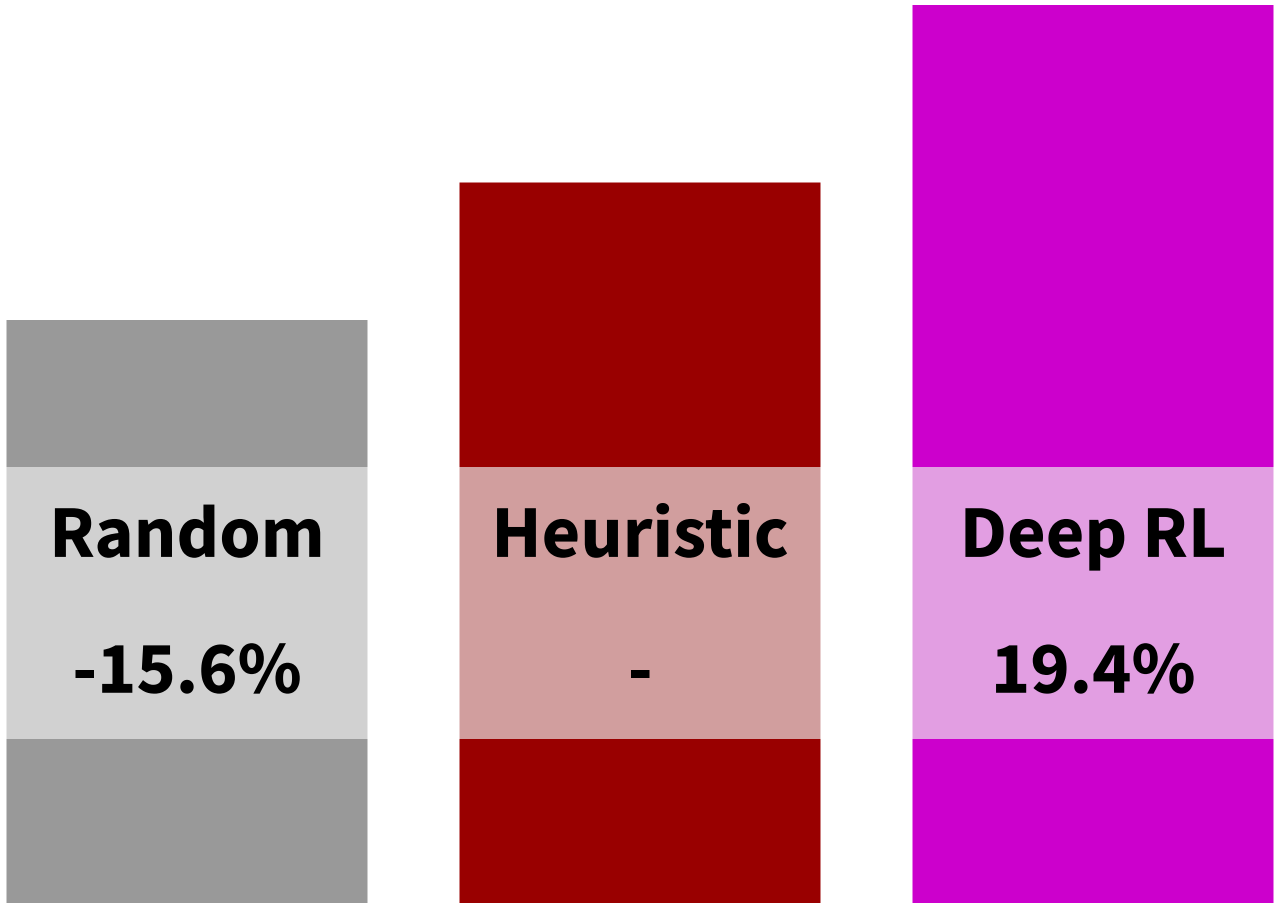



Control of Autonomous Electric Fleets for Ridehail Systems

ND Kullman, JE Mendoza, M Cousineau, JC Goodson

Deep RL + OR
Policies + Bound

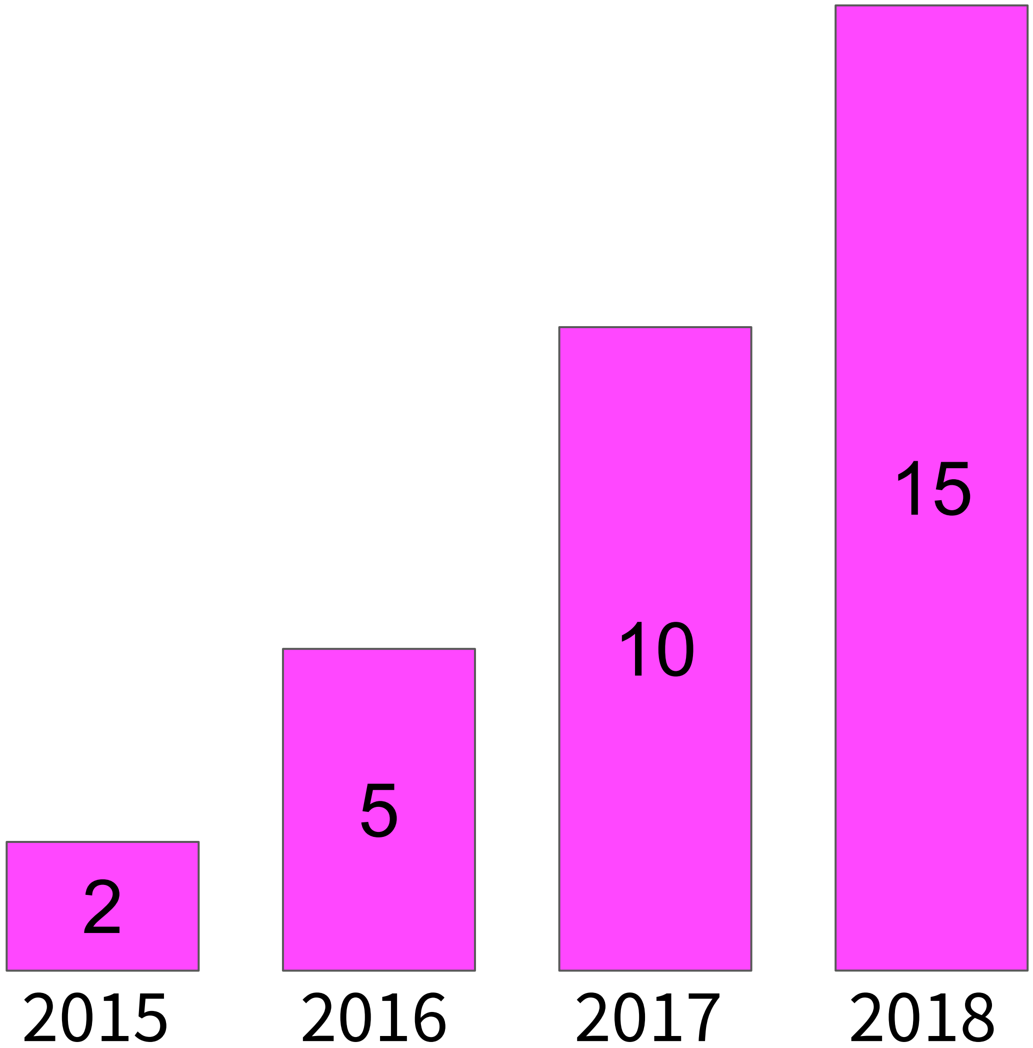
Teaser: Some success so far



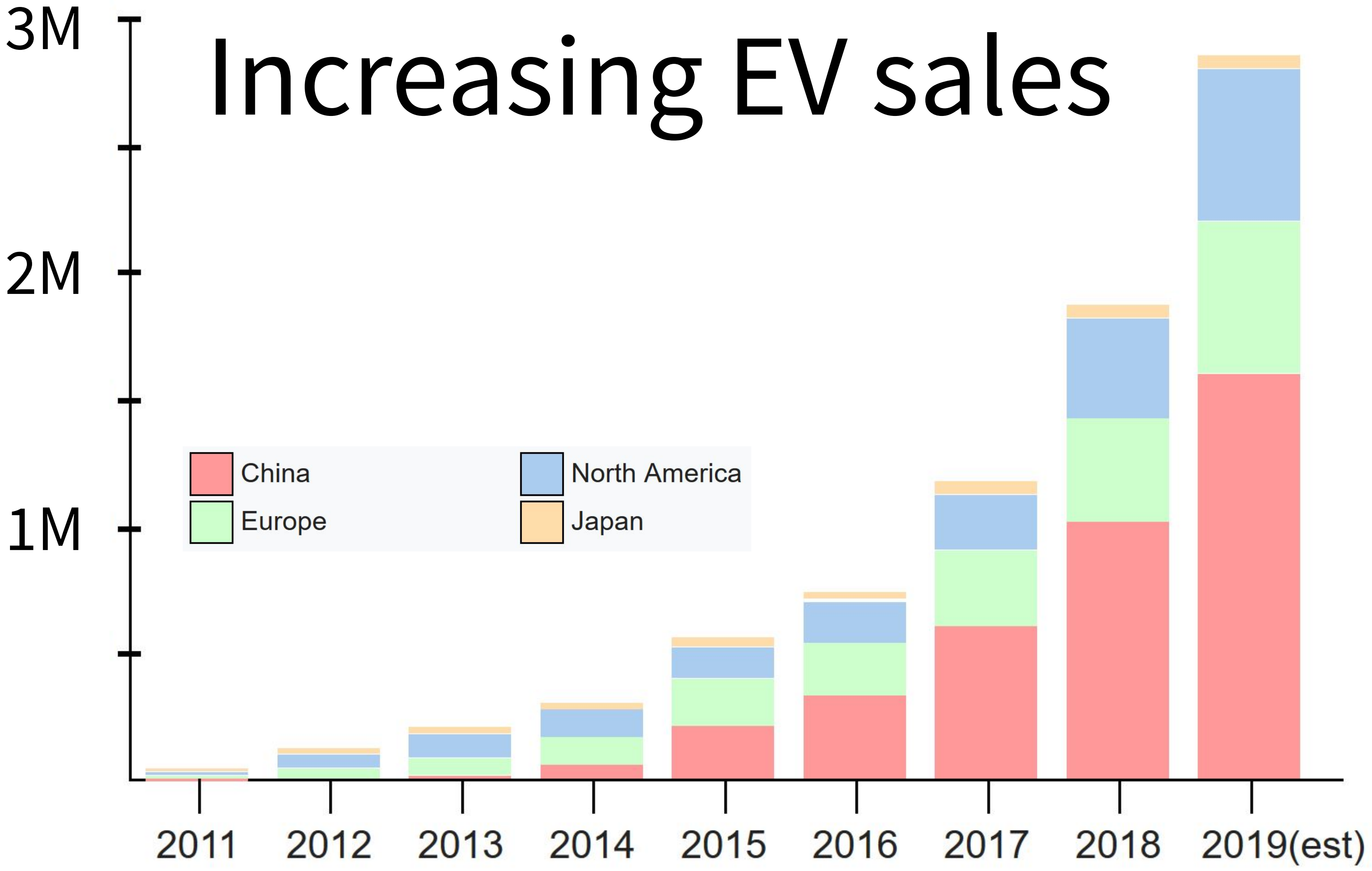
Background

Ridehail explosion

NYC ridehail trips
(M, avg/mo)



