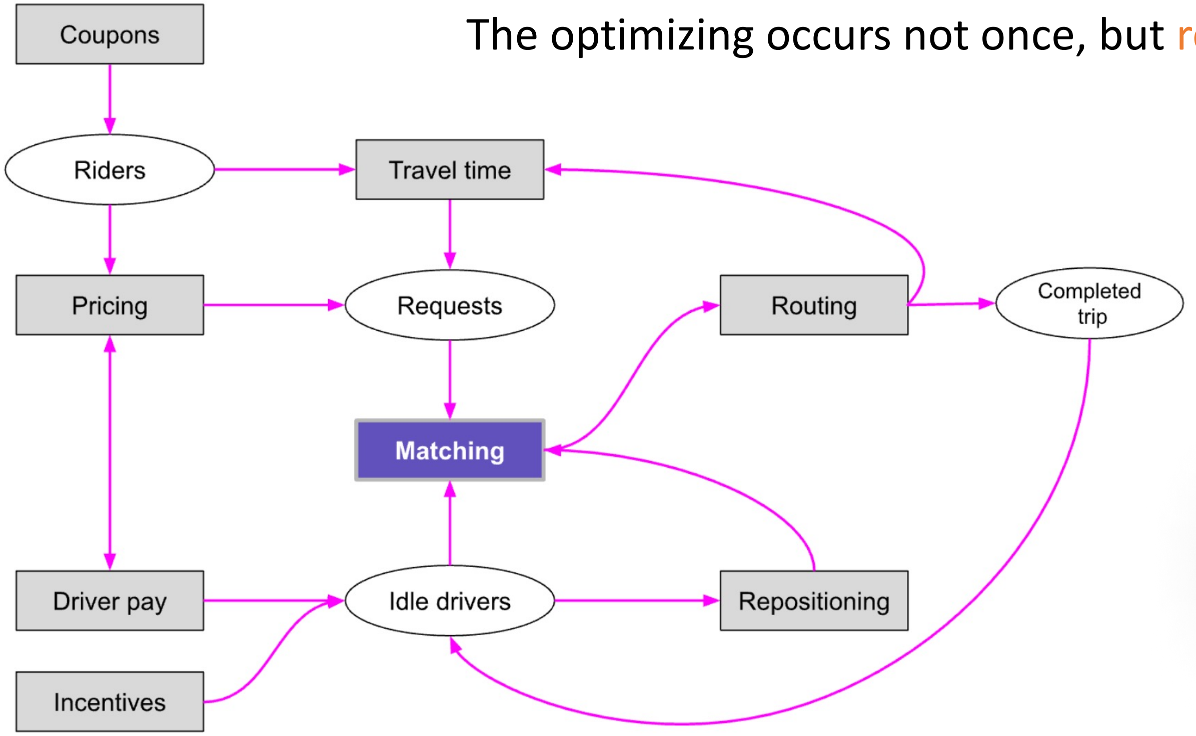


Figure 2: Multiple available drivers may be considered for a ride request (shown with purple pickup and pink destination icons), but differing pickup times will yield lower expected time-discounted rewards for more distant drivers because a distant driver requires more time to fulfill the ride and has a greater likelihood of cancellation. The closest driver (shown in green), often has the largest time-discounted reward for a given request.

B. Han, H. Lee, and S. Martin, “Real-Time Rideshare Driver Supply Values Using Online Reinforcement Learning,” in *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, in KDD ’22. Aug. 2022. doi: [10.1145/3534678.3539141](https://doi.org/10.1145/3534678.3539141).
X. Azagirre *et al.*, “A Better Match for Drivers and Riders: Reinforcement Learning at Lyft.” INFORMS Journal on Applied Analytics, 2023. doi: [10.48550/arXiv.2310.13810](https://doi.org/10.48550/arXiv.2310.13810).

Real-time ridesharing optimization modules

The optimizing occurs not once, but repeatedly over time!



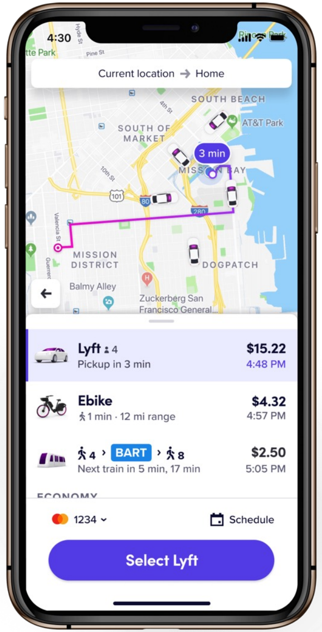


Figure 1. The Graphic Illustrates the Interconnected Optimization Modules that Power a Ridesharing Network

X. Azagirre *et al.*, “A Better Match for Drivers and Riders: Reinforcement Learning at Lyft.” INFORMS Journal on Applied Analytics, 2023. doi: 10.48550/arXiv.2310.13810.

Real-time rideshare: Non-sequential vs sequential decision making

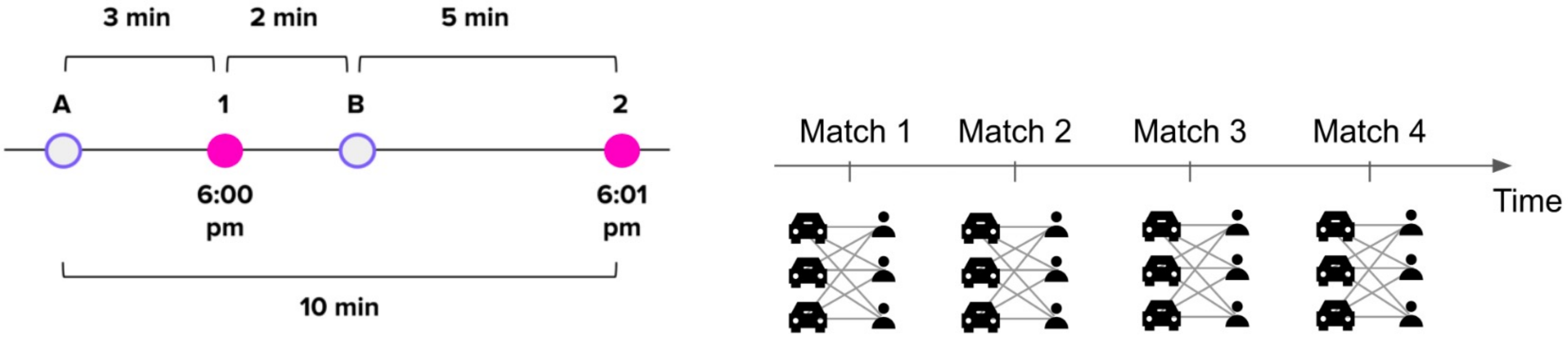


Figure 3. On the Left, the Graphic Illustrates a Stylized Example of a Ridesharing Matching Batch with Two Riders (1 and 2) and Two Available Drivers (A and B); on the Right, It Illustrates How Lyft Solves the Online Matching Problem as a Sequence of Batch Matching Decisions

X. Azagirre *et al.*, “A Better Match for Drivers and Riders: Reinforcement Learning at Lyft.” INFORMS Journal on Applied Analytics, 2023. doi: 10.48550/arXiv.2310.13810.

Real-time rideshare matching

Driver $i \in \{1, \dots, m\}$
Request $j \in \{1, \dots, n\}$
Assignment $a_{ij} \in \{0,1\}$
Immediate reward r_{ij}
Probability of cancellation p_{ij}

At each round (every 4 seconds), solve:

Bipartite Matching

$$\begin{aligned}
\operatorname*{argmax}_{a_{ij}} \quad & \sum_{i=1}^{m}\sum_{j=1}^{n} r_{ij}a_{ij} \\
\text{subject to} \quad & \sum_{i=1}^{m} a_{ij} \le 1, \quad j = 1, 2, 3, \dots, n \\
& \sum_{j=1}^{n} a_{ij} \le 1, \quad i = 1, 2, 3, \dots, m \\
& a_{ij} \in \{0, 1\}, \quad \forall (i, j)
\end{aligned}$$

Supply Value Dispatch

$$\begin{aligned}
\operatorname*{argmax}_{a_{ij}} \quad & \sum_{i=1}^{m}\sum_{j=1}^{n} \boxed{\Delta_{ij}}a_{ij} \\
\text{subject to} \quad & \sum_{i=1}^{m} a_{ij} \le 1, \quad j = 1, 2, 3, \dots, n \\
& \sum_{j=1}^{n} a_{ij} = 1, \quad i = 1, 2, 3, \dots, m \\
& a_{ij} \in \{0, 1\}, \quad \forall (i, j)
\end{aligned}$$

Instead, want the match to capture long-term value!

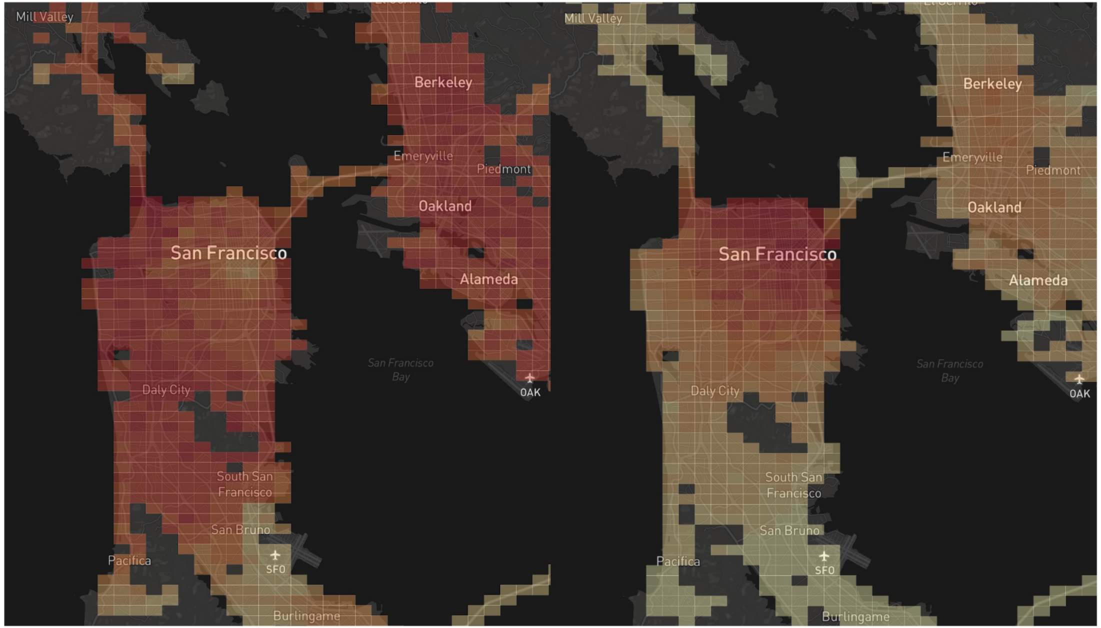
B. Han, H. Lee, and S. Martin, "Real-Time Rideshare Driver Supply Values Using Online Reinforcement Learning," in *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, in KDD '22. Aug. 2022. doi: [10.1145/3534678.3539141](https://doi.org/10.1145/3534678.3539141).

Sequential decision making

- More generally, sequential decision making is a suitable framework for modeling a problem when making decisions based on **immediate** information is expected to be suboptimal.

Real-time rideshare: Online Supply Value estimates

- What isn’t accounted for in immediate decision making



(a) Driver return estimates during weekday morning commute hour. **(b) Driver return estimates during weekend late night hours.**

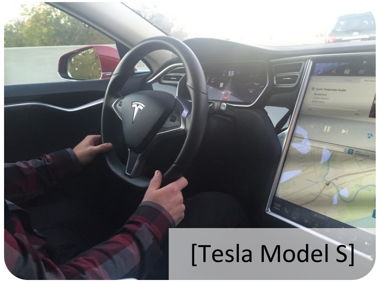
Figure 1: Online Supply Value estimates vary in space, time, and vehicle type (red is higher value).

- Reinforcement learning is about learning the long-term value of a decision

B. Han, H. Lee, and S. Martin, “Real-Time Rideshare Driver Supply Values Using Online Reinforcement Learning,” in *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, in KDD ’22. Aug. 2022. doi: 10.1145/3534678.3539141.

RL for modeling transportation system behavior

- Traffic flow smoothing with automated vehicles
- What if even one of these vehicles is not self-driving?
- Will we see benefits to the system before 100% adoption? (2050+)



Waymo

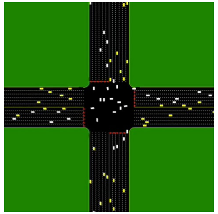
Wu, et al., "Flow: A Modular Learning Framework for Mixed Autonomy Traffic." IEEE T-RO, 2021. https://arxiv.org/abs/1710.05465

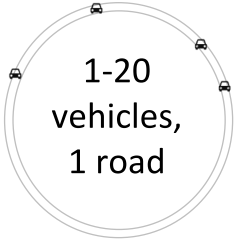
Sequential decision making

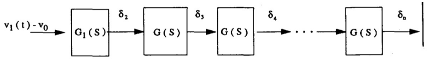
- More generally, sequential decision making is a suitable framework for modeling a problem when making decisions based on **immediate** information is expected to be suboptimal.

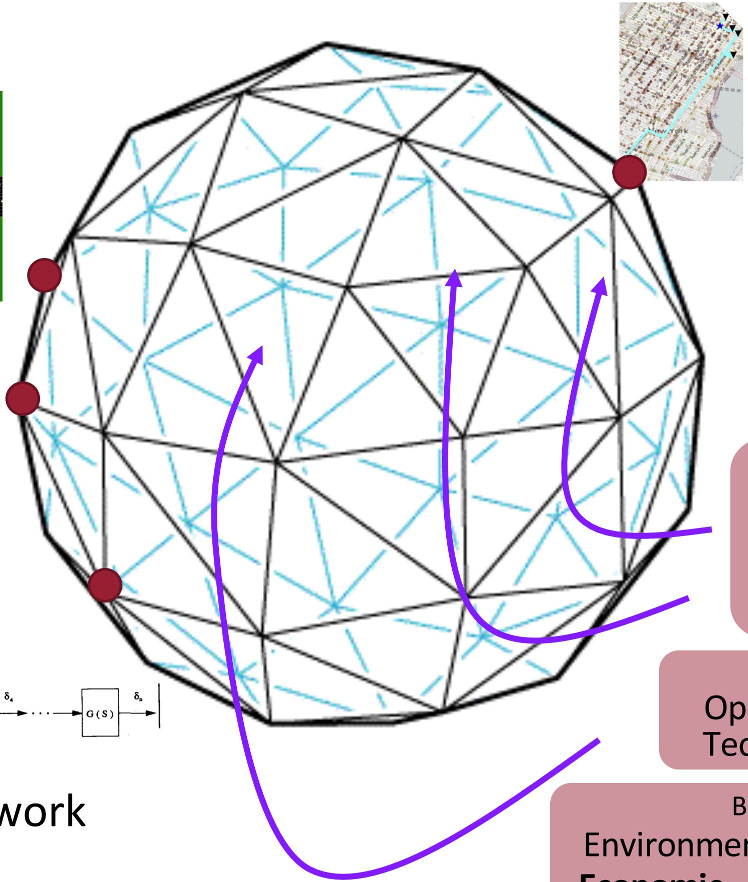
- For real-time decision making (online)
- For understanding system behavior (offline)

Autonomous mobility as a design space









- **Autonomy enables control and coordination of vehicles. To what end? How effectively? At what cost?**
- **A combinatorial problem space**
  - Multiple objectives
  - Diverse scenarios
  - Spectrum of autonomy technologies
- **Multi-agent interactions**
- **Evolving design specifications**
- **RL provides methods for rapid system analysis.**

NETWORK SCENARIO
Network topology
Adoption rate Scale
Levels of autonomy

TECHNOLOGY DESIGN
V2V, V2I communication
Operational design domain
Infrastructure support
Multi-modal integration

COSTS
Operation Risks Taxes
Technology Incentives

MARKET DESIGN
Private vs fleet
Shared rides
Multiple operators

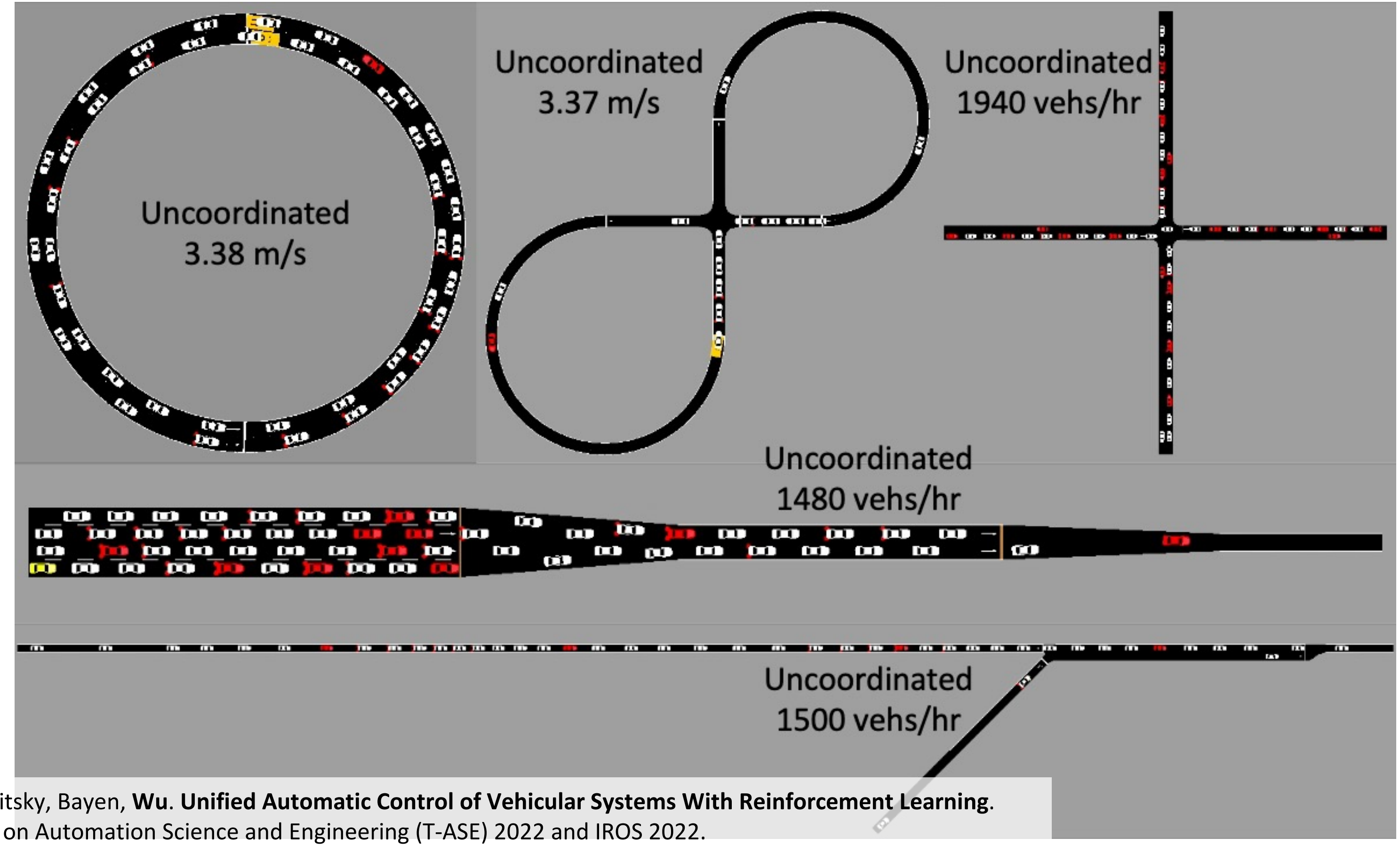
BENEFITS
Environmental Land use
Economic Safety Access