Increasing EV sales



Autonomy imminent

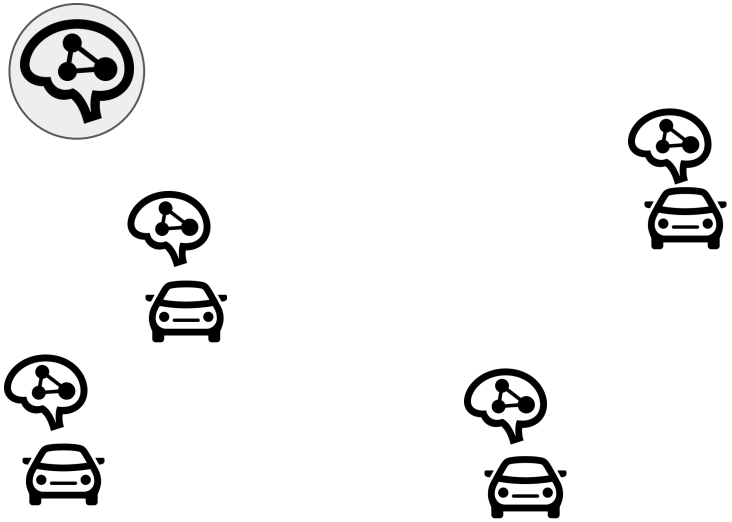
WAYMO

Autonomy imminent

TESLA

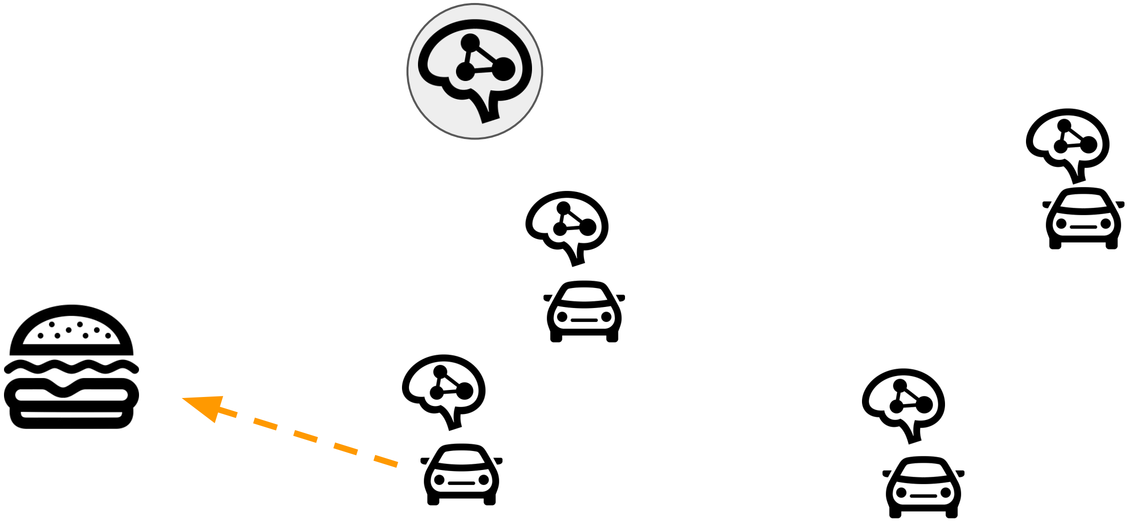
Control of AEV Fleets is Centralized

Today



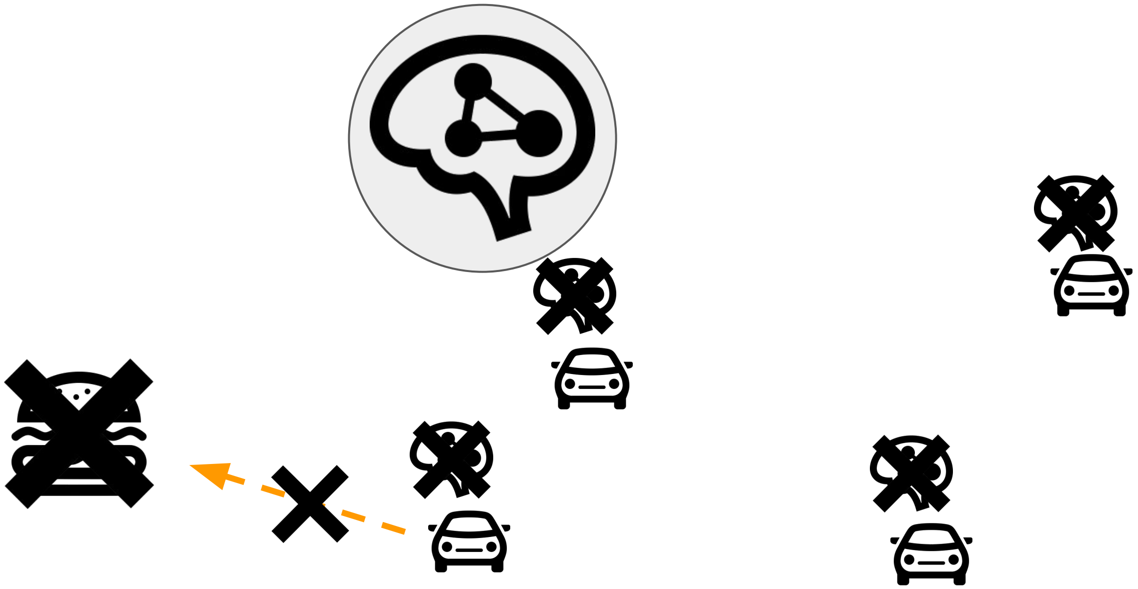
Control of AEV Fleets is Centralized

Today



Control of AEV Fleets is Centralized

w/ AEVs



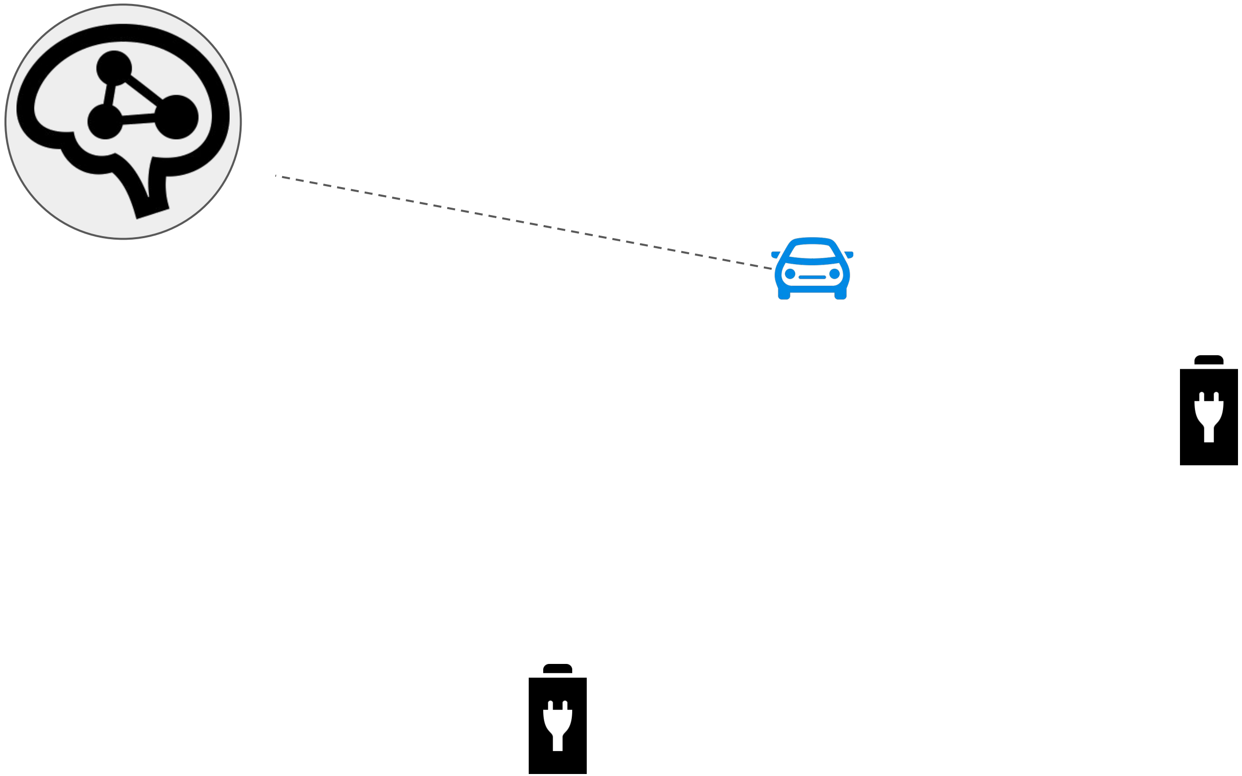
How to Control a Fleet of AEVs

How to Control a Fleet of AEVs

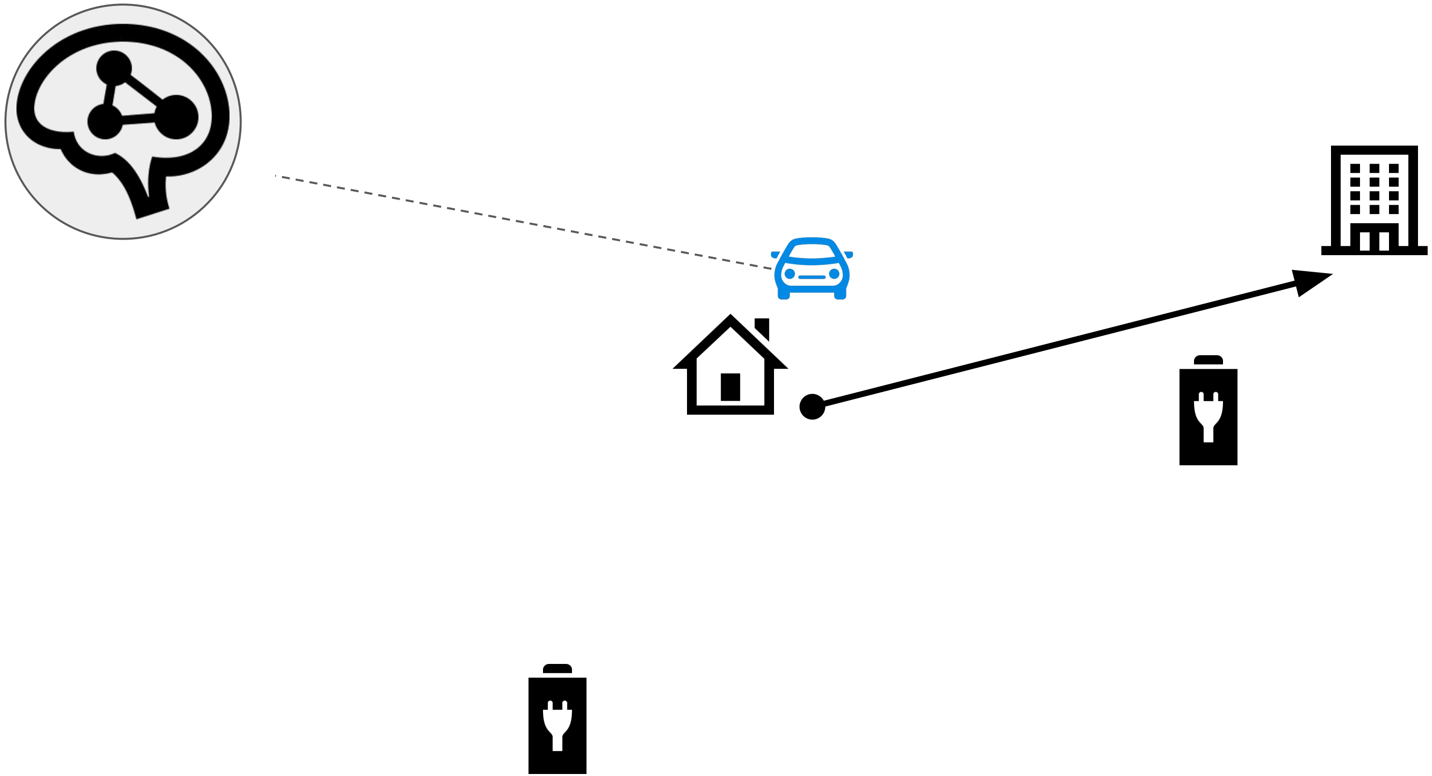
Assign vehicles to requests

Recharge/reposition vehicles

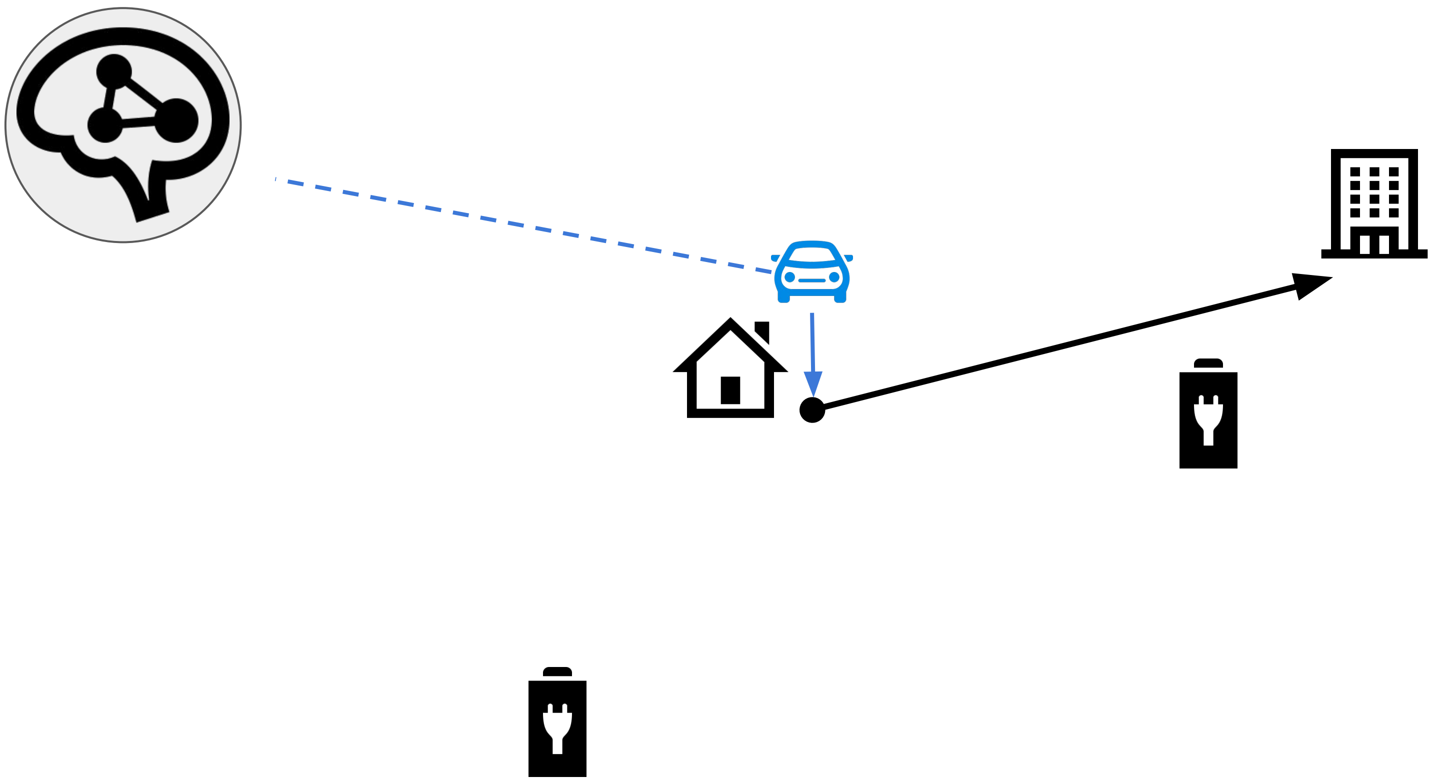
How to Control a Fleet of AEVs



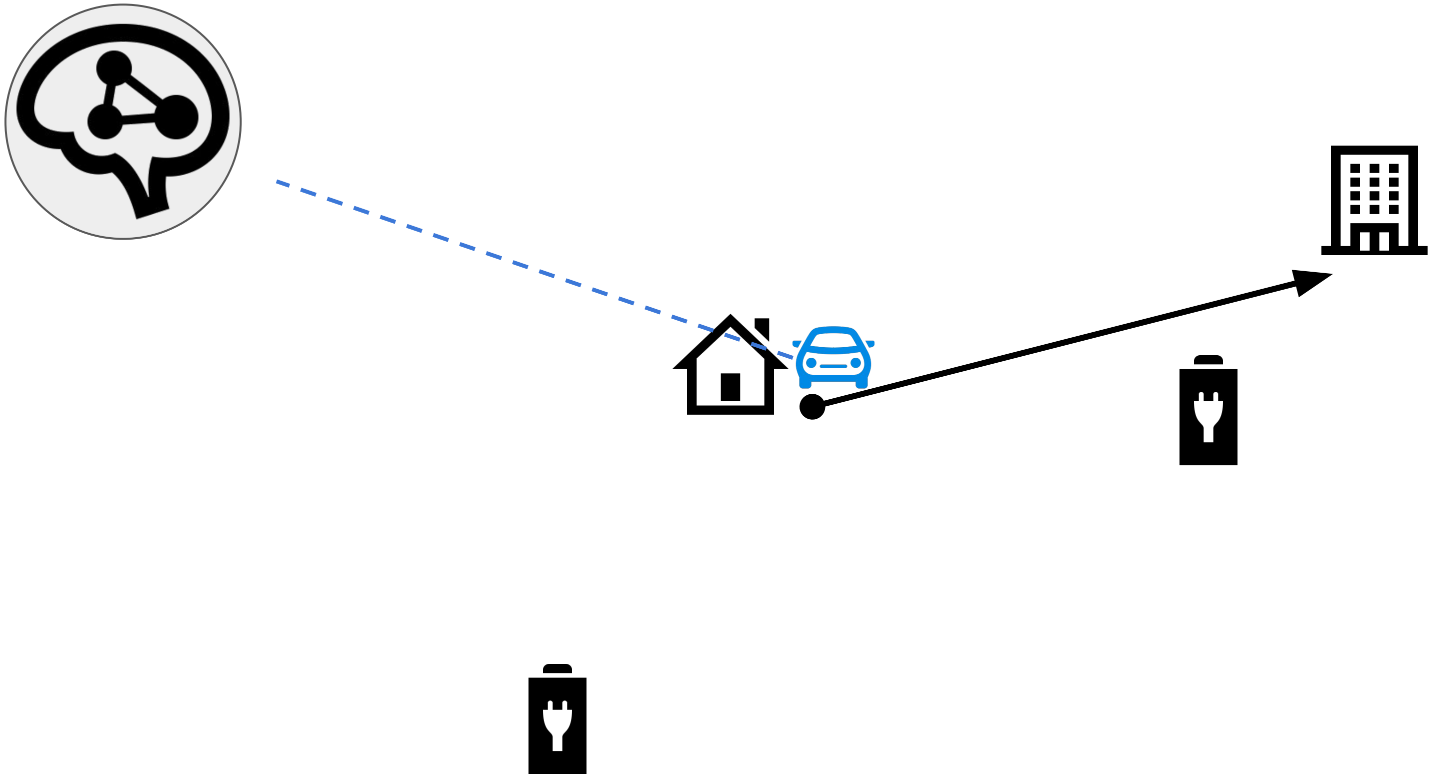
How to Control a Fleet of AEVs



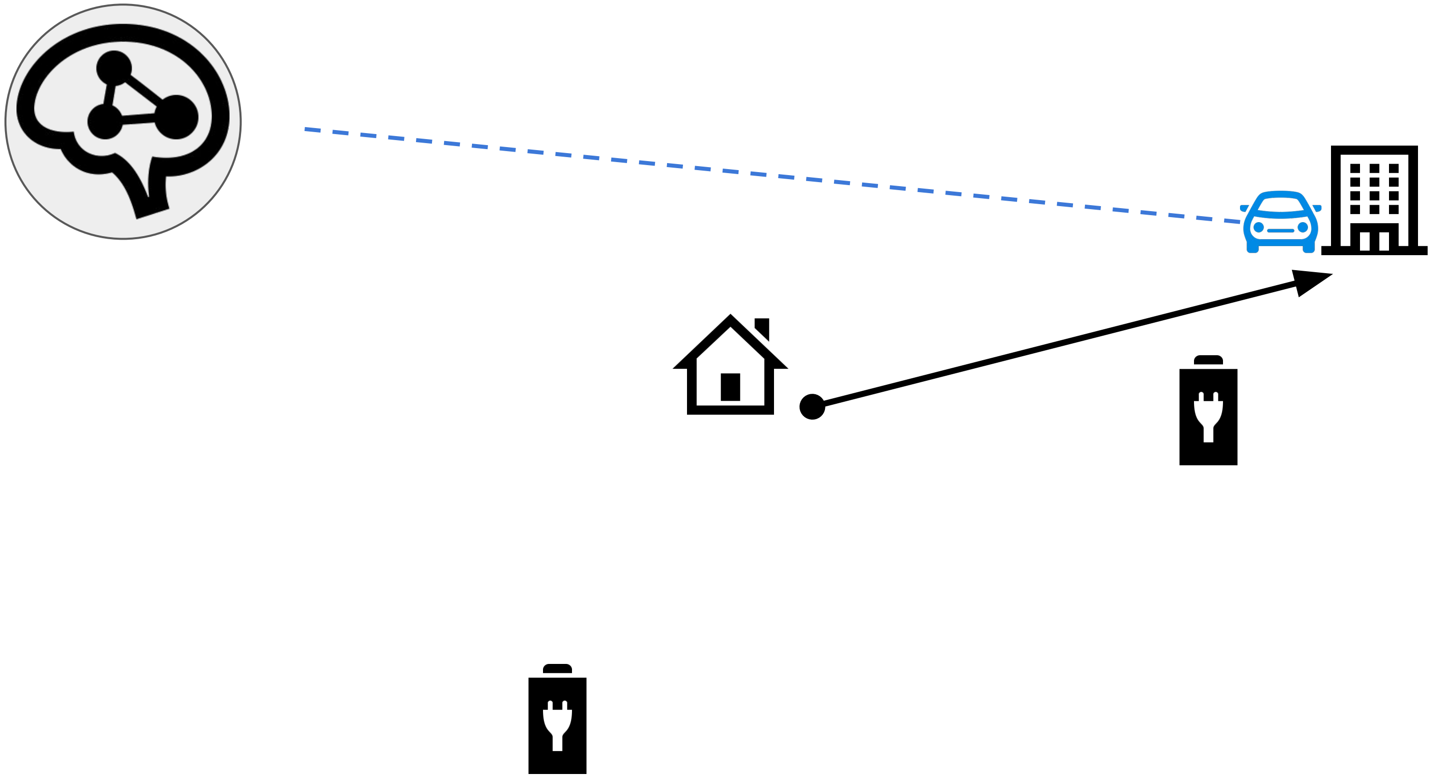
How to Control a Fleet of AEVs



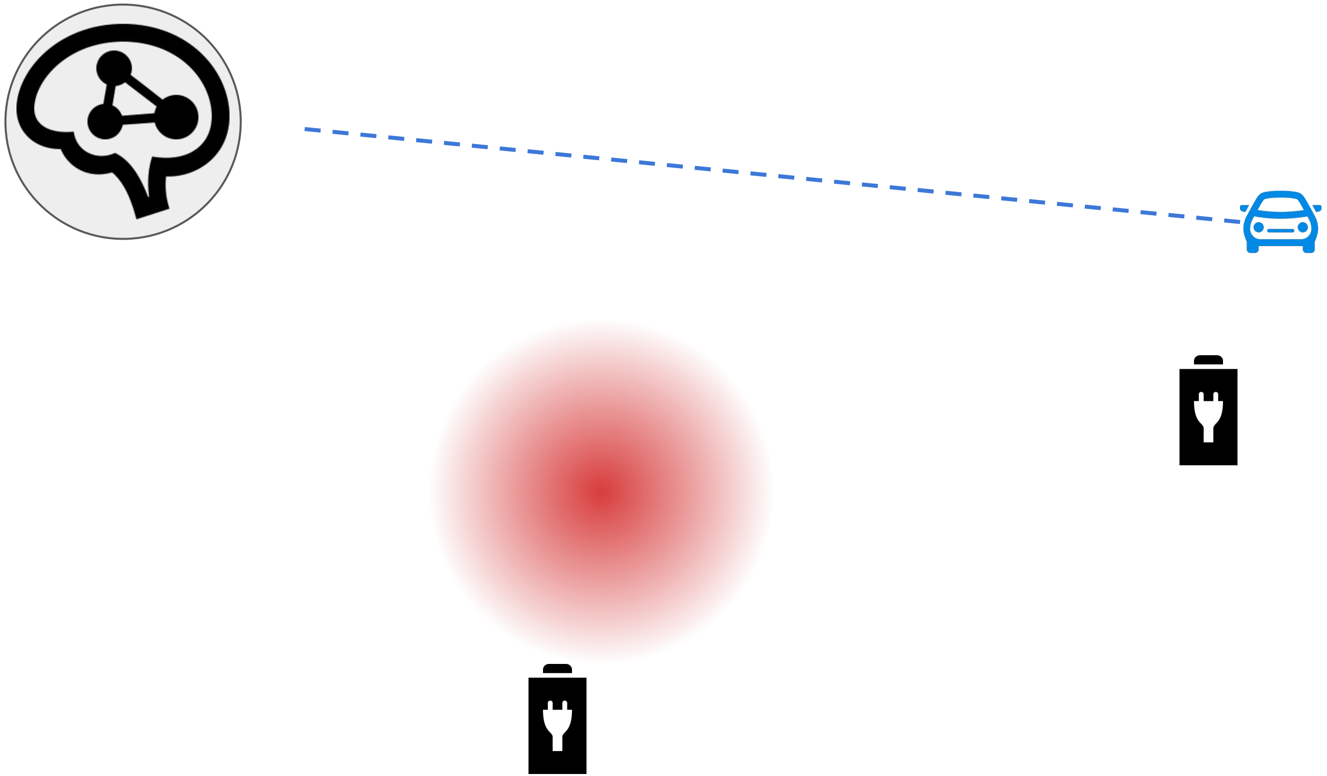
How to Control a Fleet of AEVs



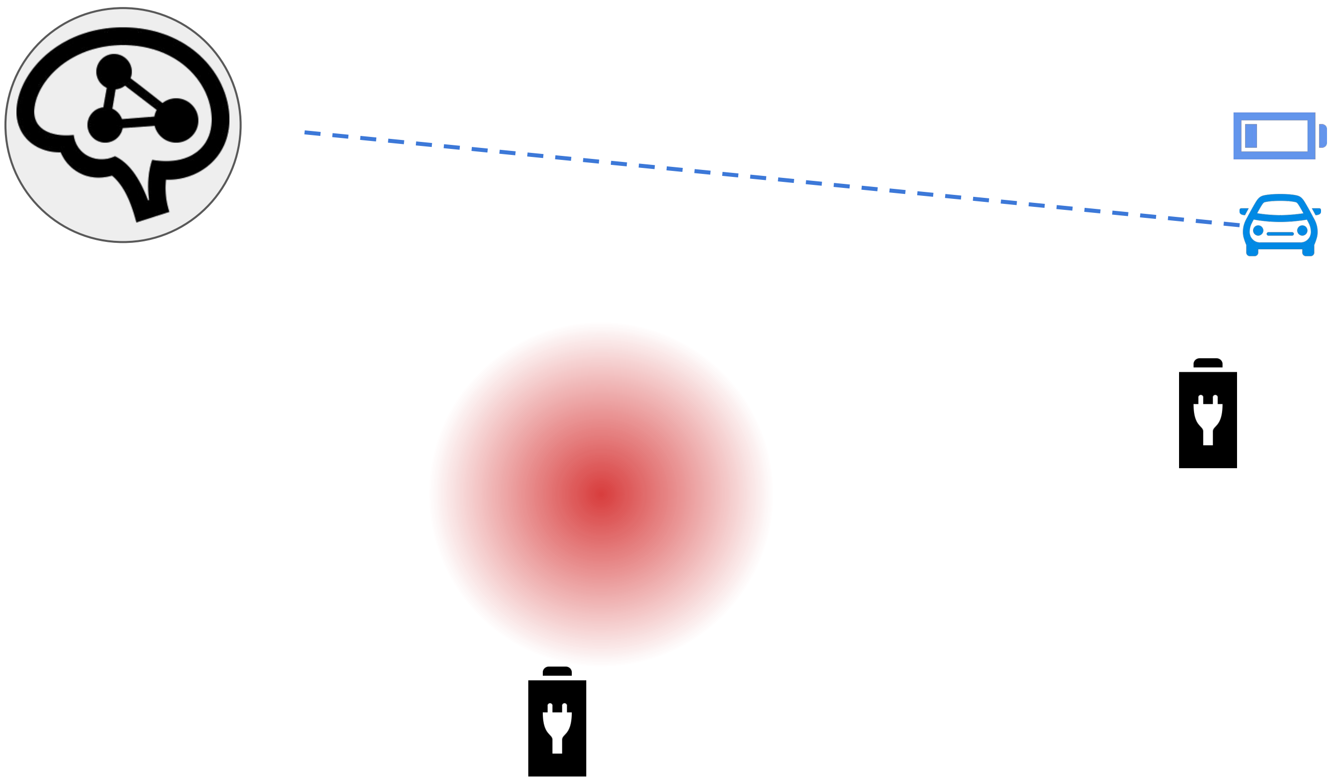
How to Control a Fleet of AEVs



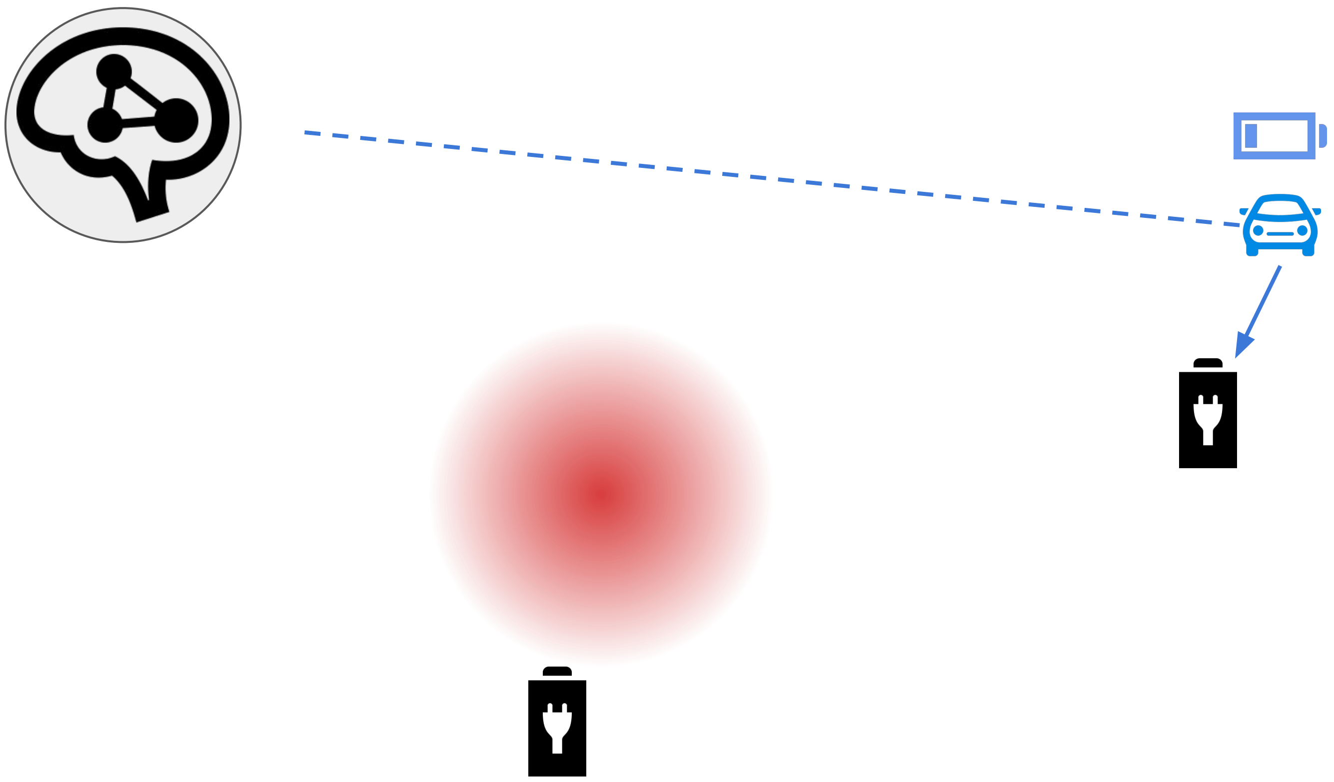
How to Control a Fleet of AEVs



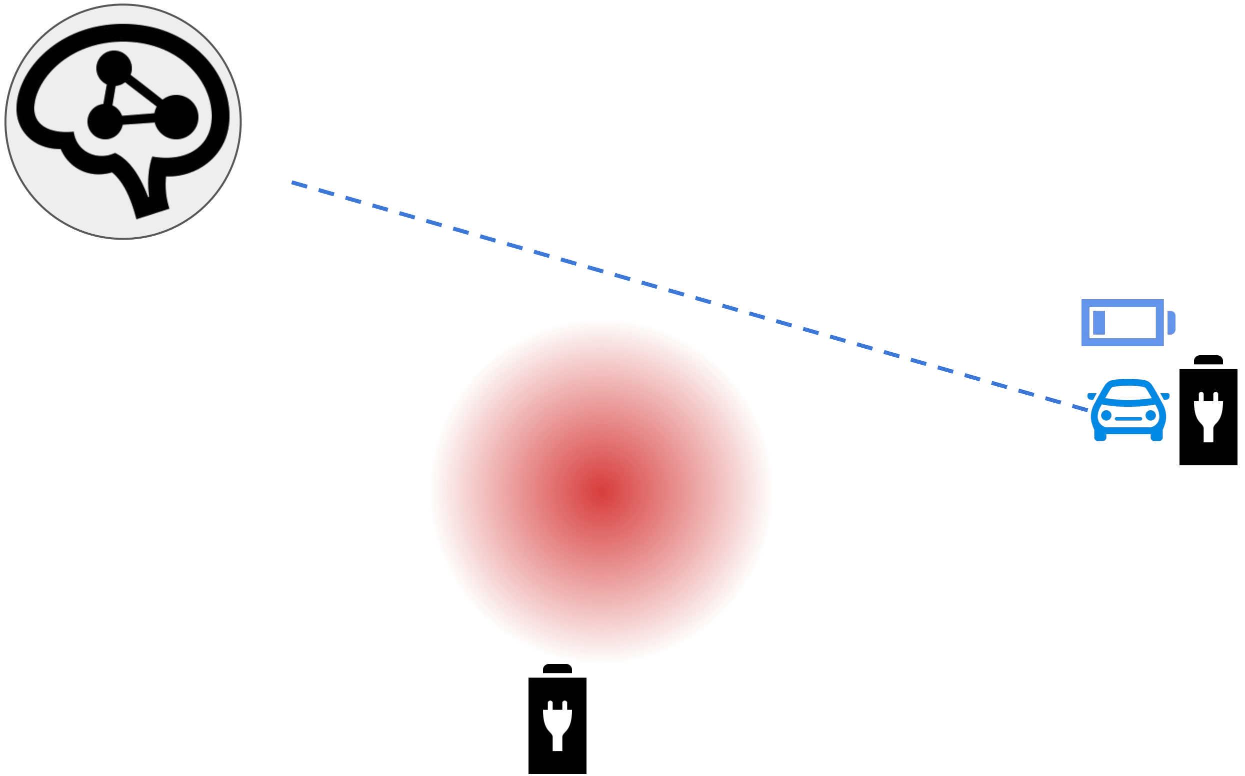
How to Control a Fleet of AEVs



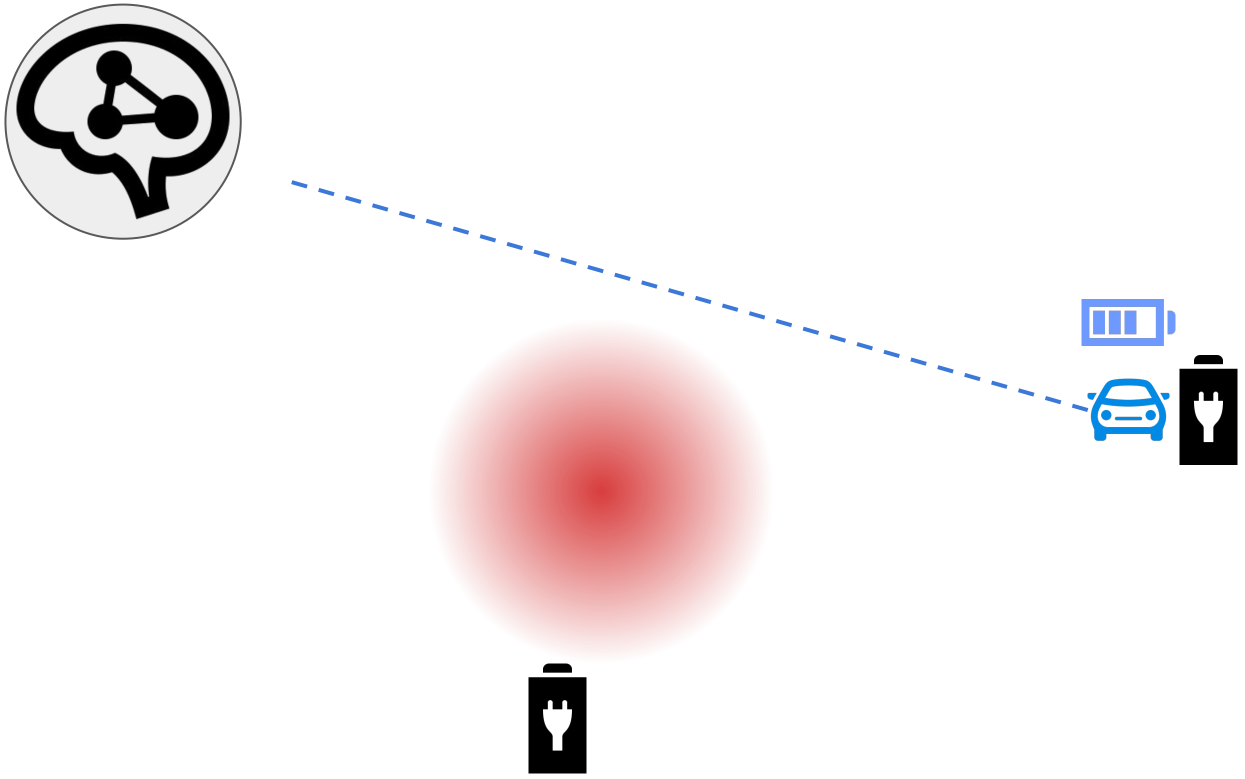
How to Control a Fleet of AEVs



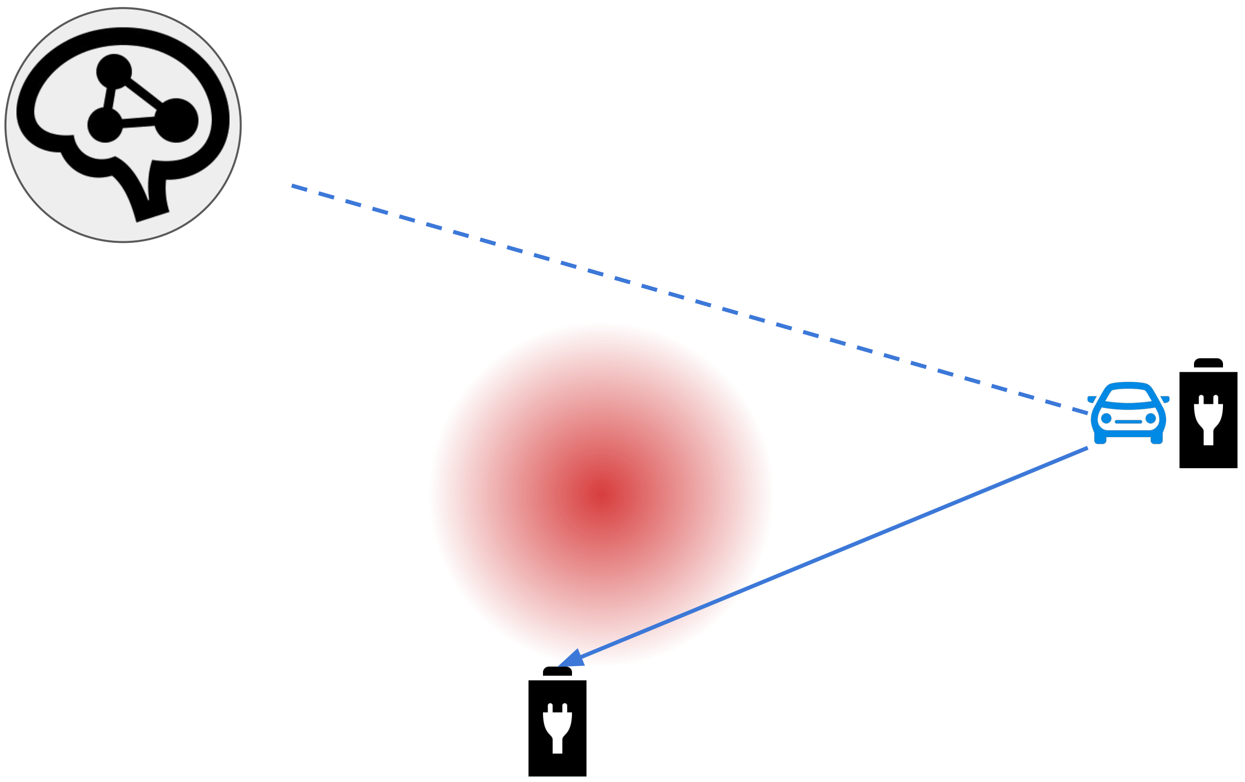