HUMAN FACTORS
Public trust
Behavior drift

● Existing work

RL + traffic LEGO blocks

5-30% CAVs → 13-120% improvement



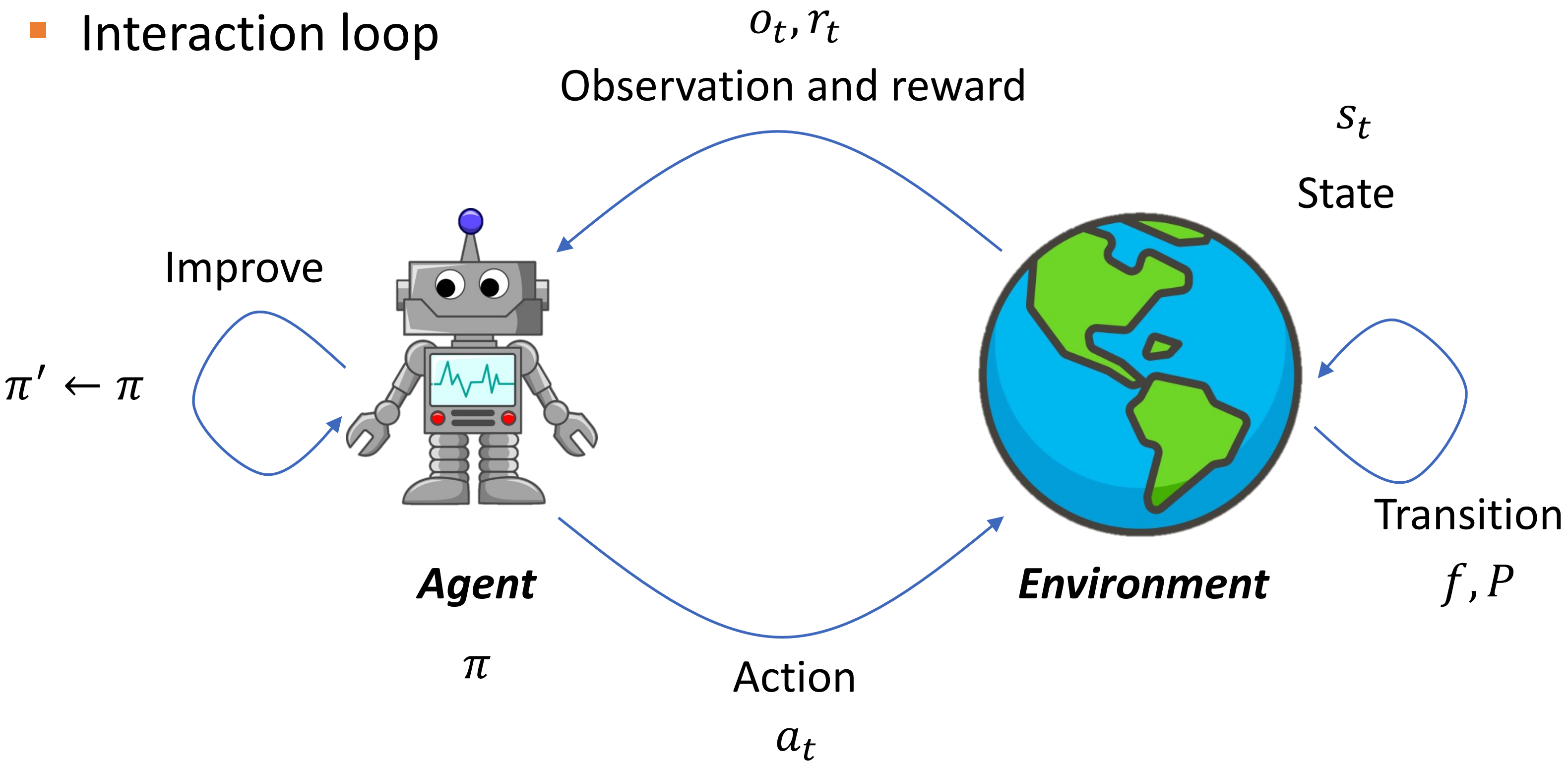
Yan, Kreidieh, Vinitsky, Bayen, **Wu**. **Unified Automatic Control of Vehicular Systems With Reinforcement Learning**.
IEEE Transactions on Automation Science and Engineering (T-ASE) 2022 and IROS 2022.

Outline

1. Reinforcement learning for transportation
2. **Markov Decision Process (MDP)**
   a. The interaction loop
   b. The modeling framework
   c. Real-time ridesharing (2022)
   d. Exploration vs exploitation
3. The optimization problem
4. Emergency medical service (EMS) vehicle problem

Introduce the characters*

- Interaction loop



Goal: maximize reward over time (returns, cumulative reward)

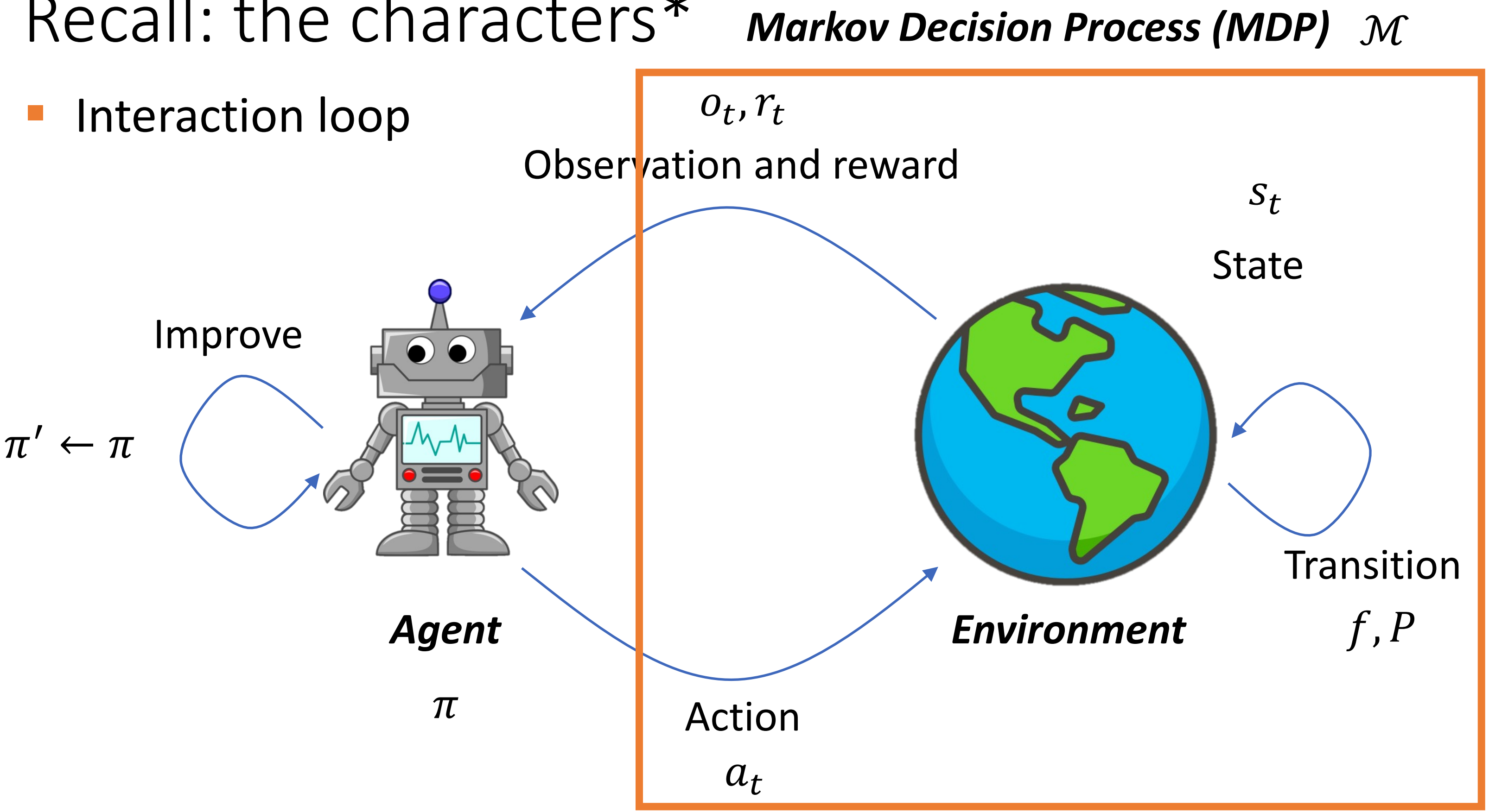
* pun intended

Outline

1. The main characters – the interaction loop
2. **Markov Decision Process (MDP)**
   a. The optimization problem
   b. Examples
   c. Assumptions
   d. Policy
3. Modeling sequential decision problems as MDPs
4. Emergency medical service vehicle problem

Recall: the characters*

- Interaction loop



Goal: maximize reward over time (returns, cumulative reward)

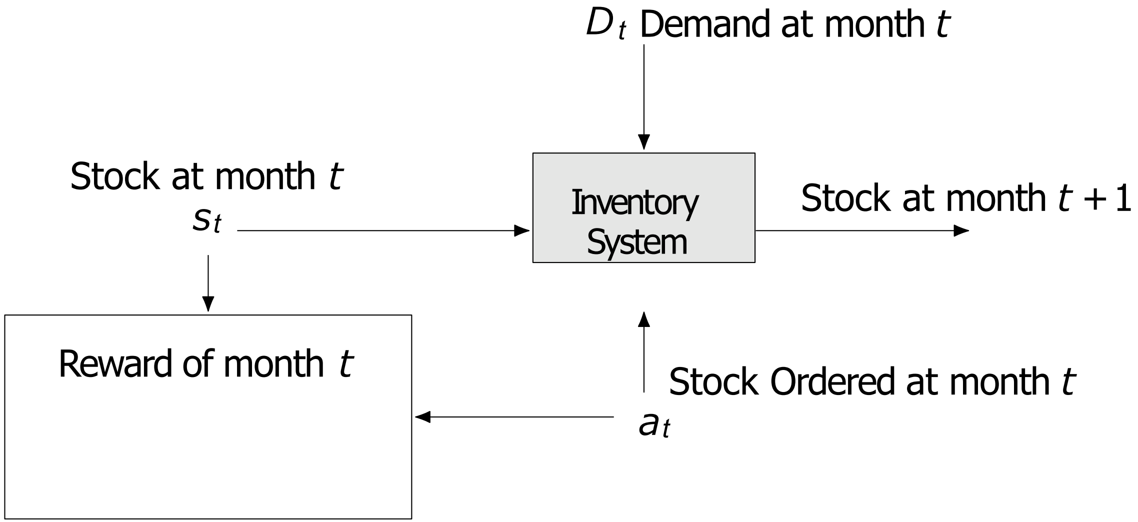
* pun intended

Assume for now: finite horizon problems, i.e. $T < \infty$

Used when: there is an intrinsic deadline to meet.

Later: infinite horizon

Example: The Amazing Goods Company Example



Example: The Amazing Goods Company Example

- *Description.* At each month t, a warehouse contains s_t *items* of a specific goods and the demand for that goods is D (stochastic). At the end of each month the manager of the warehouse can *order* a_t more items from the supplier.