How to Control a Fleet of AEVs



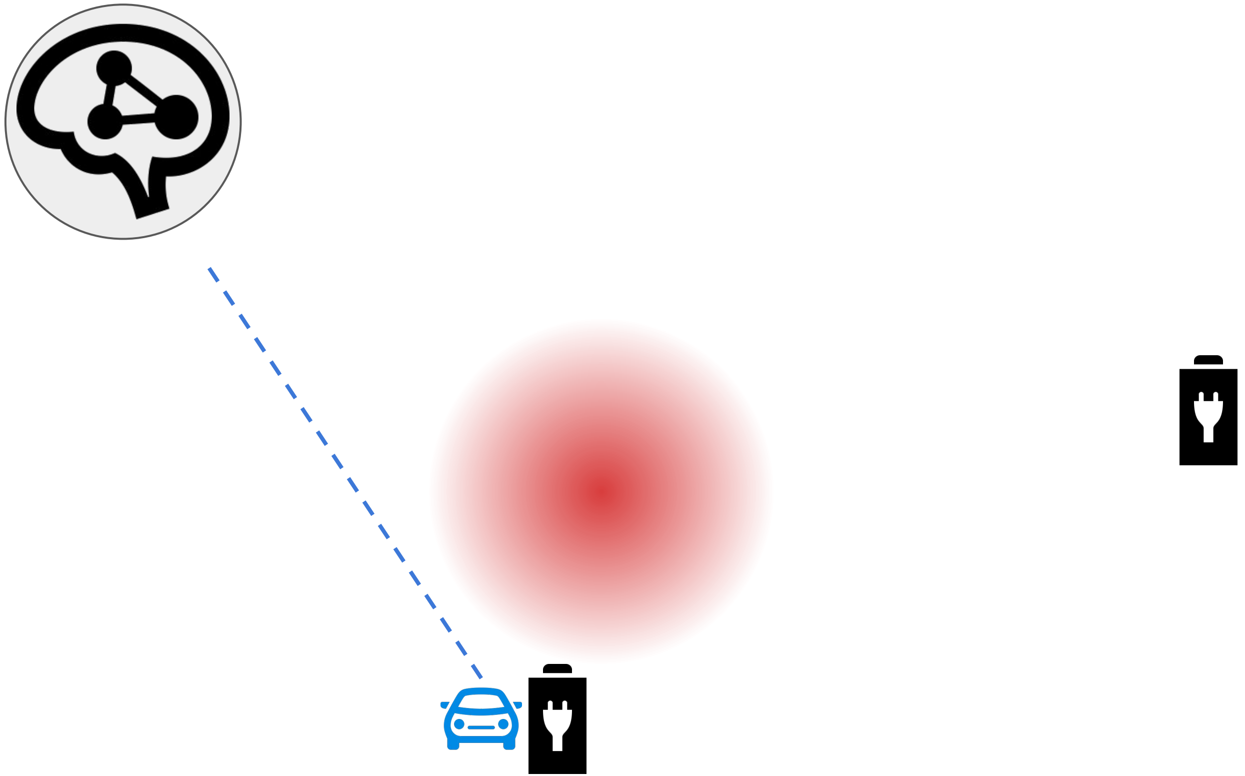
How to Control a Fleet of AEVs



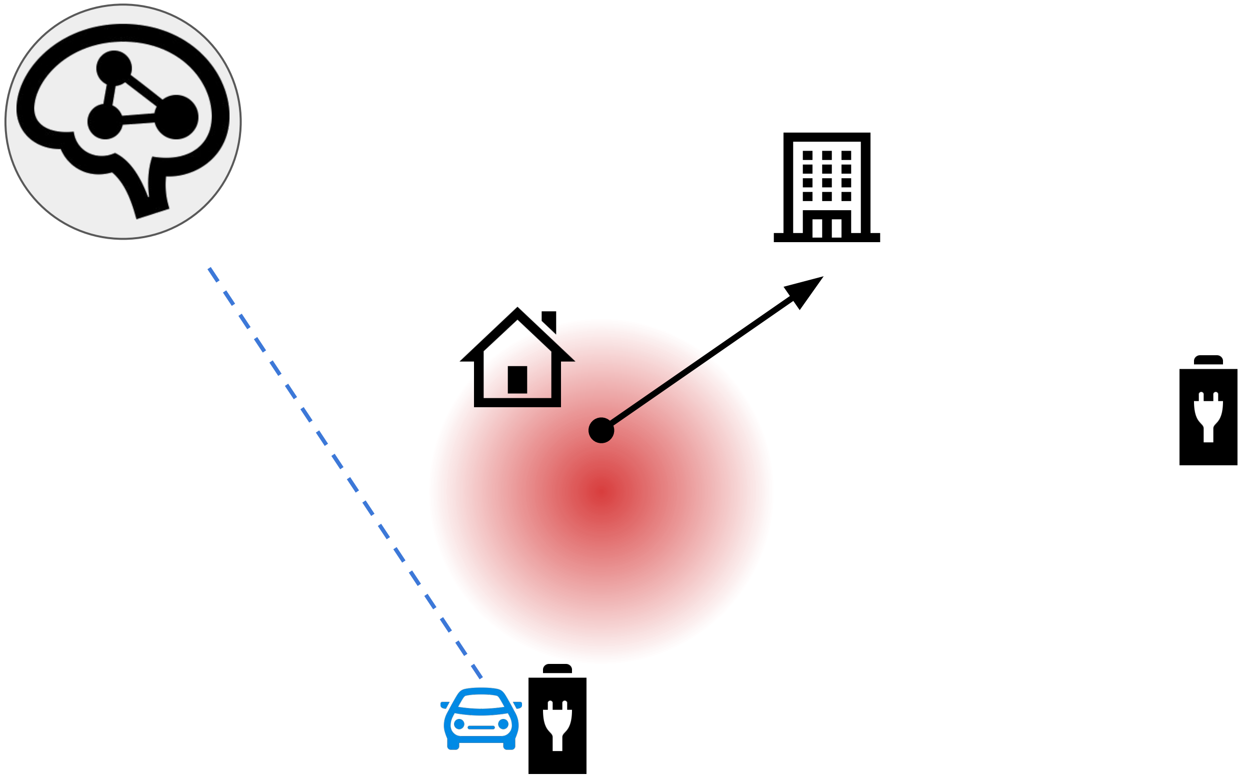
How to Control a Fleet of AEVs



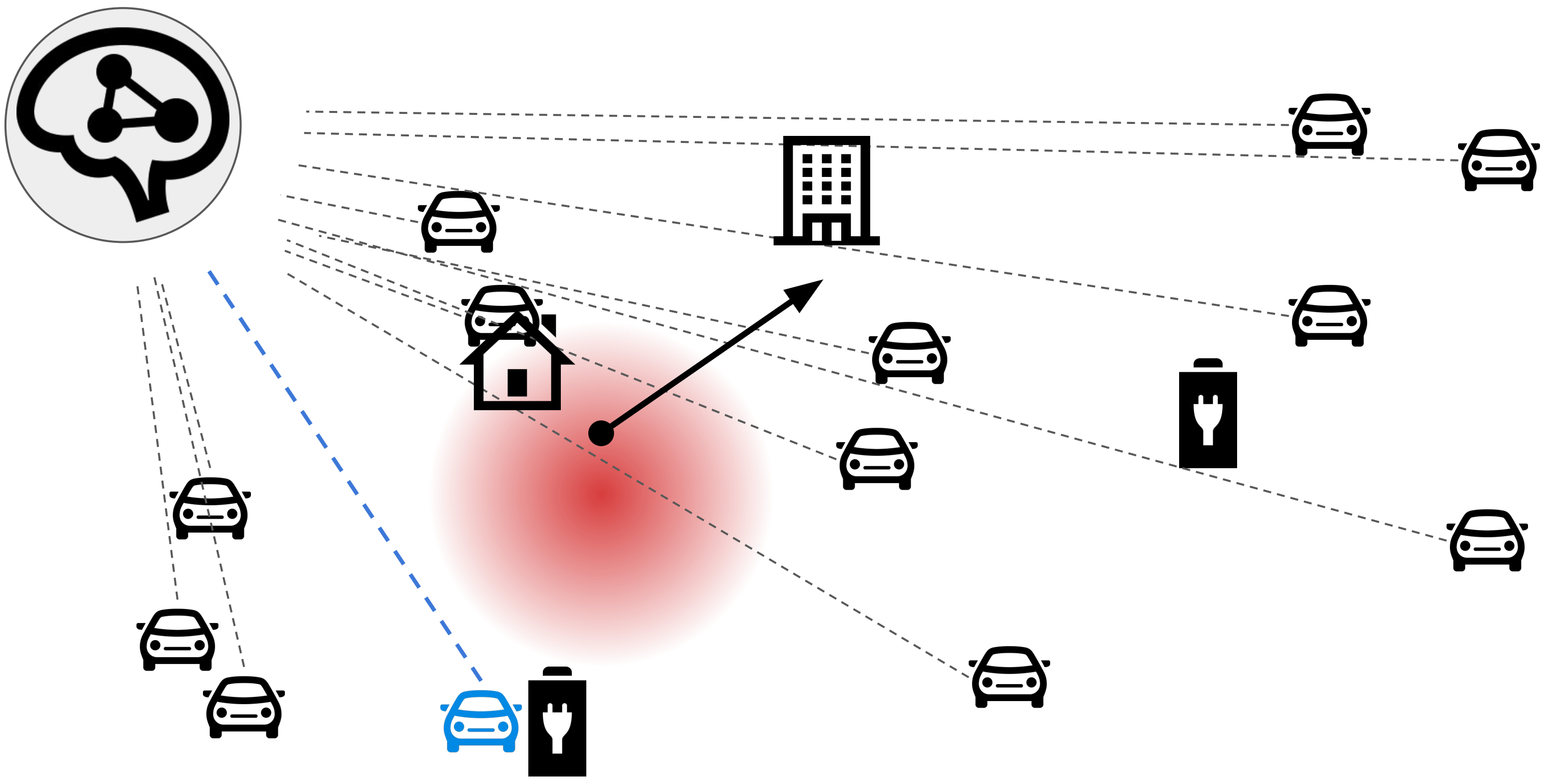
How to Control a Fleet of AEVs



How to Control a Fleet of AEVs



How to Control a Fleet of AEVs

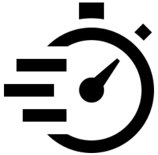


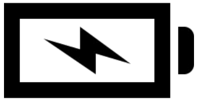
How to Control a Fleet of AEVs

- Real time decision making
- Electrification constraints
- Realistic instances

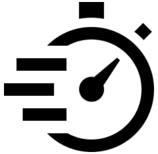
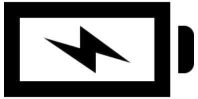
How to Control a Fleet of AEVs

Past Studies





How to Control a Fleet of AEVs

Past Studies			
Al-Kanj et al. (2018)	✅-	✅	X*

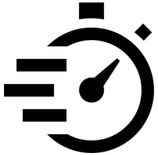
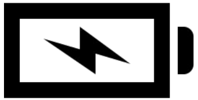
How to Control a Fleet of AEVs

Past Studies

Al-Kanj et al. (2018)	✔-	✔	X*
Holler et al. (2018)	✔	X	✔-

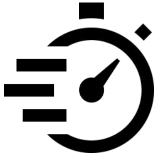
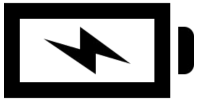
How to Control a Fleet of AEVs

Past Studies

			
Al-Kanj et al. (2018)	✔-	✔	X*
Holler et al. (2018)	✔	X	✔-
Bertsimas et al. (2019)	✔-	X	✔+

How to Control a Fleet of AEVs