- The cost of maintaining an inventory of s is $h(s)$.
- The cost to order a items is $C(a)$.
- The income for selling q items if $f(q)$.
- If the demand $d \sim D$ is bigger than the available inventory s, customers that cannot be served leave.
- The value of the remaining inventory at the end of the year is $g(s)$.
- Constraint: the store has a maximum capacity C.

Recall: Markov Chains

Definition (Markov chain)

Let the *state space S* be a subset of the Euclidean space, the discrete-time dynamic system $(s_t)_{t\in\mathbb{N}} \in S$ is a Markov chain if it satisfies the *Markov property*

$$P(s_{t+1} = s | s_t, s_{t-1}, \ldots, s_0) = P(s_{t+1} = s | s_t),$$

Given an initial state $s_0 \in S$, a Markov chain is defined by the *transition probability* p

$$p(s'|s) = P(s_{t+1} = s' | s_t = s).$$

Markov Decision Process

Definition (Markov decision process)

A **Markov decision process** (MDP) is defined as a tuple $M = (S, A, P \text{ or } f, r, H)$ where

- S is the *state* space,

Example: The Amazing Goods Company

- State space: $s \in S = \{0, 1, \ldots, C\}$.

Markov Decision Process

Definition (Markov decision process)

A **Markov decision process** (MDP) is defined as a tuple $M = (S, A, P \text{ or } f, r, H)$ where

- S is the *state* space,
- A is the *action* space,

Example: The Amazing Goods Company

- Action space: it is not possible to order more items than the capacity of the store, so the action space should depend on the current state. Formally, at state s, $a \in A(s) = \{0, 1, \ldots, C - s\}$.

Markov Decision Process

Definition (Markov decision process)

A **Markov decision process** (MDP) is defined as a tuple $M = (S, A, P \text{ or } f, r, H)$ where

- S is the *state* space,
- A is the *action* space,

often simplified to finite

- $P(s'|s,a)$ is the transition probability with

$$P(s'|s,a) = \mathbb{P}(s_{t+1} = s'|s_t = s, a_t = a)$$

transition equation

$$s' = f_t(s, a, w_t)$$

where $w_t \sim W_t$

Example: The Amazing Goods Company

- Dynamics: $s_{t+1} = [s_t + a_t - d_t]^+$.
- The demand d_t is stochastic and time-independent. Formally, $d_t \overset{\text{i.i.d.}}{\sim} D$.

Markov Decision Process

Definition (Markov decision process)

A **Markov decision process** (MDP) is defined as a tuple $M = (S, A, P \text{ or } f, r, H)$ where

- S is the *state* space,
- A is the *action* space,

often simplified to finite

- $P(s'|s, a)$ is the transition probability with

$$P(s'|s,a) = \mathbb{P}(s_{t+1} = s'|s_t = s, a_t = a)$$

- $r(s, a, s')$ is the immediate reward at state s upon taking action a,

sometimes simply $r(s)$, assumed to be bounded

Example: The Amazing Goods Company

- Reward: $r_t = -C(a_t) - h(s_t + a_t) + f([s_t + a_t - s_{t+1}]^+)$. This corresponds to a purchasing cost, a cost for excess stock (storage, maintenance), and a reward for fulfilling orders.

Markov Decision Process

Definition (Markov decision process)

A **Markov decision process** (MDP) is defined as a tuple $M = (S, A, P \text{ or } f, r, H)$ where

- S is the *state* space,
- A is the *action* space,

often simplified to finite

- $P(s'|s,a)$ is the transition probability with
$$P(s'|s,a) = \mathbb{P}(s_{t+1} = s'|s_t = s, a_t = a)$$
- $r(s,a,s')$ is the immediate reward at state s upon taking action a,

sometimes simply $r(s)$

- H is the horizon.

Example: The Amazing Goods Company

- The horizon of the problem is 12 (12 months in 1 year).

Markov Decision Process (infinite horizon preview)

Definition (Markov decision process)

A **Markov decision process** (MDP) is defined as a tuple $M = (S, A, P \text{ or } f, r, H)$ where

- S is the *state* space,
- A is the *action* space,

often simplified to finite

- $P(s'|s,a)$ is the transition probability with

$$P(s'|s,a) = \mathbb{P}(s_{t+1} = s'|s_t = s, a_t = a)$$

- $r(s,a,s')$ is the immediate reward at state s upon taking action a,

sometimes simply $r(s)$

- $\gamma \in [0,1)$ is the discount factor.

Example: The Amazing Goods Company

- Discount: $\gamma = 0.91667$. A dollar today is worth more than a dollar tomorrow.
- The effective horizon of the problem is 12 (12 months in 1 year), i.e. $\mathrm{H} \approx \frac{1}{1-\gamma}$.

Markov Decision Process

Definition (Markov decision process)

A **Markov decision process** (MDP) is defined as a tuple $M = (S, A, P \text{ or } f, r, H)$ where

- S is the *state* space,
- A is the *action* space,

often simplified to finite

- $P(s'|s,a)$ is the transition probability with
$$P(s'|s,a) = \mathbb{P}(s_{t+1} = s'|s_t = s, a_t = a)$$
- $r(s,a,s')$ is the immediate reward at state s upon taking action a,

sometimes simply $r(s)$

- H is the horizon.

Example: The Amazing Goods Company

- Objective: $V(s_0; a_0, \dots) = \sum_{t=0}^{H-1} r_t + r_H$, where $\mathrm{r}_{12} = \mathrm{g}(\mathrm{s}_{12})$. This corresponds to the cumulative reward, including the value of the remaining inventory at "the end."

Markov Decision Process

Definition (Markov decision process)

A **Markov decision process** (MDP) is defined as a tuple $M = (S, A, P \text{ or } f, r, H)$ where

- S is the *state* space,
- A is the *action* space,

often simplified to finite

- $P(s'|s,a)$ is the transition probability with

$$P(s'|s,a) = \mathbb{P}(s_{t+1} = s'|s_t = s, a_t = a)$$

- $r(s,a,s')$ is the immediate reward at state s upon taking action a,

sometimes simply $r(s)$

- H is the horizon.

☞ In general, a non-Markovian decision process's transitions could depend on much more information:

$$\mathbb{P}(s_{t+1} = s'|s_t = s, a_t = a, s_{t-1}, a_{t-1}, \ldots, s_0, a_0),$$

Markov Decision Process

Definition (Markov decision process)

A **Markov decision process** (MDP) is defined as a tuple $M = (S, A, P \text{ or } f, r, H)$ where

- S is the *state* space,
- A is the *action* space,

often simplified to finite

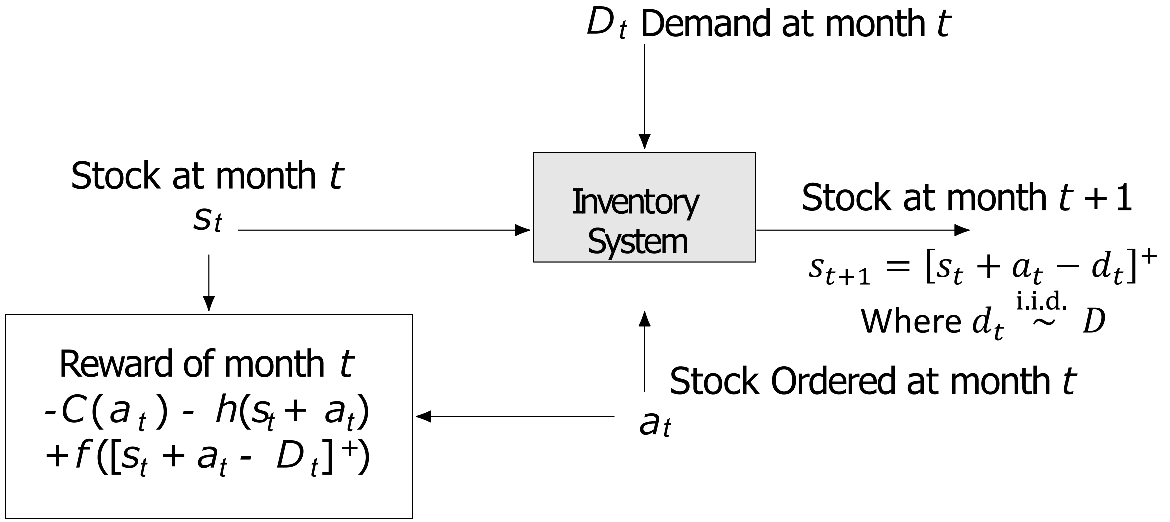
- $P(s'|s, a)$ is the transition probability with
$$P(s'|s, a) = \mathbb{P}(s_{t+1} = s'|s_t = s, a_t = a)$$
- $r(s, a, s')$ is the immediate reward at state s upon taking action a,

sometimes simply $r(s)$

- H is the horizon.

☞ The process generates trajectories $\tau_t = (s_0, a_0, \ldots, s_{t-1}, a_{t-1}, s_t)$, with $s_{t+1} \sim P(\cdot\, | s_t, a_t)$

Example: The Amazing Goods Company Example



- State space: $s \in S = \{0, 1, \ldots, C\}$.
- Action space: it is not possible to order more items than the capacity of the store, so the action space should depend on the current state. Formally, at state s, $a \in A(s) = \{0, 1, \ldots, C - s\}$.
- Objective: $V(s_0; a_0, \ldots) = \sum_{t=0}^{H-1} r_t + r_H$, where $H = 12$ and $r_{12} = g(s_{12})$

Modeling real-time rideshare matching as an MDP

For each driver:

- **State**: Location, time, and vehicle type of the idle driver
- **Action**: Request destination location/time
- **Reward**: Expected assignment earnings or 0 for idle drivers
- **Time step**: Total trip duration or 4 sec for idle drivers
- **Transition function**: New idle state of the driver given request (deterministic)
- **Horizon**: Infinite
  - **Discount factor**: γ = 0.9992, or a half-life of roughly one hour using a four second time-step (i.e. $\gamma^{3600/4} \approx 0.5$)

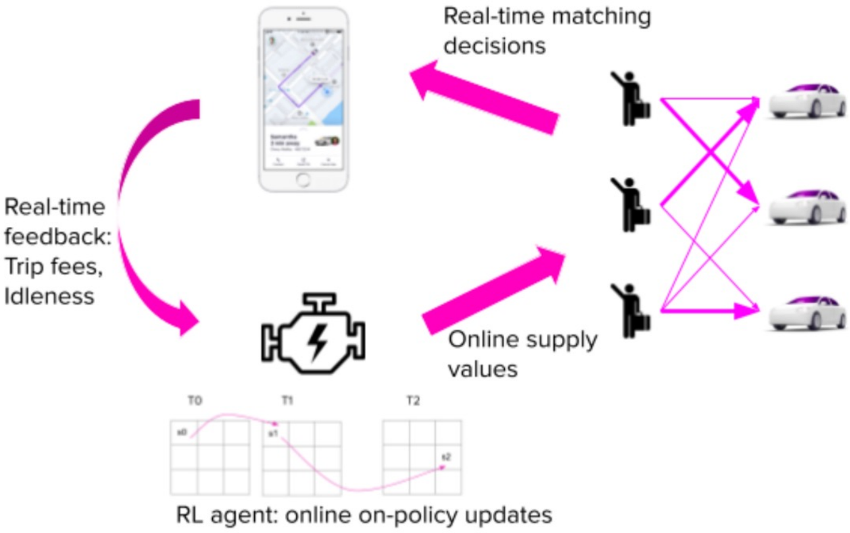


Figure 7. The Graphic Depicts the Online RL (Approximate Value Iterations) Framework

B. Han, H. Lee, and S. Martin, "Real-Time Rideshare Driver Supply Values Using Online Reinforcement Learning," in *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, in KDD '22. Aug. 2022. doi: 10.1145/3534678.3539141.