Past Studies

			
Al-Kanj et al. (2018)	✔ -	✔	X*
Holler et al. (2018)	✔	X	✔ -
Bertsimas et al. (2019)	✔ -	X	✔ +
Hyland & Mahmassani (2018)	✔	X	✔ -

How to Control a Fleet of AEVs

Past Studies

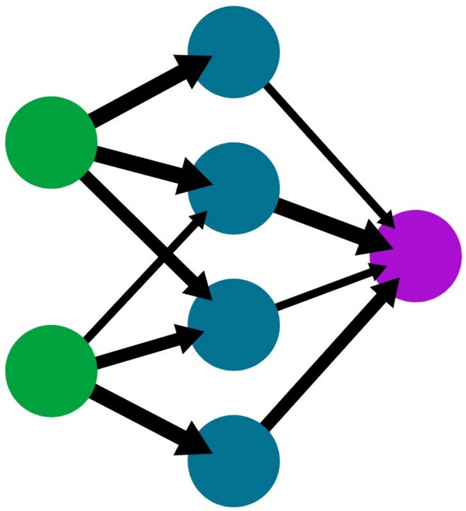
Al-Kanj et al. (2018)	✓-	✓	X*
Holler et al. (2018)	✓	X	✓-
Bertsimas et al. (2019)	✓-	X	✓+
Hyland & Mahmassani (2018)	✓	X	✓-
Kullman et al. (2019)	✓	✓	✓

Model & Methods

Methods combine Deep RL and OR

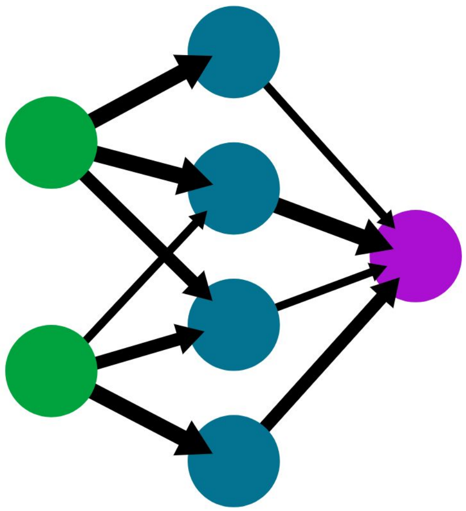
Methods combine Deep RL and OR

Dynamic problem

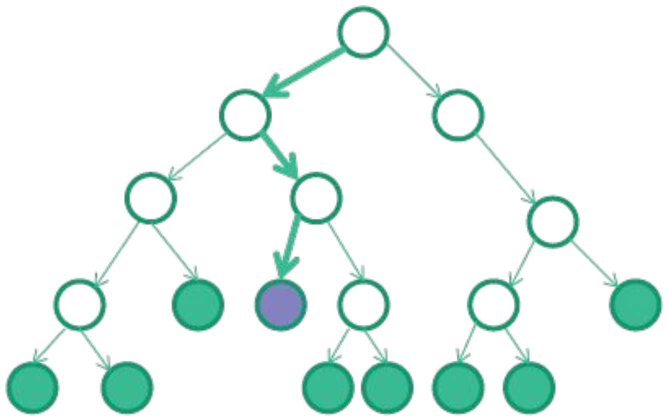


Methods combine Deep RL and OR

Dynamic problem

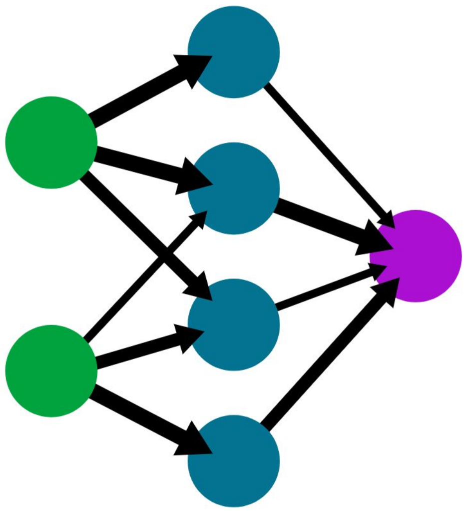


Static Bound



Methods combine Deep RL and OR

Dynamic problem

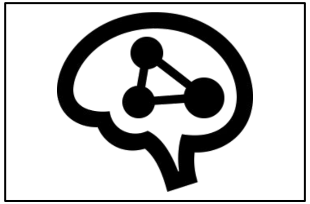


Static Bound

Dynamic Problem Model: MDP

Dynamic Problem Model: MDP

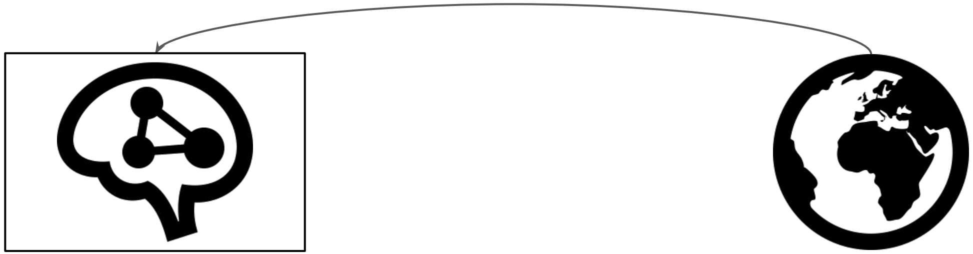
Agent



Environment

Dynamic Problem Model: MDP

State



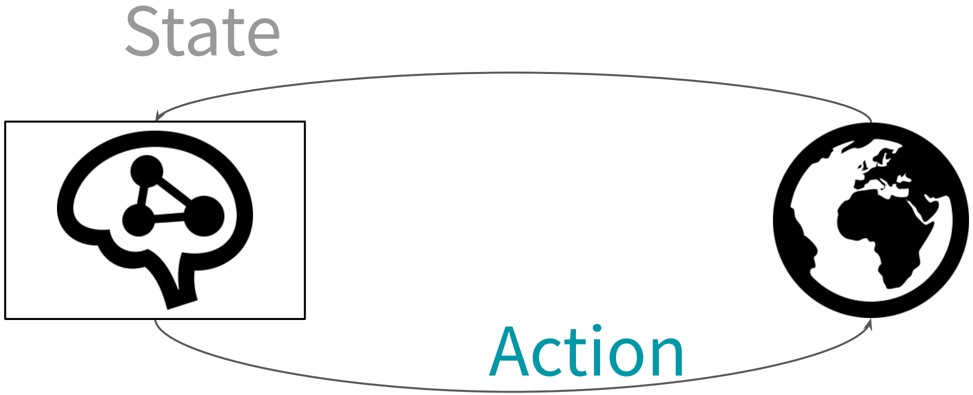
Time

Vehicles':

- Positions
- Charges
- Scheduled jobs

Request

Dynamic Problem Model: MDP



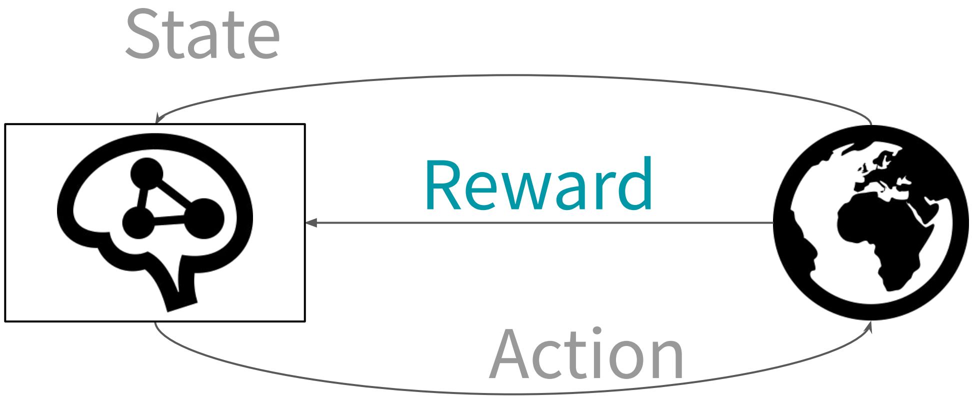
Assign vehicle to new request

For each vehicle:

  New reposition/recharge instructions

Depends on state

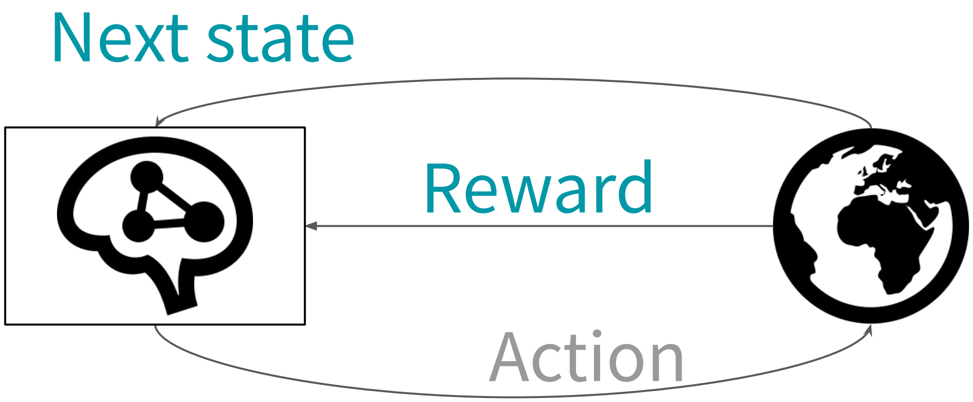
Dynamic Problem Model: MDP



Value of request, if served

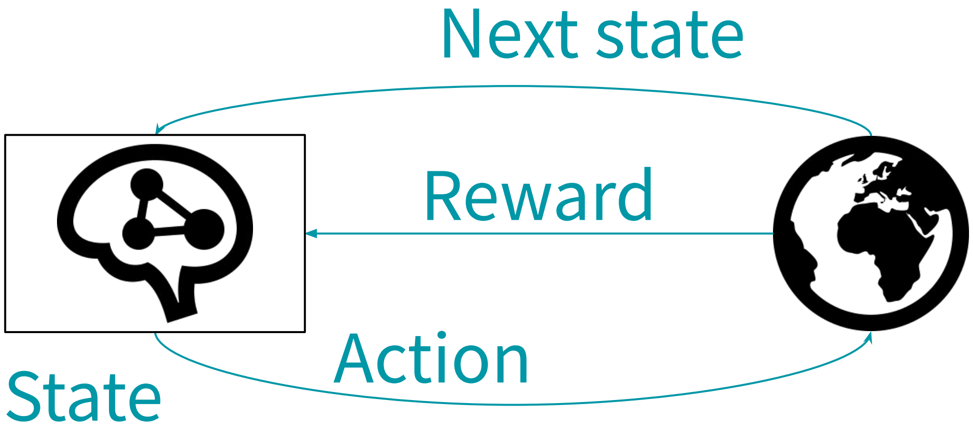
Fixed + distance-dependent

Dynamic Problem Model: MDP



Update state

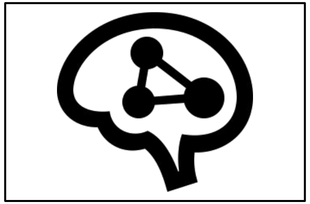
Dynamic Problem Model: MDP



Repeat until episode's terminal state:

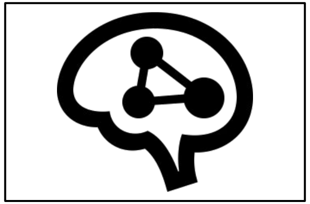
When time horizon reached

Dynamic Problem Model: MDP



Objective

Dynamic Problem Model: MDP



Objective

Find agent/policy maximizing E[sum of rewards]