Key challenge: huge state spaces

- State: location, time, and vehicle type
  - Location is encoded from geohash6 (precise location) [1,600] and geohash5 (neighborhood) [50]
  - Time encoded from hour-of-week categories [168]
  - Vehicle type: standard, luxury, SUV, or handicap accessible [4]
- State space is ≈1600x50x168x4=54M
- For reference: SF Bay Area population is 8M
  - Naïve approach: Would need everyone to take at least 7 rides to gather enough data

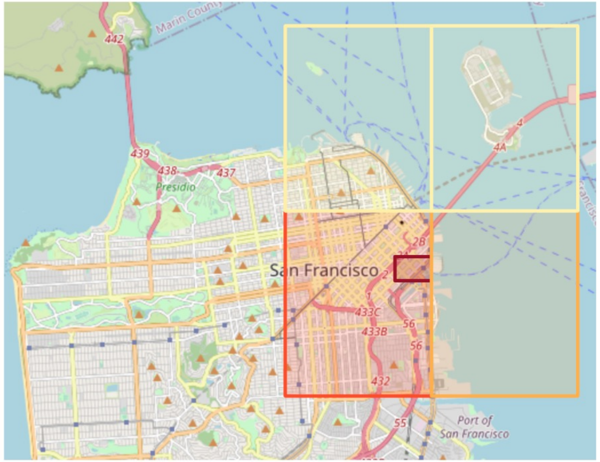


Figure 4: Spatial factor weights are weighted and normalized by the inverse of the distance from the geohash centroid to smoothly interpolate the four closest geohash5 state factors. A similiar interpolation is applied using the two nearest hours of the week, yielding a cross-product of eight spatiotemporal factors and weights. Additional factors also consider the vehicle type, such as standard, luxury, SUV, or handicap accessible.

Cannot only explore. Cannot only exploit.

Must trade off exploration and exploitation.

B. Han, H. Lee, and S. Martin, "Real-Time Rideshare Driver Supply Values Using Online Reinforcement Learning," in *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, in KDD '22. Aug. 2022. doi: 10.1145/3534678.3539141.

Transition function

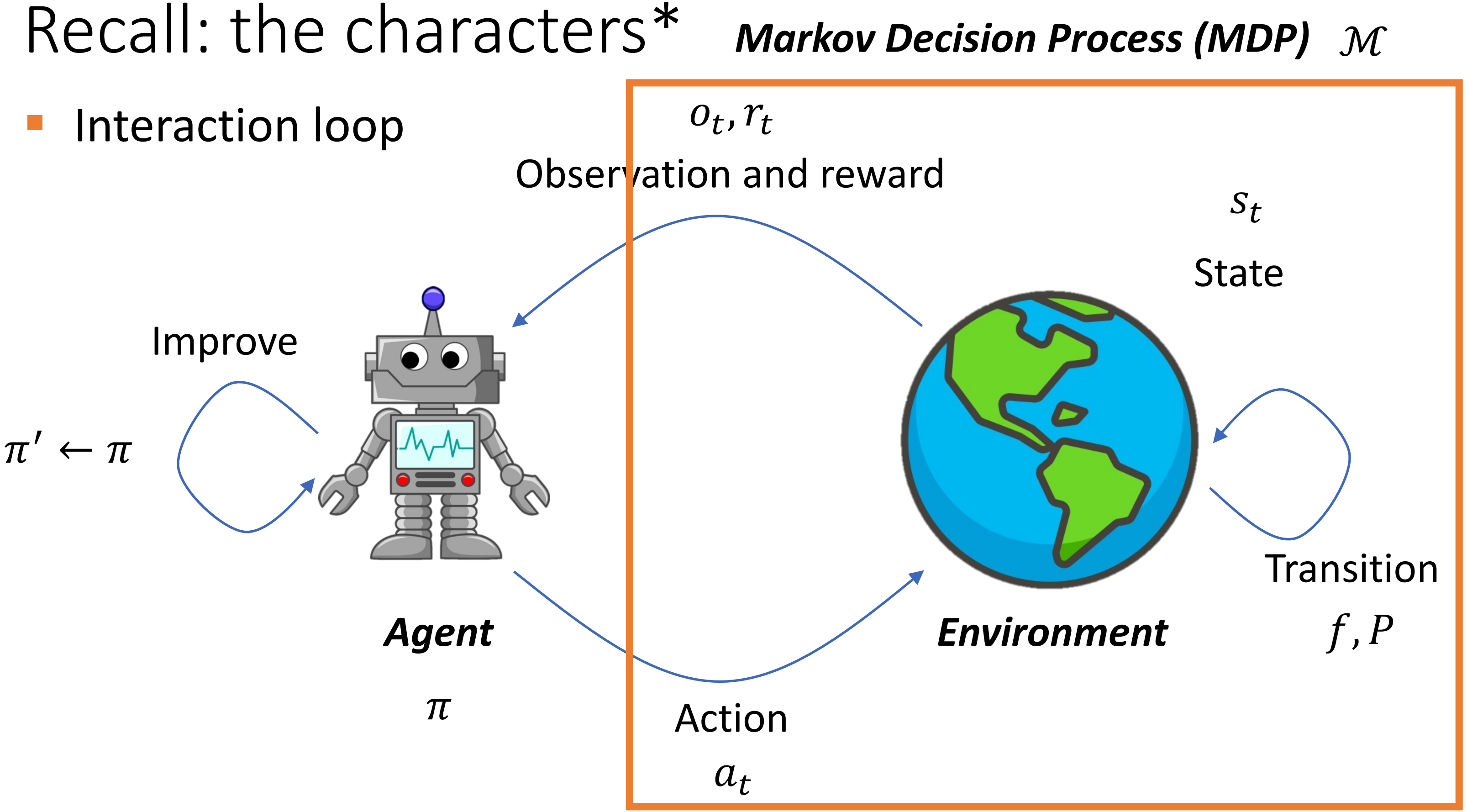
Driver $i \in \{1, \dots, m\}$
Request $j \in \{1, \dots, n\}$
Assignment $a_{ij} \in \{0,1\}$
Immediate reward r_{ij}
Probability of cancellation p_{ij}

Outline

1. Reinforcement learning for transportation
2. Markov Decision Process (MDP)
3. **The optimization problem**
    a. Value function
    b. Policy
    c. Mixed autonomy traffic (2017)
4. Emergency medical service (EMS) vehicle problem

Recall: the characters*

- Interaction loop



Goal: maximize reward over time (returns, cumulative reward)

* pun intended

The value function

Given a policy π (deterministic to simplify notation)

- Finite time horizon T: deadline at time T, the agent focuses on the sum of the rewards up to T.

$$V^\pi(t,s) = \mathbb{E}\left[\sum_{\tau=t}^{T-1} r(s_\tau, \pi(s_\tau)) + R(s_T) | s_t = s; \pi\right]$$

where R is a value function for the final state.

- Shorthand: $V_t^\pi(s)$ or simply V_t^π (think: vector of size $|S|$)

Optimization Problem

- Our goal: achieve the best value
  - Max value-to-go (min cost-to-go)

Definition (Optimal policy and optimal value function)

The solution to an MDP is an optimal policy π^* satisfying

$$\pi^* \in \arg\max_{\pi \in \Pi} V_0^{\pi}$$

where Π is some policy set of interest.

The corresponding value function is the optimal value function

$$V^* = V_0^{\pi^*}$$

Expectations

- Technical note: the expectations refer to all possible stochastic trajectories.
- A (possibly non-stationary stochastic) policy π applied from state s_0 returns

$$(s_0, r_0, s_1, r_1, s_2, r_2, \dots)$$

- Where $r_t = r(s_t, a_t)$ and $s_{t+1} \sim p(\cdot \mid s_t, a_t = \pi_t(s_t))$ are random realizations.