Agents

Agents

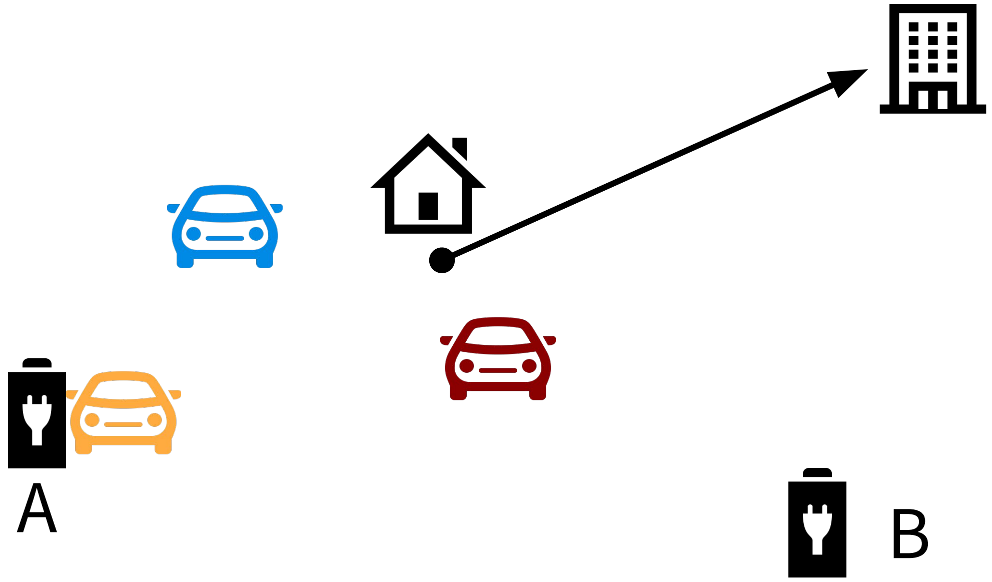
1. Random

Agents

1. Random
2. Nearest (heuristic)

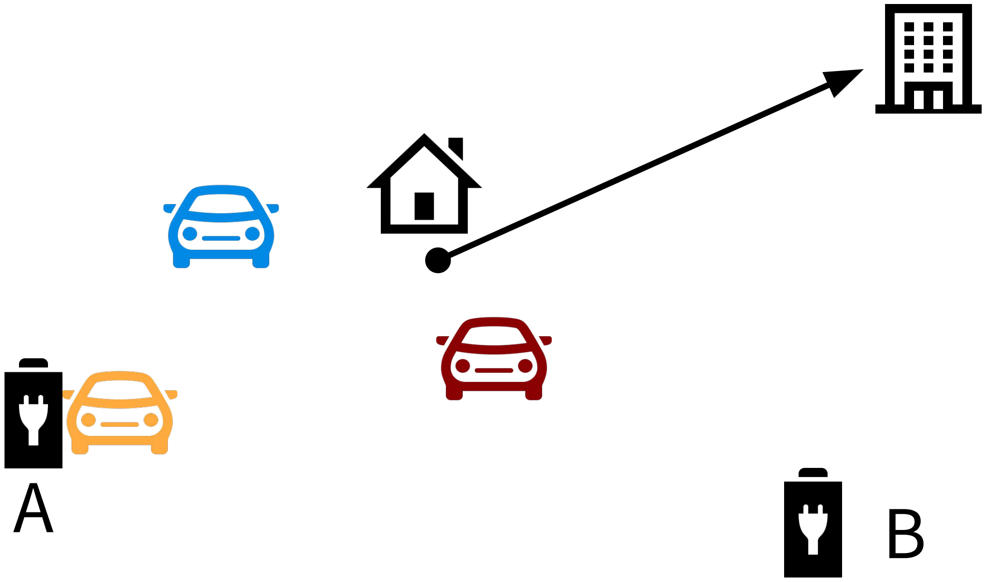
Agents

1. Random
2. Nearest



Agents

1. Random
2. Nearest

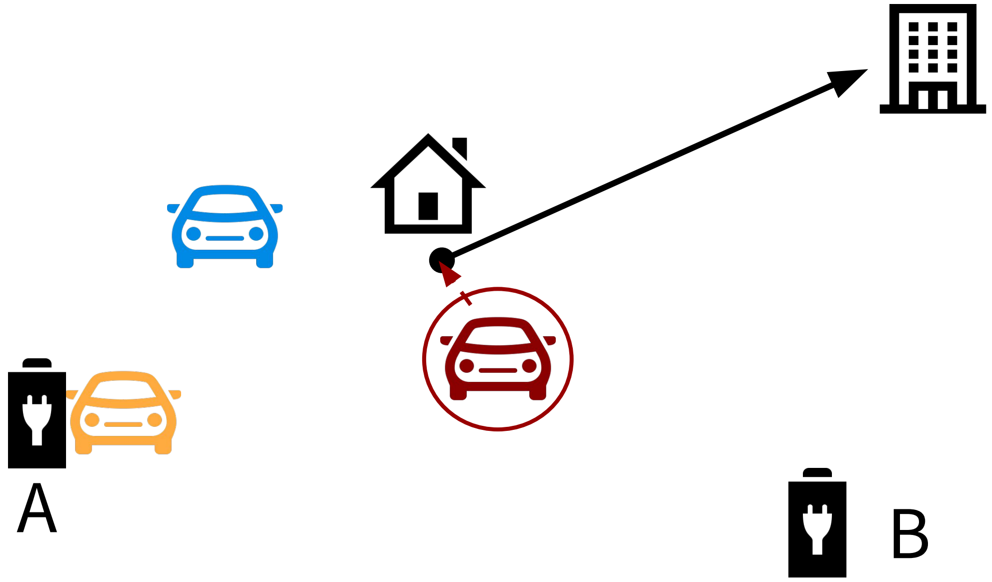


Assignment

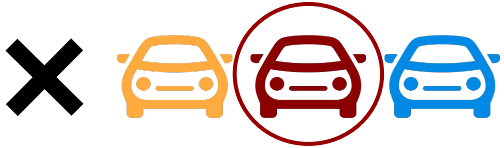
Reposition

Agents

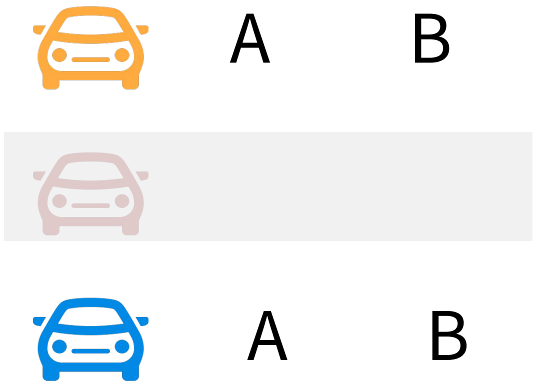
1. Random
2. Nearest



Assignment

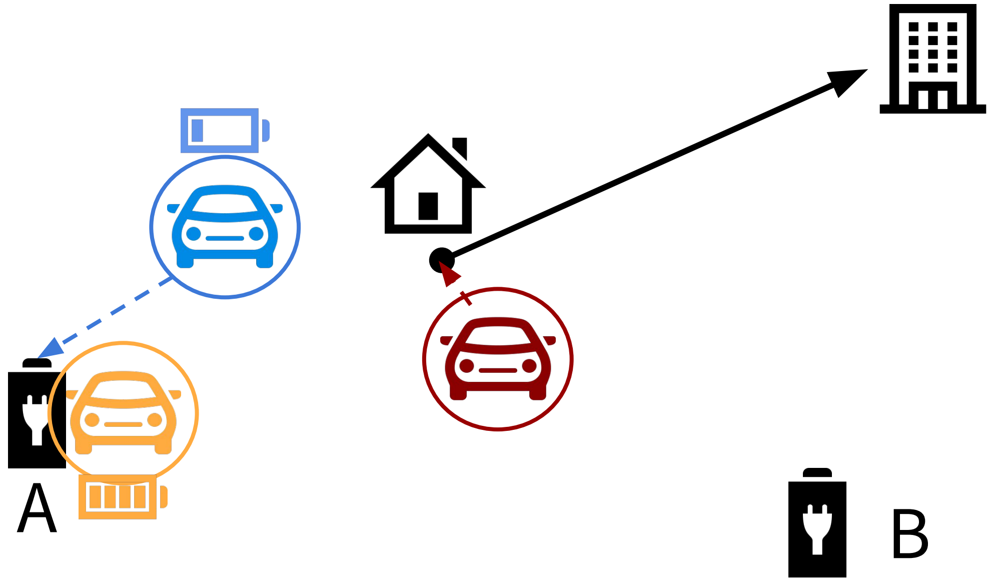


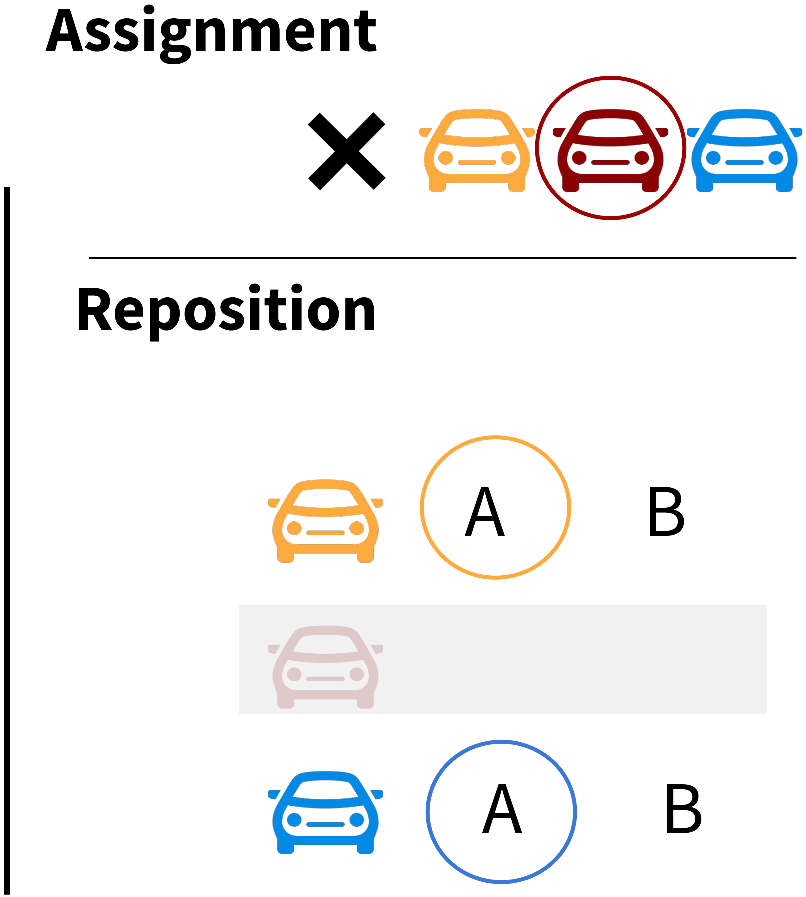
Reposition



Agents

1. Random
2. Nearest



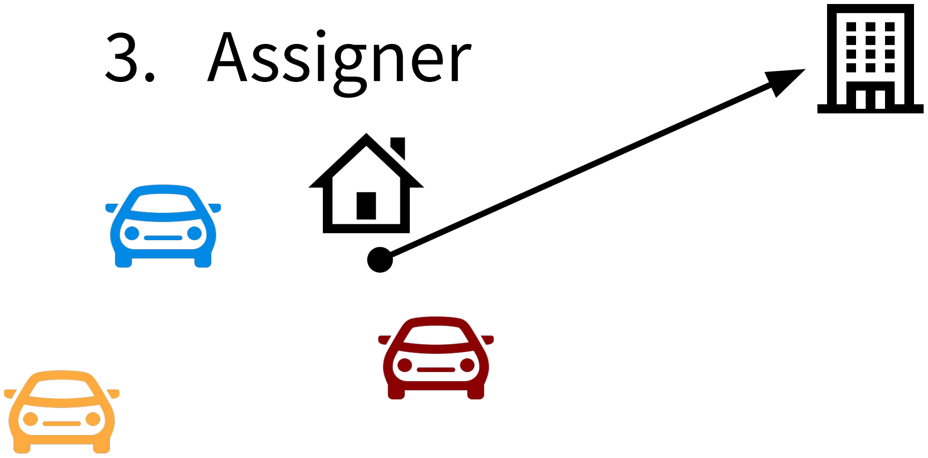


Agents

1. Random
2. Nearest
3. Assigner

Agents

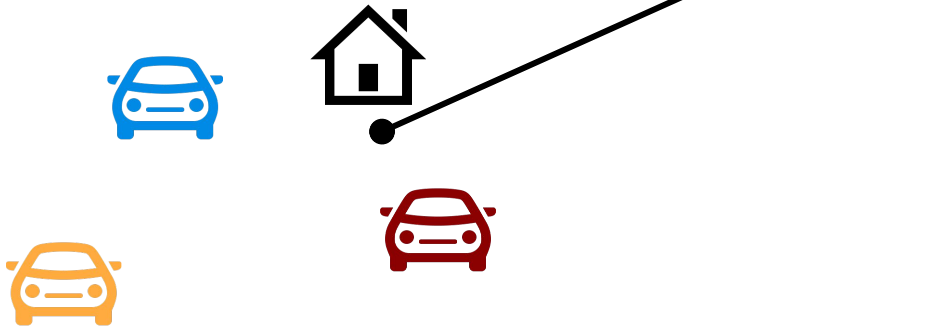
1. Random
2. Nearest
3. Assigner



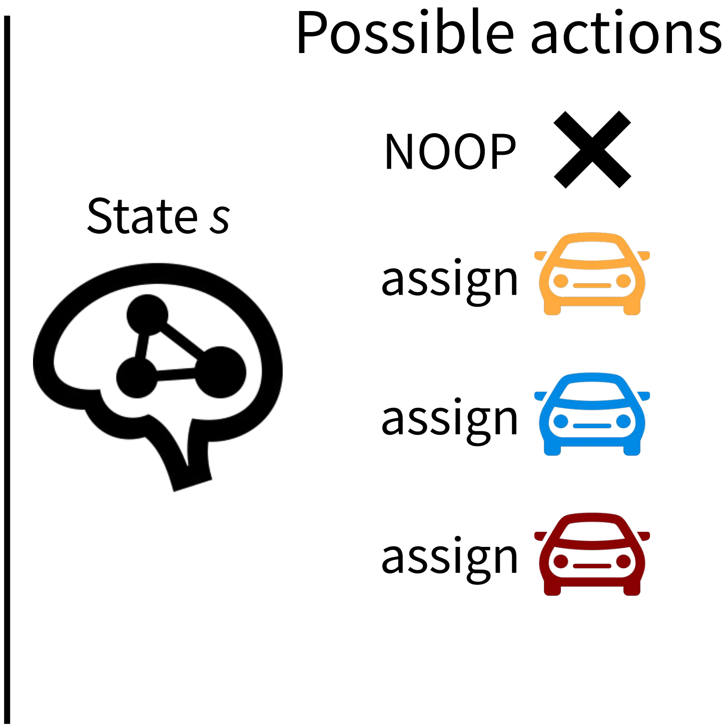
Assignment

Agents

1. Random
2. Nearest
3. Assigner

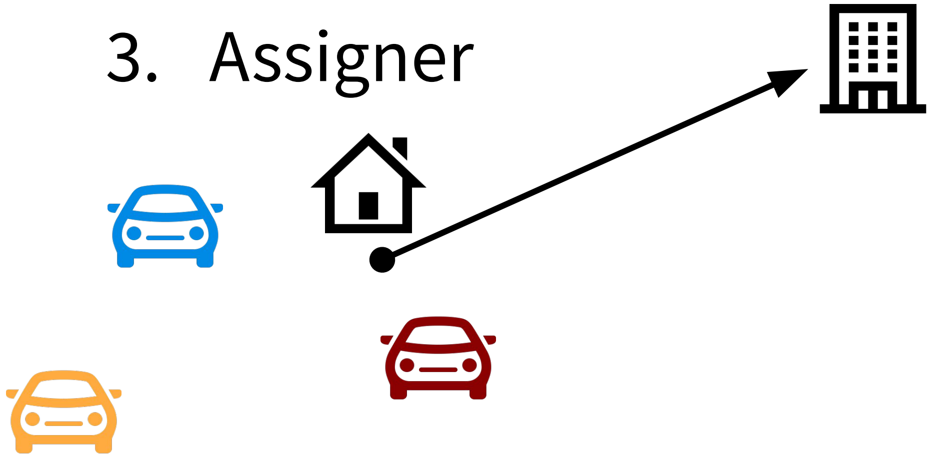


Assignment



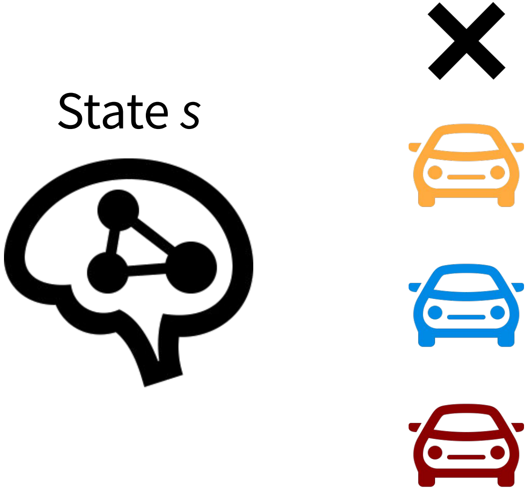
Agents

1. Random
2. Nearest
3. Assigner



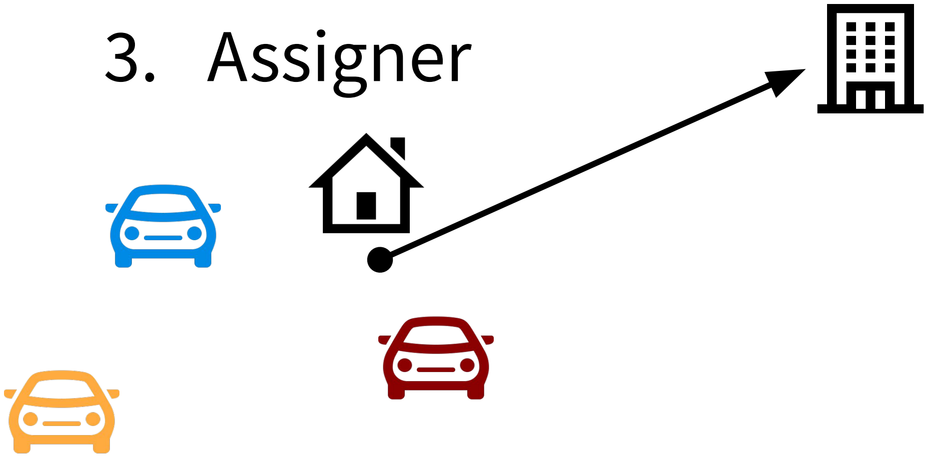
Assignment

Value of action?



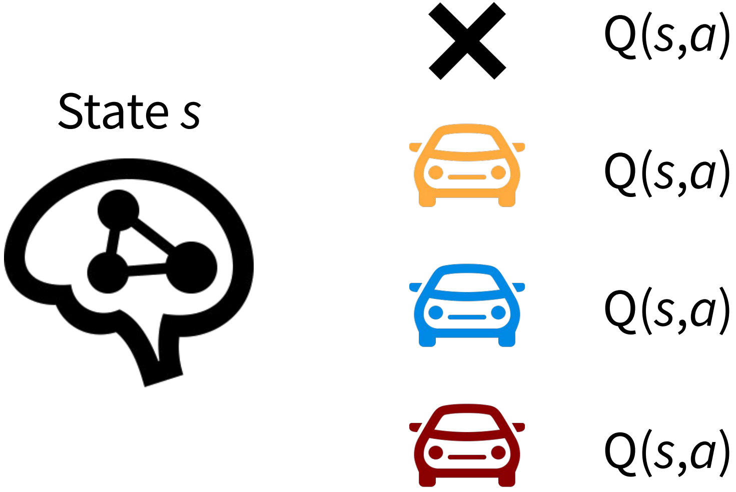
Agents

1. Random
2. Nearest
3. Assigner



Assignment

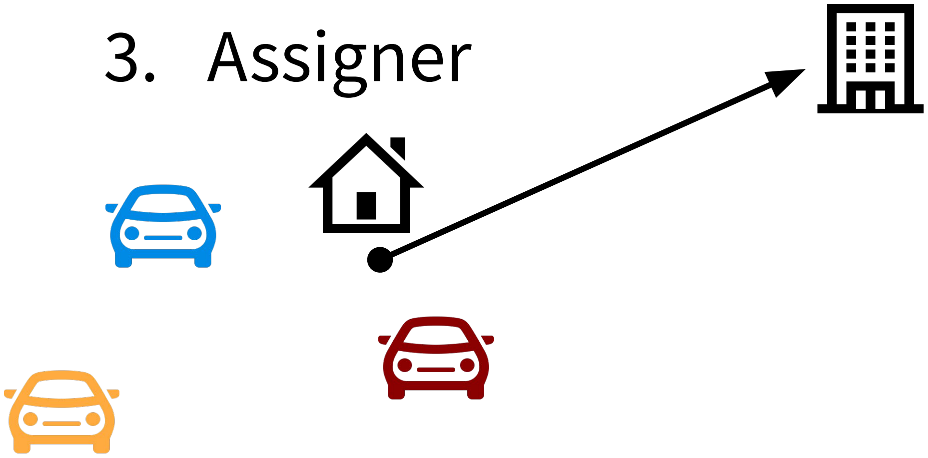
Value of action?



Immediate reward + reward-to-go

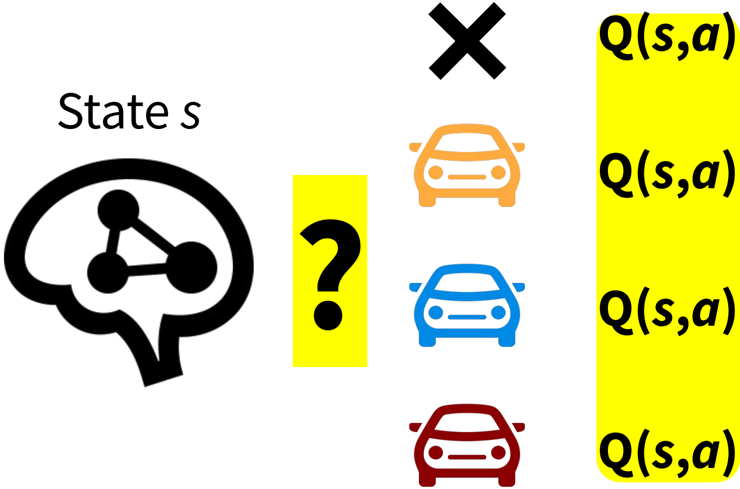
Agents

1. Random
2. Nearest
3. Assigner



Assignment

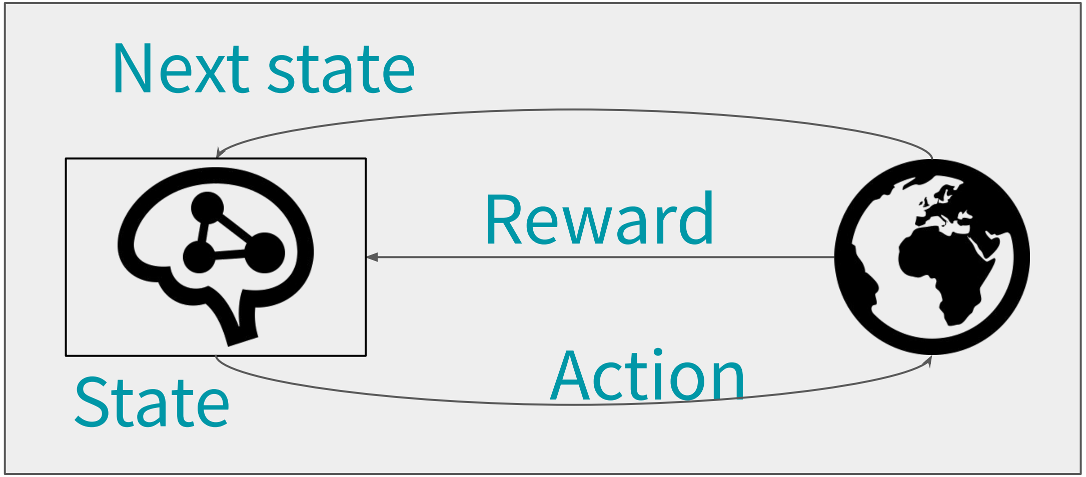
Value of action?



How to learn Q(,)?

Agents

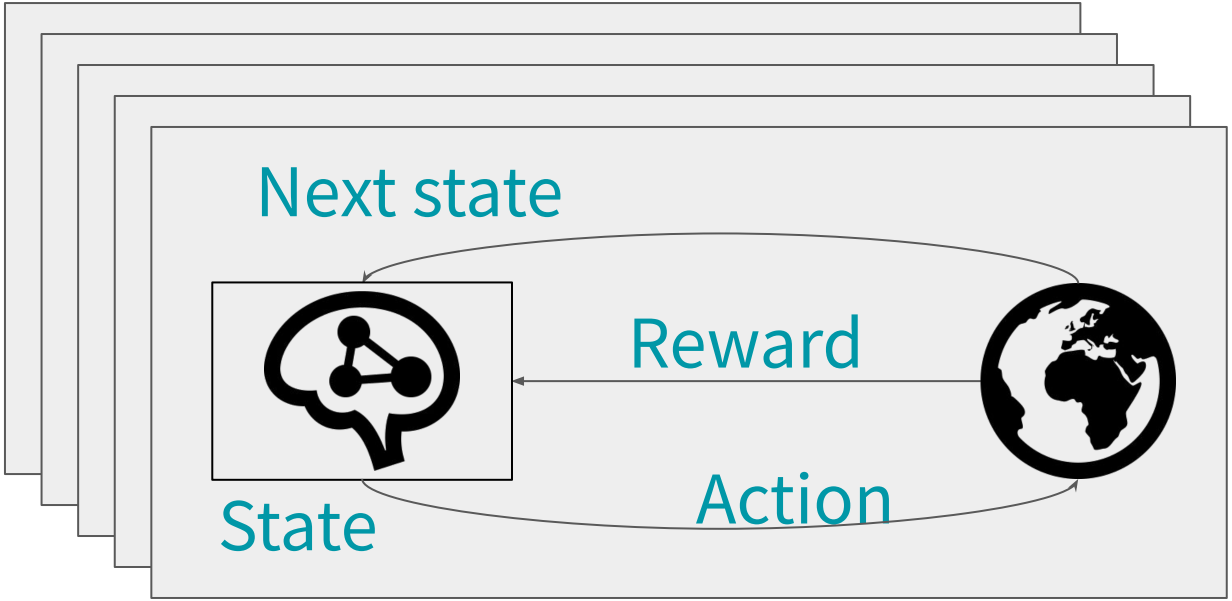
1. Random
2. Nearest
3. Assigner



How does it learn?

Agents

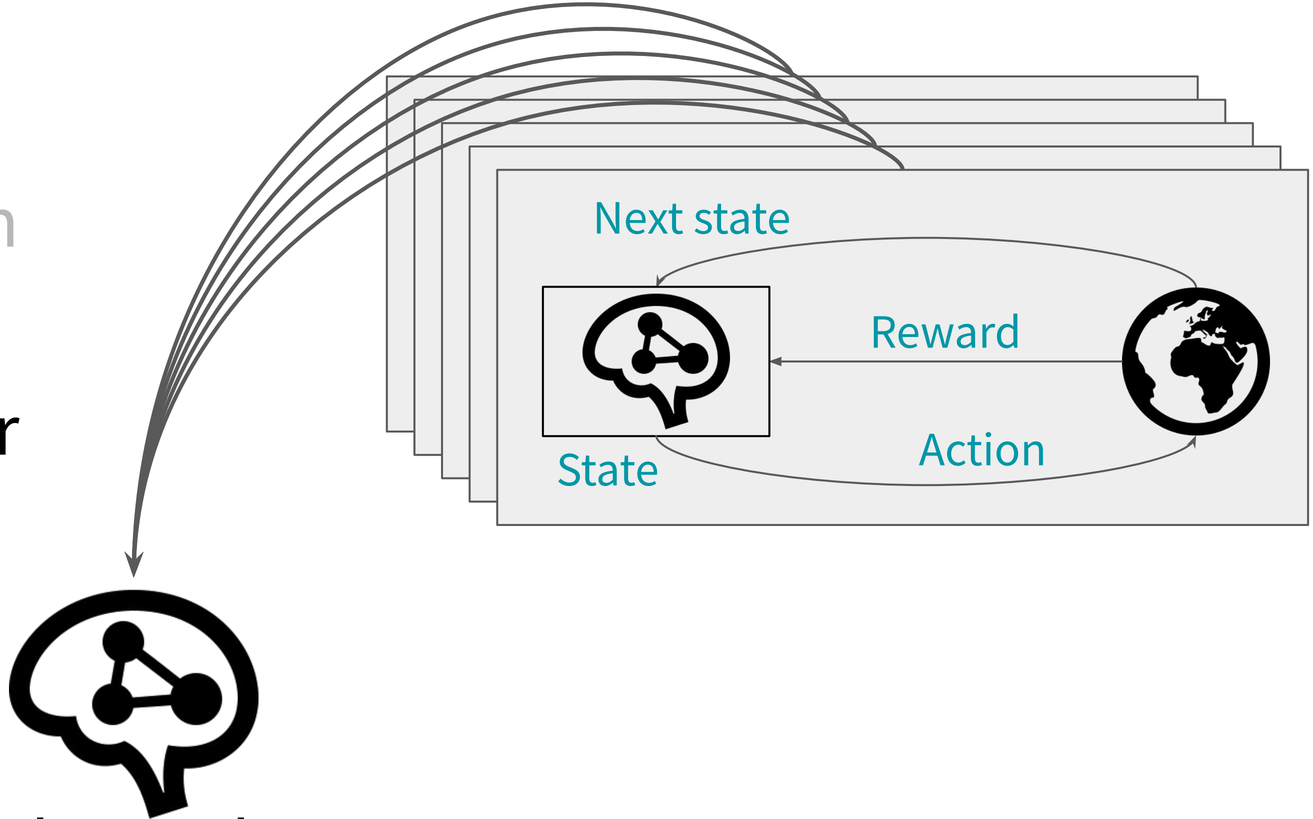
1. Random
2. Nearest
3. Assigner



How does it learn?

Agents

1. Random
2. Nearest
3. Assigner



How does it learn?

Agents

1. Random
2. Nearest
3. Assigner

Architecture

Agents

1. Random
2. Nearest
3. Assigner

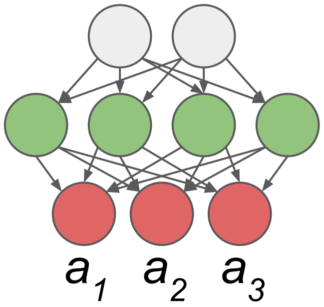
Architecture

DQN

Input s

"Hidden"

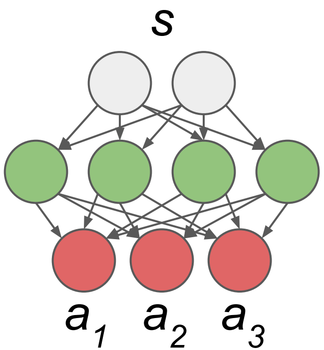
$Q(s, a)$



Agents

1. Random
2. Nearest
3. Assigner

Architecture



+

Hessel, Matteo, et al. "Rainbow: Combining improvements in deep reinforcement learning." *Thirty-Second AAAI Conference on Artificial Intelligence*. 2018.

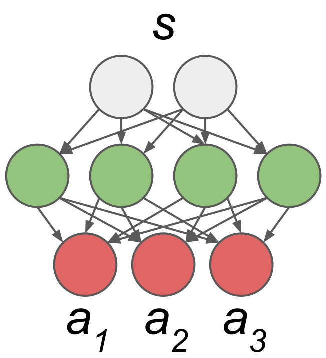
Agents

1. Random
2. Nearest
3. Assigner
4. Full Control

Agents

1. Random
2. Nearest
3. Assigner
4. Full Control

Architecture



Agents

1. Random
2. Nearest
3. Assigner
4. Full Control

State s

For each vehicle:

Value of action...

$Q(s,a)$	**Do Nothing**
$Q(s,a)$	**Serve**
$Q(s,a)$	**Go To First CS**
$Q(s,a)$	**...**
$Q(s,a)$	**Go to Last CS**

Agents

1. Random
2. Nearest
3. Assigner
4. Full Control

State s

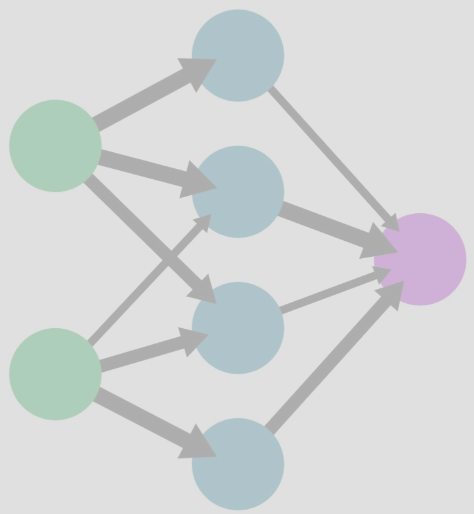
Per-vehicle Q-values improves scalability

For each vehicle:
Value of action...