- The value function is

$$V^\pi(t, s) = \mathbb{E}_{(s_1, s_2, \dots)}\left[\sum_{\tau=t}^{T-1} r(s_\tau, \pi(s_\tau)) + R(s_T) \middle| s_t = s; \pi\right]$$

- More generally, for stochastic policies:

$$V^\pi(t, s) = \mathbb{E}_{(a_0, s_1, a_1, s_2, \dots)}\left[\sum_{\tau=t}^{T-1} r(s_\tau, \pi(s_\tau)) + R(s_T) \middle| s_t = s; \pi\right]$$

Driver $i \in \{1, \dots, m\}$
Request $j \in \{1, \dots, n\}$
Assignment $a_{ij} \in \{0,1\}$
Immediate reward r_{ij}
Probability of cancellation p_{ij}

Real-time rideshare matching

At each round (every 4 seconds), solve:

Bipartite Matching

$$\operatorname*{argmax}_{a_{ij}} \sum_{i=1}^{m} \sum_{j=1}^{n} r_{ij} a_{ij}$$

$$\text{subject to} \sum_{i=1}^{m} a_{ij} \le 1, \quad j = 1, 2, 3, \dots, n$$

$$\sum_{j=1}^{n} a_{ij} \le 1, \quad i = 1, 2, 3, \dots, m$$

$$a_{ij} \in \{0, 1\}, \quad \forall (i, j)$$

Supply Value Dispatch

$$\operatorname*{argmax}_{a_{ij}} \sum_{i=1}^{m} \sum_{j=1}^{n} \boxed{\Delta_{ij}} a_{ij}$$

Instead, want the match to capture long-term value!

$$\text{subject to} \sum_{i=1}^{m} a_{ij} \le 1, \quad j = 1, 2, 3, \dots, n$$

$$\sum_{j=1}^{n} a_{ij} = 1, \quad i = 1, 2, 3, \dots, m$$

$$a_{ij} \in \{0, 1\}, \quad \forall (i, j)$$

Preview of Unit 3:

Advantage $\Delta_{ij} = Q(s_i, a_{ij}) - V(s_i)$ State-value $V(s) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t r_t \mid s_0 = s]$

Action-value $Q(s_i, a_{ij}) = r_{ij} + (1 - p_{ij})\gamma^{d_{ij}} V(s_{ij}) + p_{ij}\gamma V(s_i)$ RL methods solve for V, Q, Δ_{ij}

B. Han, H. Lee, and S. Martin, "Real-Time Rideshare Driver Supply Values Using Online Reinforcement Learning," in *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, in KDD '22. Aug. 2022. doi: 10.1145/3534678.3539141.

Reinforcement learning for real-time ridesharing

Name	**Description**	**Impact of RL Approach**
Unavailability	Ride requests for which we could not find a driver to match divided by total number of ride requests	-13.0%
Rider cancellation	Ride requests canceled by a rider divided by the total number of ride requests	-3.0%
Five-star ratings	Completed rides with five-star rating (maximum rating) divided by the total number of completed rides	+1.0%
Revenue (annualized)	Expected incremental revenue (with respect to the baseline) summed across the calendar year	>$30 million

Table 2. The Table Shows the Results We Achieved from Our Experiments on the RL Approach

X. Azagirre *et al.*, "A Better Match for Drivers and Riders: Reinforcement Learning at Lyft." INFORMS Journal on Applied Analytics, 2023. doi: [10.48550/arXiv.2310.13810](https://doi.org/10.48550/arXiv.2310.13810).

Policy

Definition (Policy)

A decision rule d can be

- Deterministic: $d: S \to A$,
- Stochastic: $d: S \to \Delta(A)$,
- History-dependent: $d: H_t \to A$,
- Markov: $d: S \to \Delta(A)$,

A policy (strategy, plan) can be

- Stationary: $\pi = (d, d, d, \dots)$,
- (More generally) Non-stationary: $\pi = (d_0, d_1, d_2, \dots)$

☞ For simplicity, we will typically write π instead of d for stationary policies, and π_t instead of d_t for non-stationary policies.

Recall: The Amazing Goods Company Example

- *Description.* At each month t, a warehouse contains s_t *items* of a specific goods and the demand for that goods is D (stochastic). At the end of each month the manager of the warehouse can *order* a_t more items from the supplier.

- The cost of maintaining an inventory of s is $h(s)$.
- The cost to order a items is $C(a)$.
- The income for selling q items if $f(q)$.
- If the demand $d \sim D$ is bigger than the available inventory s, customers that cannot be served leave.
- The value of the remaining inventory at the end of the year is $g(s)$.
- Constraint: the store has a maximum capacity C.

Recall: The Amazing Goods Company Example

- *Description.* At each month t, a warehouse contains s_t *items* of a specific goods and the demand for that goods is D (stochastic). At the end of each month the manager of the warehouse can *order* a_t more items from the supplier.

- The cost of maintaining an inventory of s is $h(s)$.
- The cost to order a items is $C(a)$.
- The income for selling q items if $f(q)$.
- If the demand $d \sim D$ is bigger than the available inventory s, customers that cannot be served leave.
- The value of the remaining inventory at the end of the year is $g(s)$.
- Constraint: the store has a maximum capacity C.

Stationary policy composed of deterministic Markov decision rules

$$\pi(s) = \begin{cases} C - s & \text{if } s < M/4 \\ 0 & \text{otherwise} \end{cases}$$

Recall: The Amazing Goods Company Example

- *Description.* At each month t, a warehouse contains s_t *items* of a specific goods and the demand for that goods is D (stochastic). At the end of each month the manager of the warehouse can *order* a_t more items from the supplier.

- The cost of maintaining an inventory of s is $h(s)$.
- The cost to order a items is $C(a)$.
- The income for selling q items if $f(q)$.
- If the demand $d \sim D$ is bigger than the available inventory s, customers that cannot be served leave.
- The value of the remaining inventory at the end of the year is $g(s)$.
- Constraint: the store has a maximum capacity C.

Stationary policy composed of stochastic history-dependent decision rules

$$\pi(s_t) = \begin{cases} U(C - s_{t-1}, C - s_{t-1} + 10) & \text{if } s_t < s_{t-1}/2 \\ 0 & \text{otherwise} \end{cases}$$

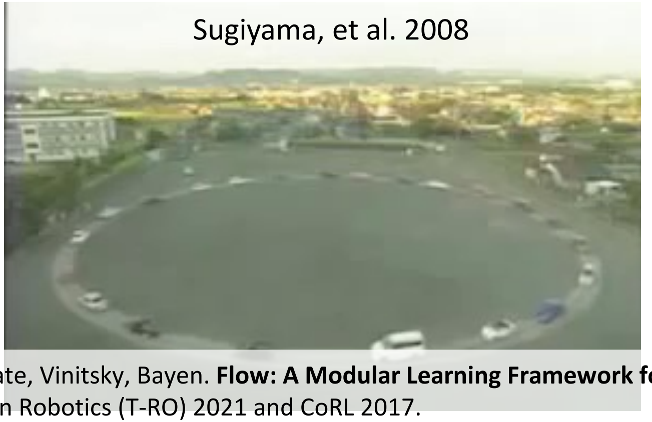
Traffic flow smoothing

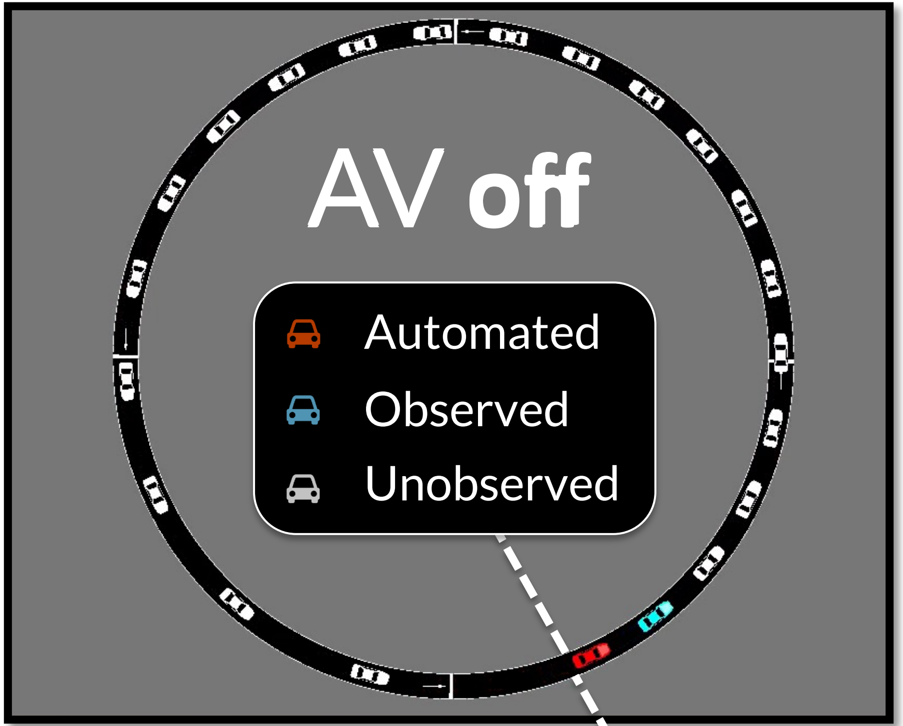
Sugiyama, et al. 2008

Wu, et al. "Flow: A Modular Learning Framework for Mixed Autonomy Traffic." T-RO, 2021.

Traffic flow smoothing

- Setup
  - Circular track. Sufficient to reproduce traffic waves & jams.
  - 1 self-driving car, 21 human drivers
  - State: relative velocity & headway
  - Action: acceleration
  - Reward: average velocity (for all cars)
  - Timestep: 0.1 sec
  - Horizon: 5 minutes
  - Algorithm: TRPO



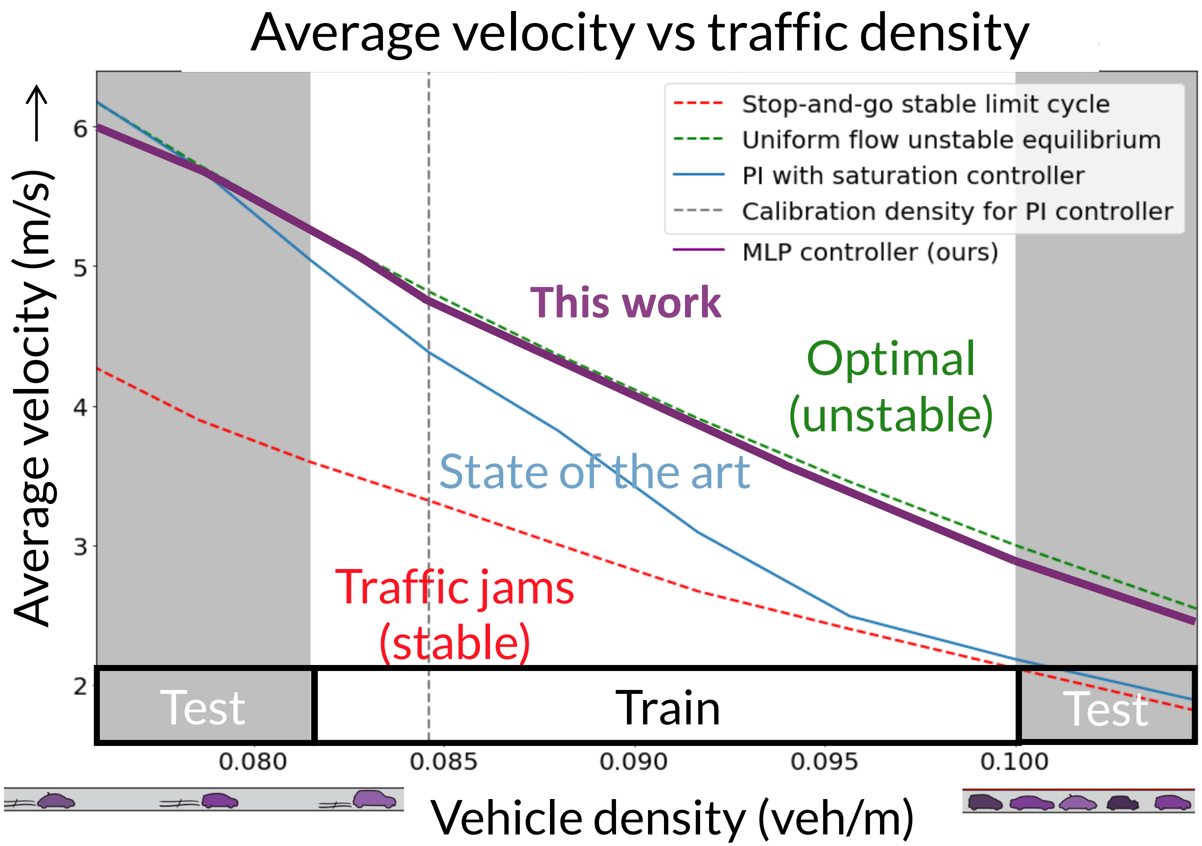


SUMO

Wu, Kreidieh, Parvate, Vinitsky, Bayen. **Flow: A Modular Learning Framework for Mixed Autonomy Traffic**. IEEE Transactions on Robotics (T-RO) 2021 and CoRL 2017.

Traffic flow smoothing

- 5% AVs → 50% improvement in velocity for <u>all</u> cars
- Near-optimal
- Robust
- Training time: a few hours on 1 CPU
- Tweaks that made it work
  - Partial observation sufficient → fast training
  - "Sufficient": Control theory → optimal performance



Traffic flow smoothing: model interpretation

Deep neural network

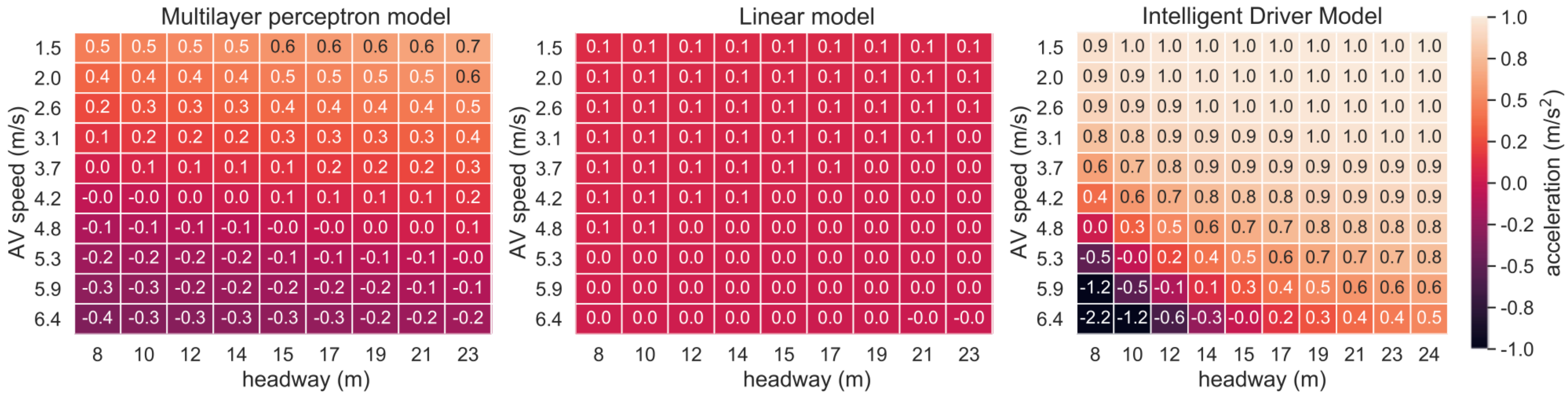
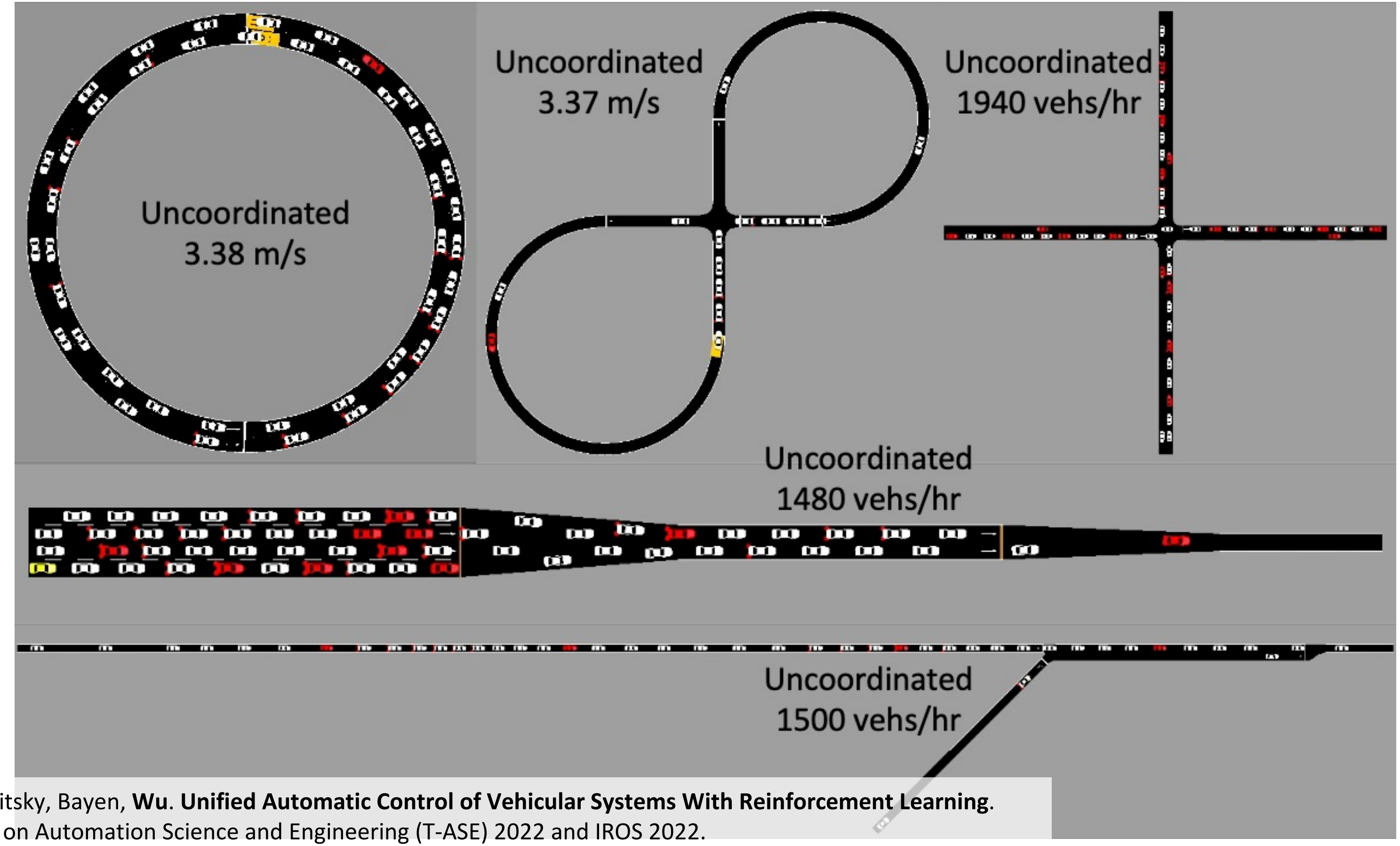


Fig. 6. *Visualization of vehicle control laws.* The heatmaps are 2-D slices of the controllers (3-D), and the color depicts the output (acceleration). The *x*-axis is a representative range of headways seen by vehicles during training. The *y*-axis is a representative range of AV speeds. Displayed is the slice of acceleration values of the model when the leader vehicle speed is fixed at 4.2 m/s (a typical speed for the 250 m track). The single colorbar is shared by all plots. *Left*: Learned MLP model, with failsafes disabled. *Middle*: Learned Linear model, with failsafes enabled. *Right*: IDM.

C. **Wu**, A. R. Kreidieh, K. Parvate, E. Vinitsky, and A. M. Bayen, “Flow: A modular learning framework for mixed autonomy traffic,” *IEEE Transactions on Robotics (T-RO)*, Jul. 2021, doi: 10.1109/TRO.2021.3087314.

RL + traffic LEGO blocks

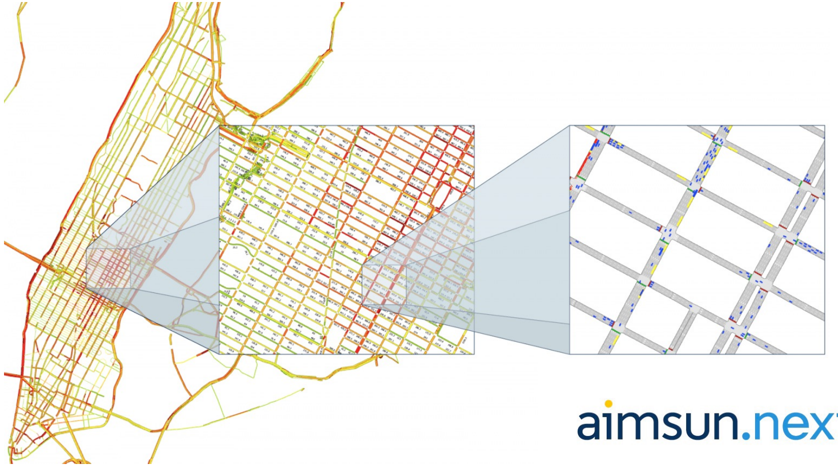
5-30% CAVs → 13-120% improvement



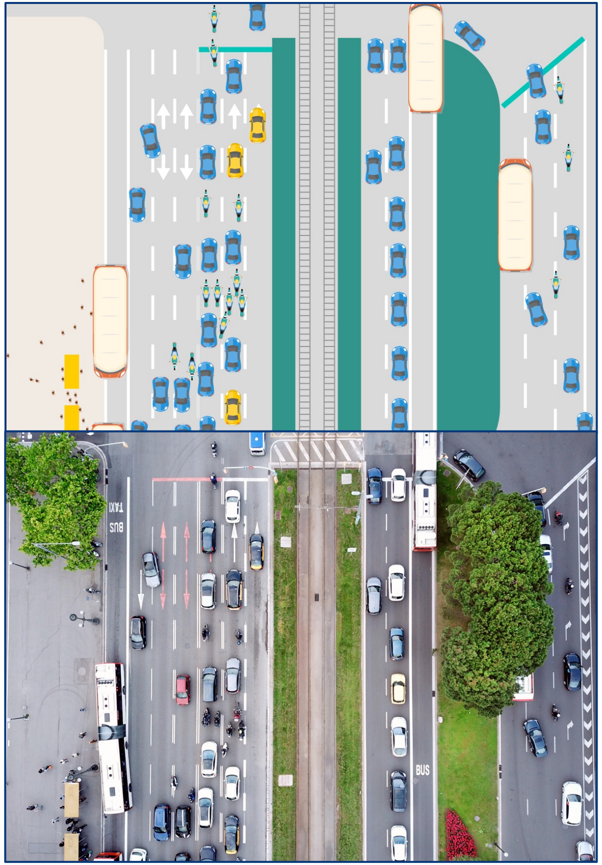
Yan, Kreidieh, Vinitsky, Bayen, **Wu**. **Unified Automatic Control of Vehicular Systems With Reinforcement Learning**.
IEEE Transactions on Automation Science and Engineering (T-ASE) 2022 and IROS 2022.