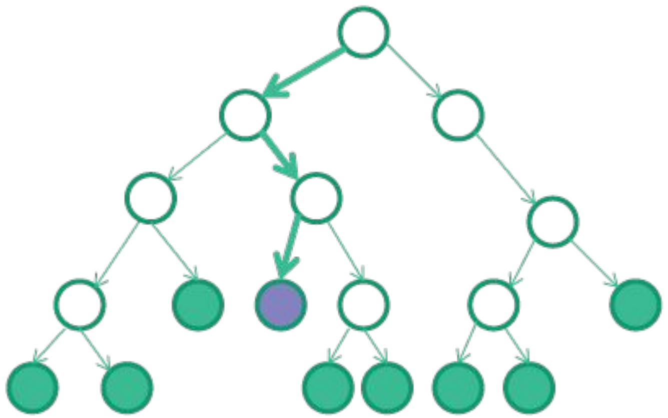
$Q(s,a)$	**Do Nothing**
$Q(s,a)$	**Serve**
$Q(s,a)$	**Go To First CS**
$Q(s,a)$	**...**
$Q(s,a)$	**Go to Last CS**

Methods combine Deep RL and OR

Dynamic problem



Static Bound



Static Problem

Static Problem

Perfect Information

Perfect Info: OR tools

Perfect Info: Benders decomposition

Perfect Info: Benders decomposition

Master problem

Subproblem

Perfect Info: Benders decomposition

Master problem

Assign requests to vehicles

Time feasibility

Subproblem

Perfect Info: Benders decomposition

Master problem

Assign requests to vehicles

Time feasibility

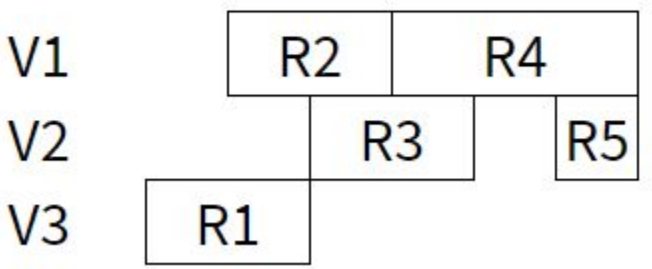
Subproblem

Energy feasibility

Charging decisions

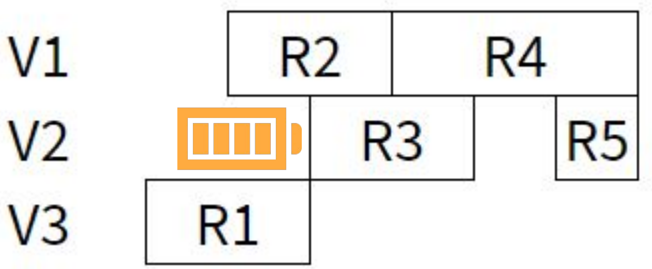
Perfect Info: Benders decomposition

Master problem



Perfect Info: Benders decomposition

Master problem

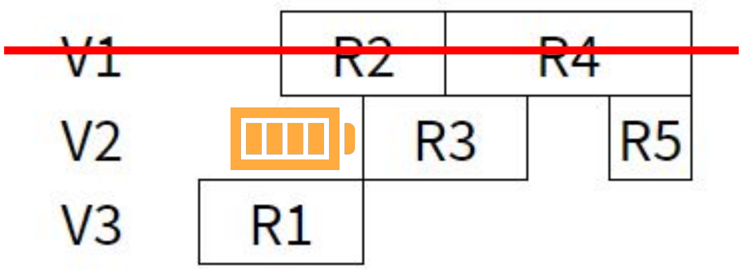


Subproblem

FRVCP: Froger et al. (2018)

Perfect Info: Benders decomposition

Master problem



Subproblem

FRVCP: Froger et al. (2018)

Data & Empirical Results

Data: Manhattan-based instances

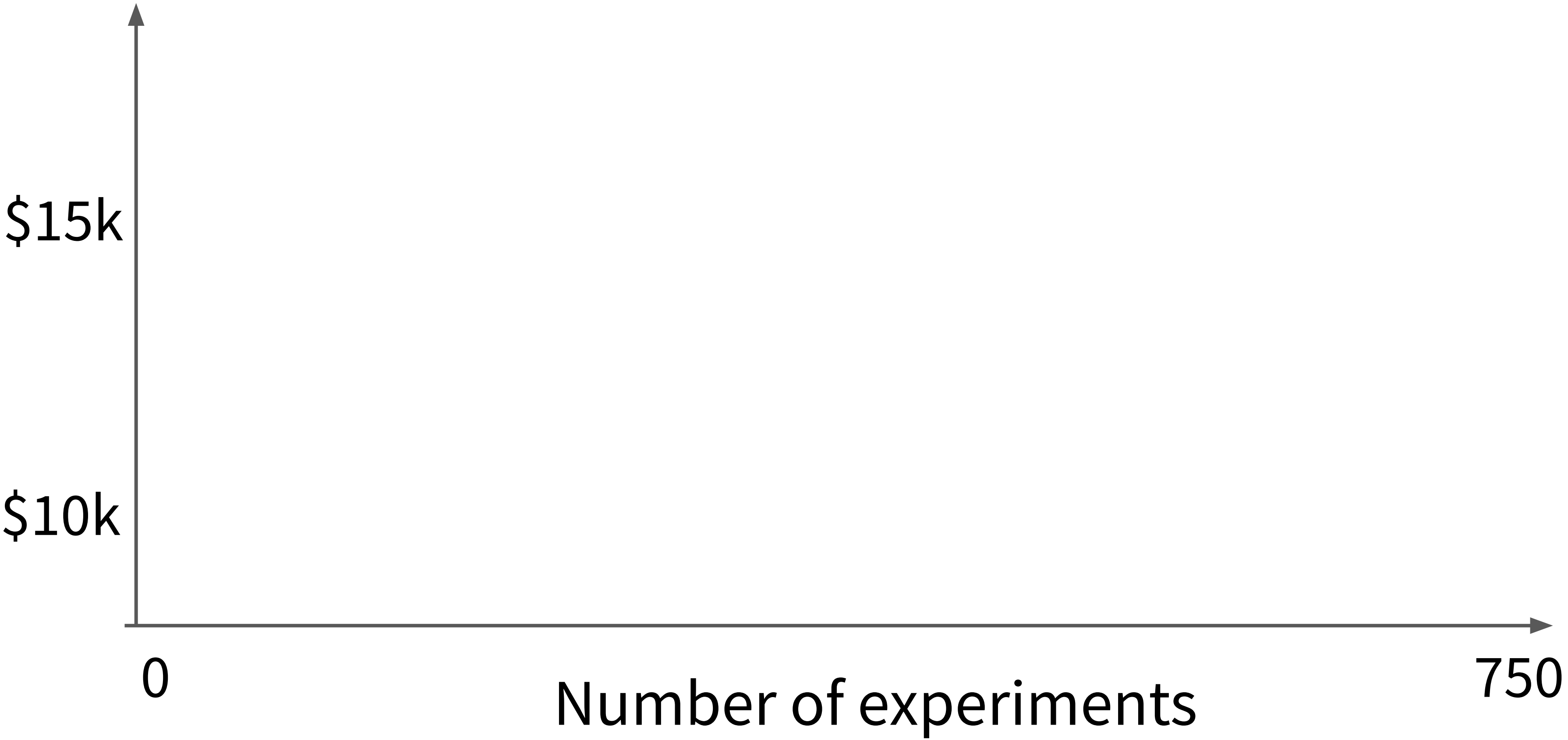
Data: Manhattan-based instances

- Trips:
  - NYC Taxi + ridehail data (2018)
- CSs:
  - All current and planned CSs
- Vehicles:
  - Mid-range Tesla Model 3

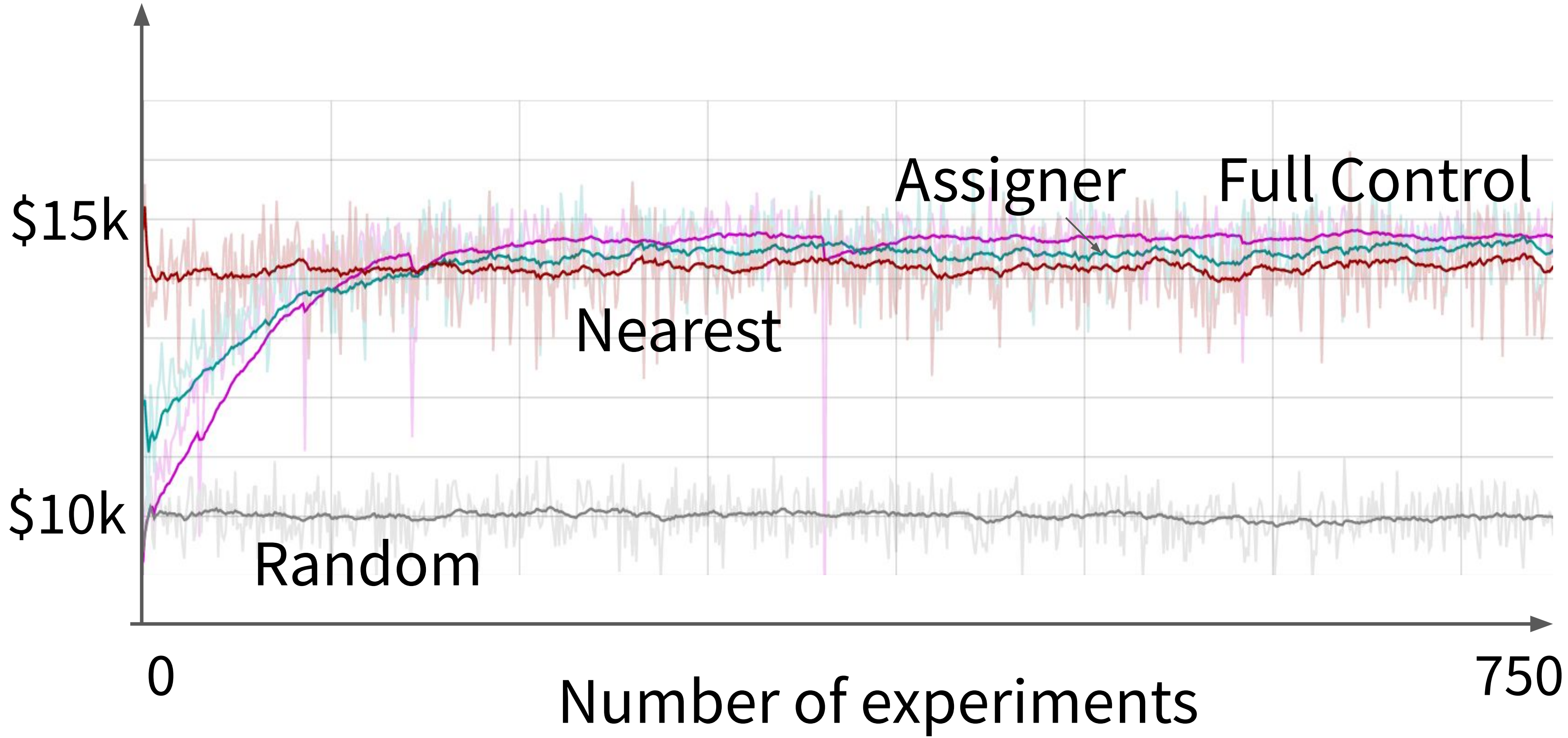
Data: Manhattan-based instances

Trips/day	1400	1400	14,000
Vehicles	43	14	140
Methods	Policies + bound	Policies	Policies - Assigner

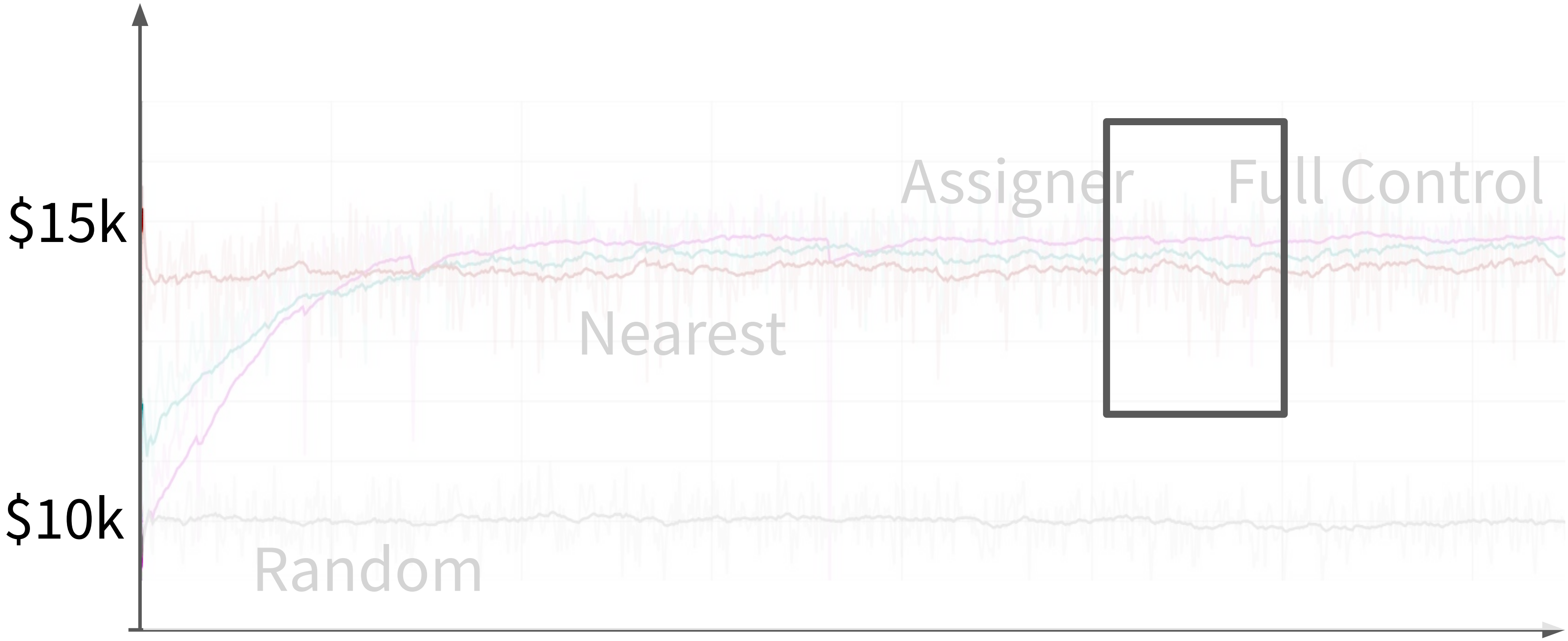
Results (1400/43)



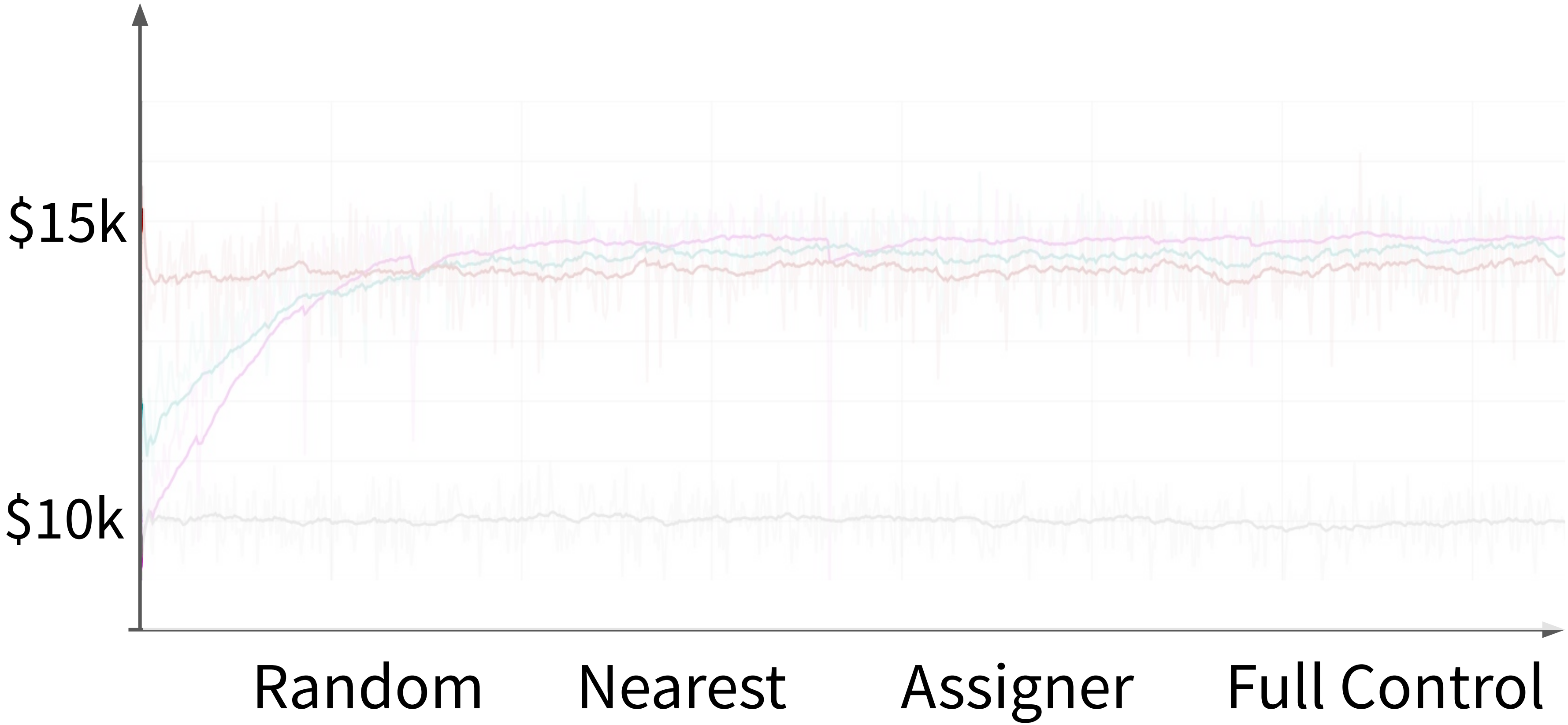
Results (1400/43)