Traffic flow smoothing

- Near misses
  - Traffic is notoriously difficult to analyze (cascaded nonlinear dynamics, delayed effects, multi-agent, partially observed)
  - Human driving is fairly predictable in aggregate → can simulate data
  - Traffic phenomena can be reproduced with minimal system complexity → cheap simulation

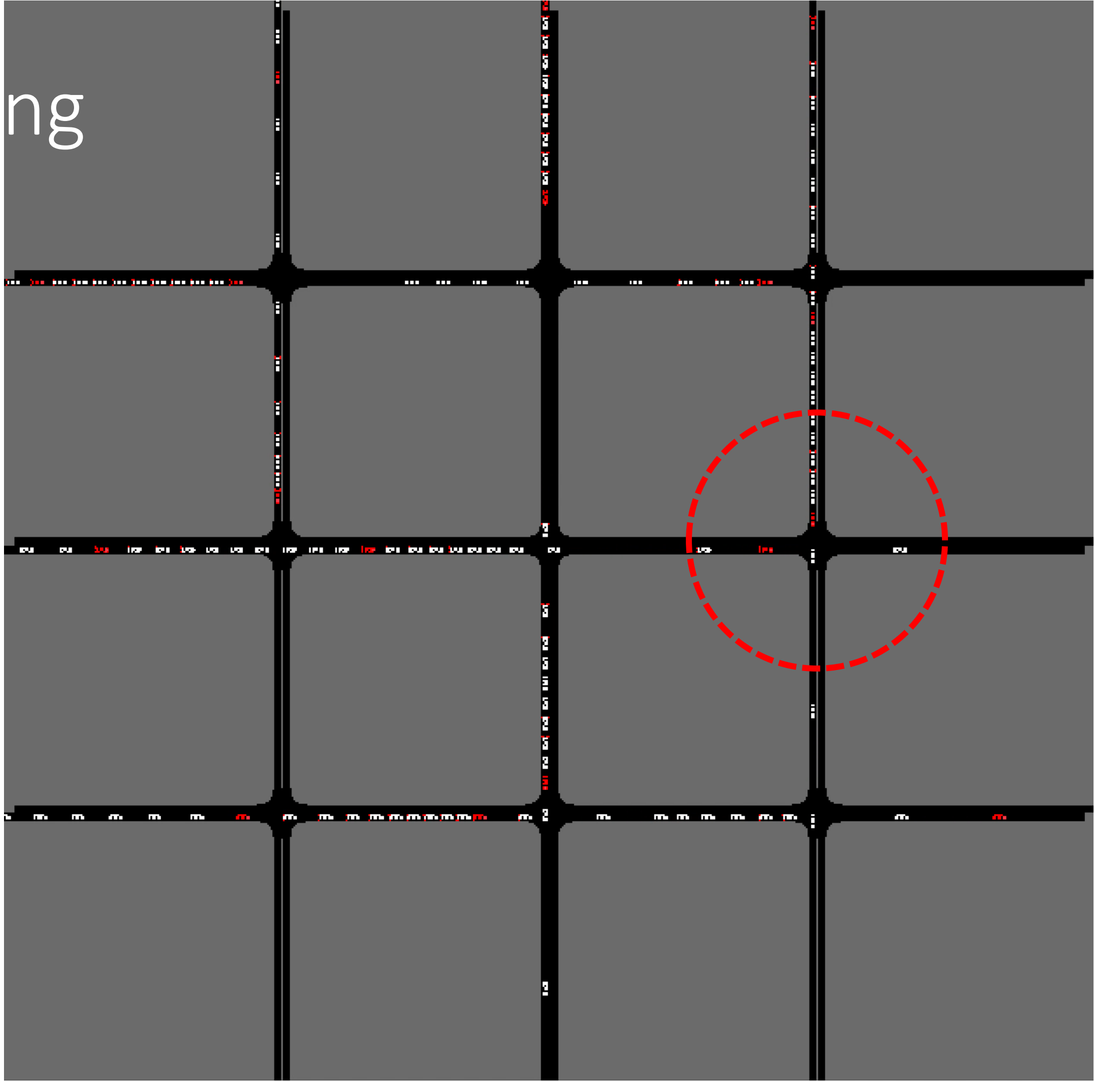


aimsun.next



Traffic flow smoothing

- Challenge: grid network
  - Long-horizon multi-agent coordination & control
- Result: 10% AVs → 26% improvement over human driving baseline
- Tweaks that make it work
  - Shared parameter (homogenous) multi-agent training
  - Restricted observation space
  - (Zero-shot) transfer learning



Yan, Z. and **Wu, C**. In review.

Outline

1. Reinforcement learning for transportation
2. Markov Decision Process (MDP)
3. The optimization problem
4. **Emergency medical service (EMS) vehicle problem**

EMS maneuver under mixed autonomy

- Emergency medical service (EMS) vehicle
- Scenario:
  - EMS may stop or travel at low speeds on congested roads (e.g., signalized intersections)

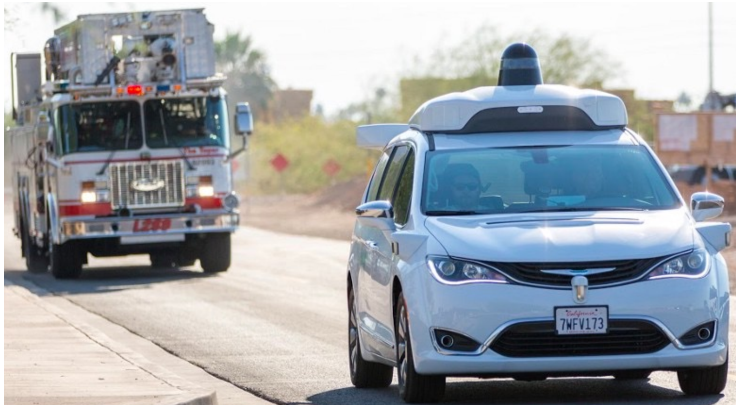
- Motivation: **Reduce** emergency service vehicle (EMS) **travel times** to reduce mortality rate [OBENAUF et al. 2019]

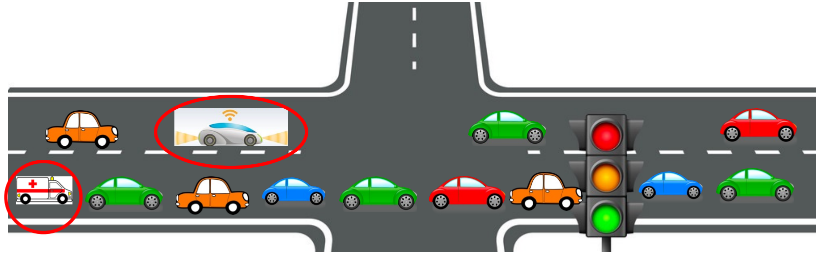
Originated as 1.200 class project!

D. Suo*, V. Jayawardana*, and **C. Wu**, "Model-free Learning of Corridor Clearance: A Near-term Deployment Perspective," *IEEE Transactions on Intelligent Transportation Systems (T-ITS)*, 2024, doi: 10.1109/TITS.2023.3344473.

EMS maneuver under mixed autonomy (Suo et al., 2023)

- Problem: How should the AV take maneuvers to assist EMS in crossing intersections?
- A specific scenario:
  - Right lane (where the EMS currently locates) fully congested
  - An autonomous vehicle can receive inputs from onboard sensors (e.g., lidar, radar, camera)
  - The AV can communicate with traffic infrastructure and EMS for non-line-of-sight conditions
- The goal of the AV is to assist EMS maneuvers to reduce its travel time crossing the intersection





Exercise: Define a Markov Decision Process to model the problem, including the state space, action space, transitions, reward, and objective function.

Based on: Suo, Jayawardana, **Wu**. "Model-free Learning of Multi-objective Corridor Clearance in Mixed Autonomy," 2023. Under review.

EMS maneuver under mixed autonomy

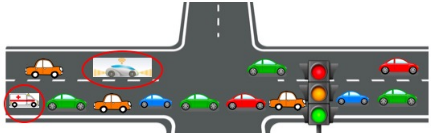
What **observations** available to the AV?
How to define **reward** for the AV?

o_t, r_t

s_t
State?

Improve

$\pi' \leftarrow \pi$



Transition

P

Agent
π

Environment
$\mathcal{M}$

Actions available to the AV?
a_t

Based on: Suo, Jayawardana, **Wu**. "Model-free Learning of Multi-objective Corridor Clearance in Mixed Autonomy," 2023. Under review.

References

1. Morales, Miguel. Grokking deep reinforcement learning. 2020. Chapter 2: Mathematical Foundations of Reinforcement Learning.
2. Morales, Miguel. Grokking deep reinforcement learning. 2020. Chapter 1: Introduction to Deep Reinforcement Learning.
3. Sutton, R. S. and Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.
4. Some slides adapted from Alessandro Lazaric, Matteo Pirotta, Cameron Hickert.