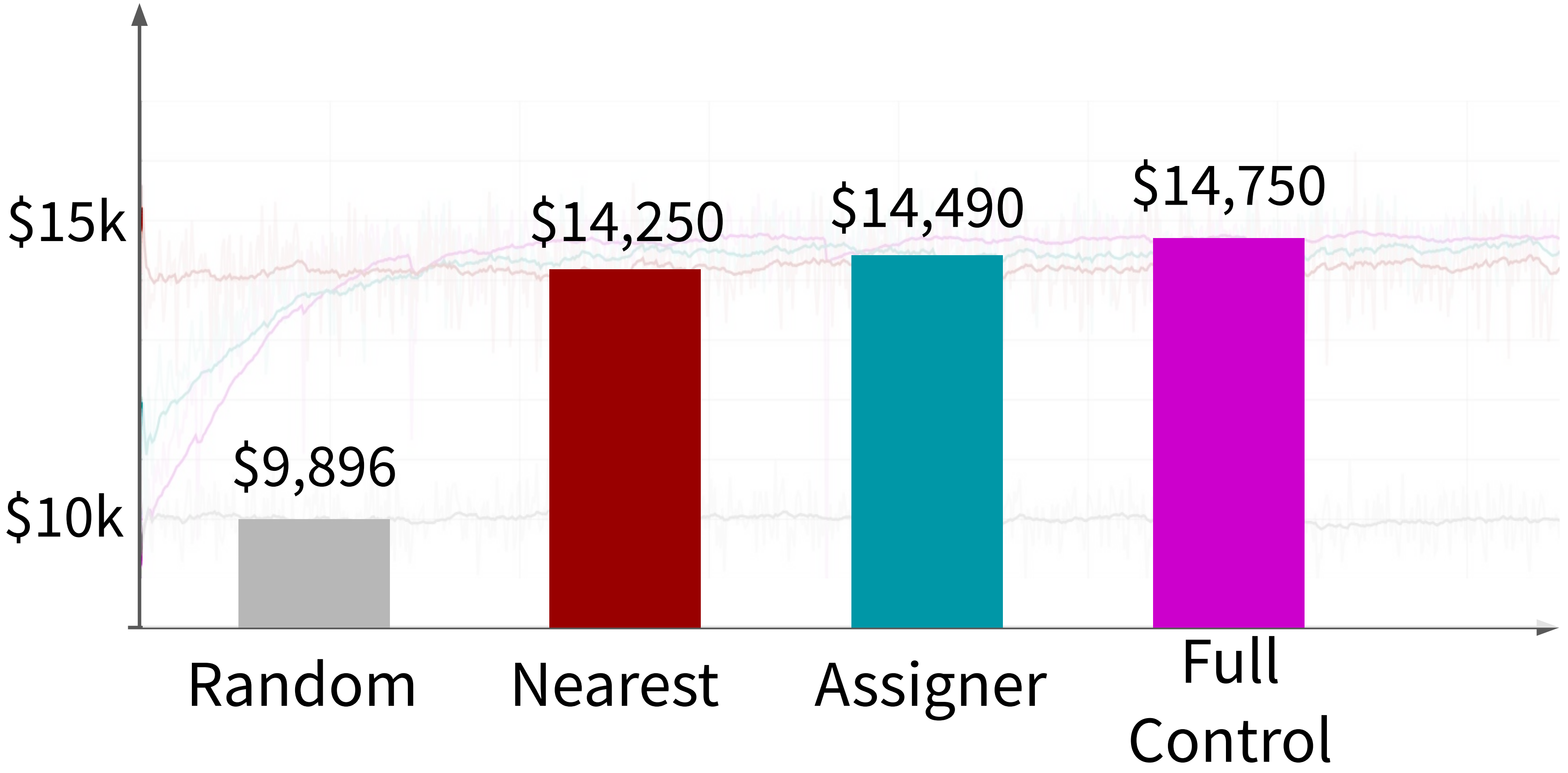
Results (1400/43)



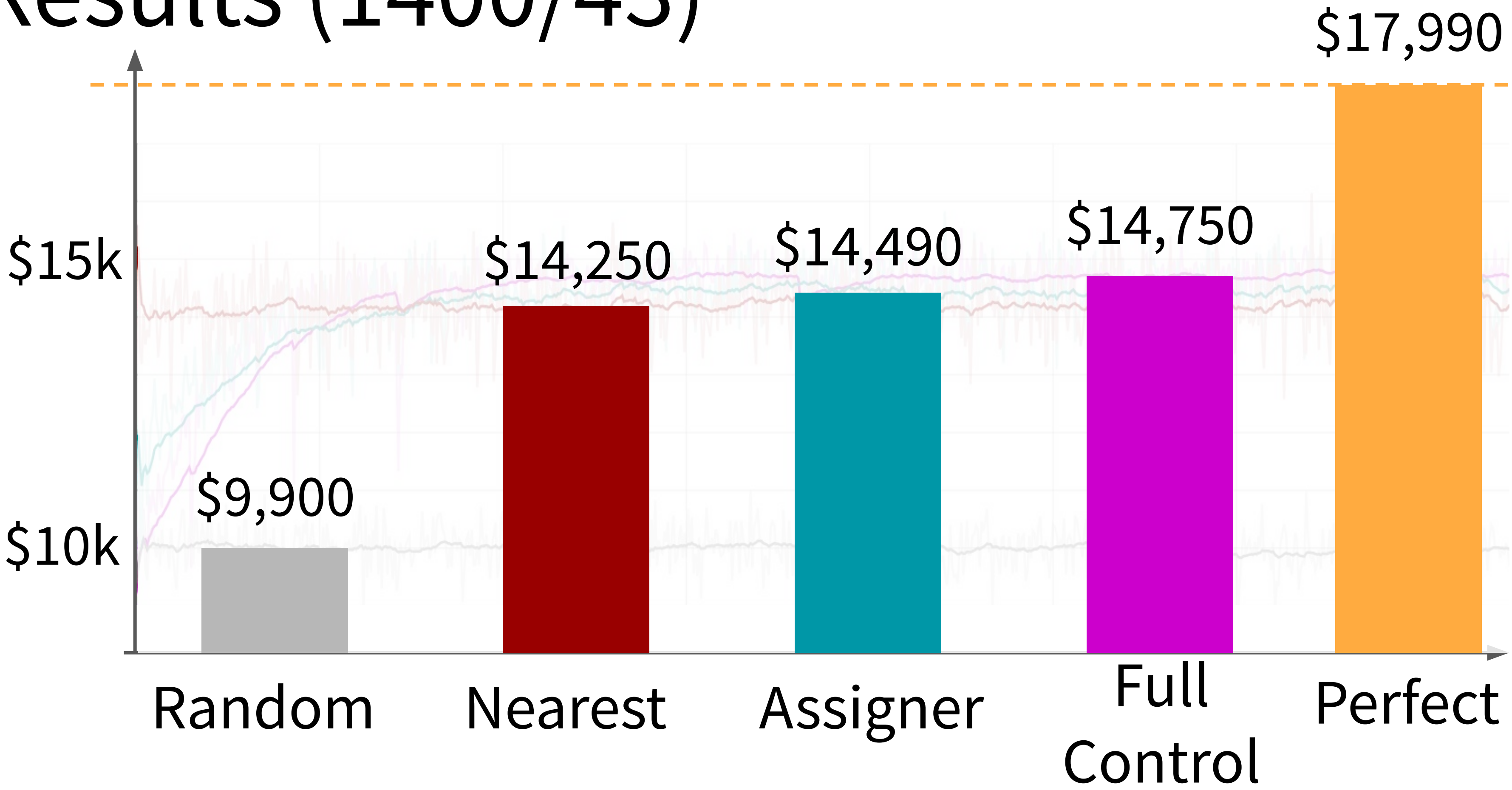
Results (1400/43)



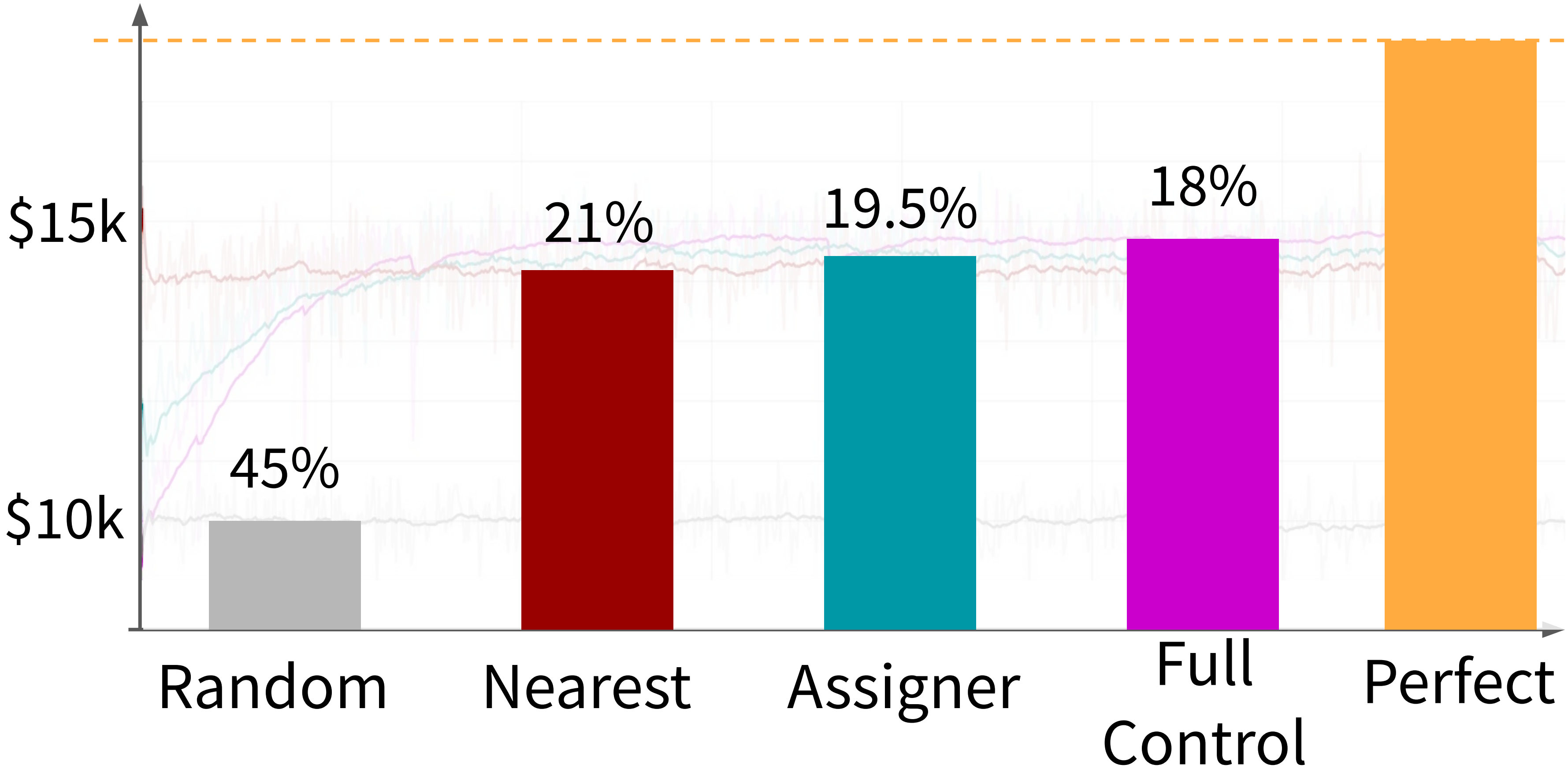
Results (1400/43)



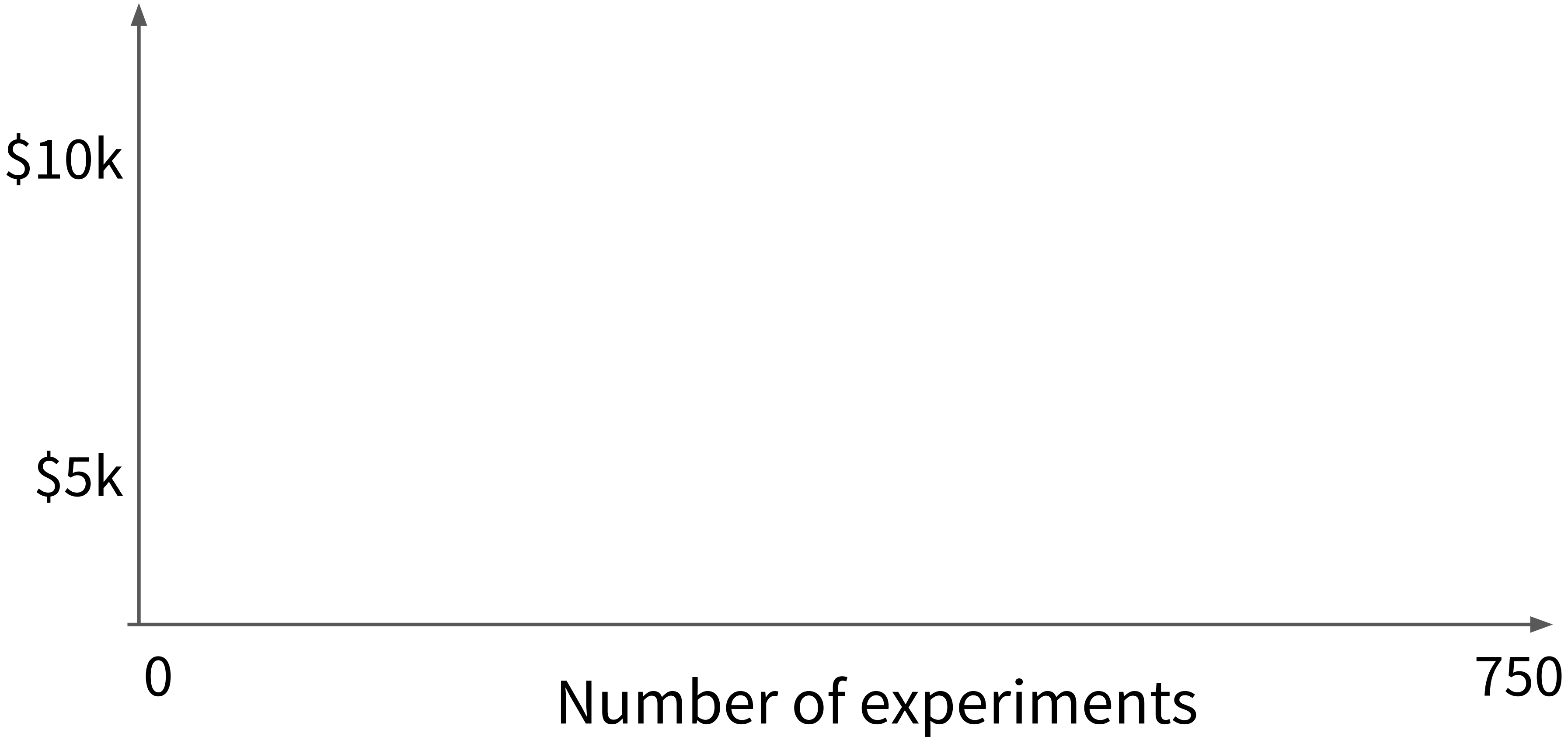
Results (1400/43)



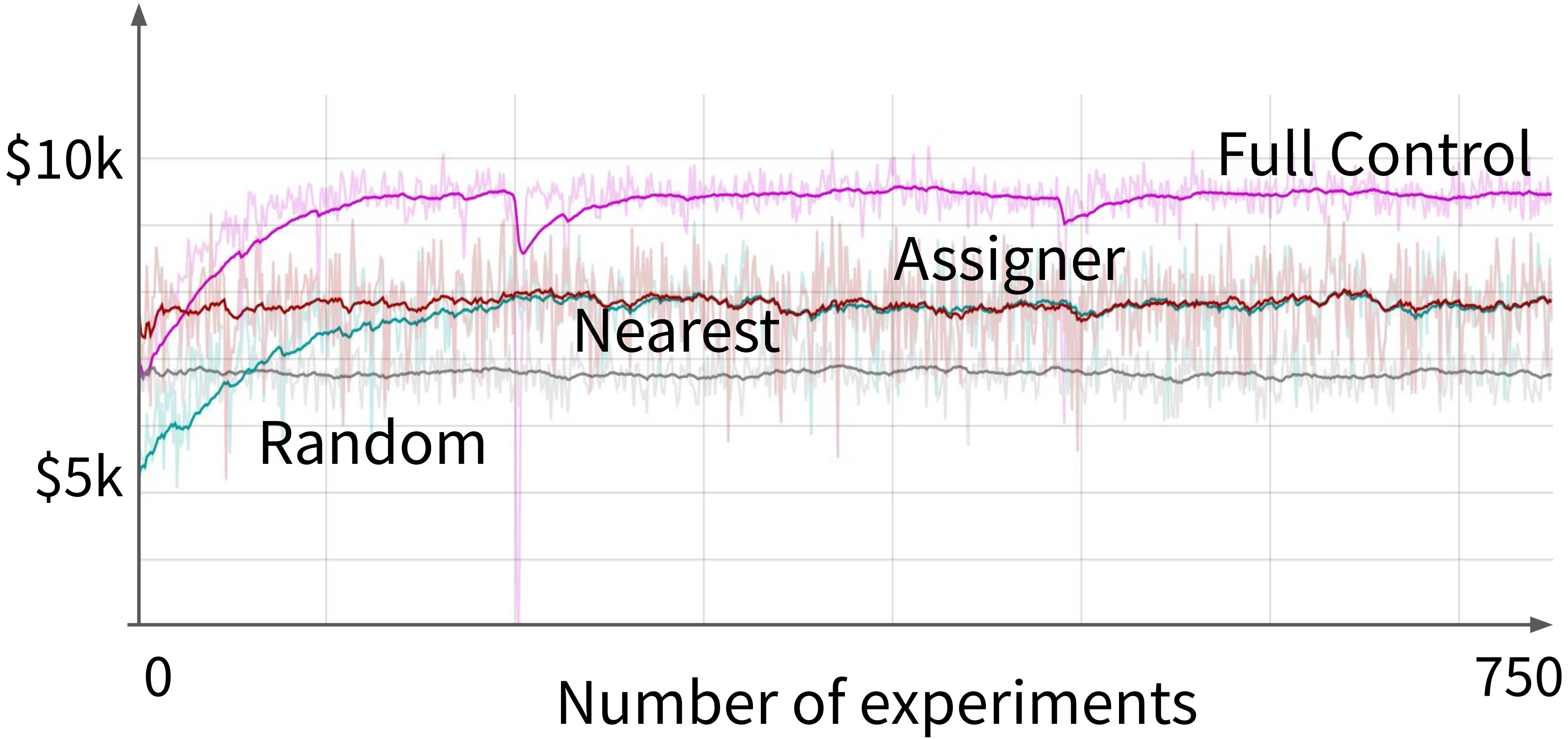
Results (1400/43)



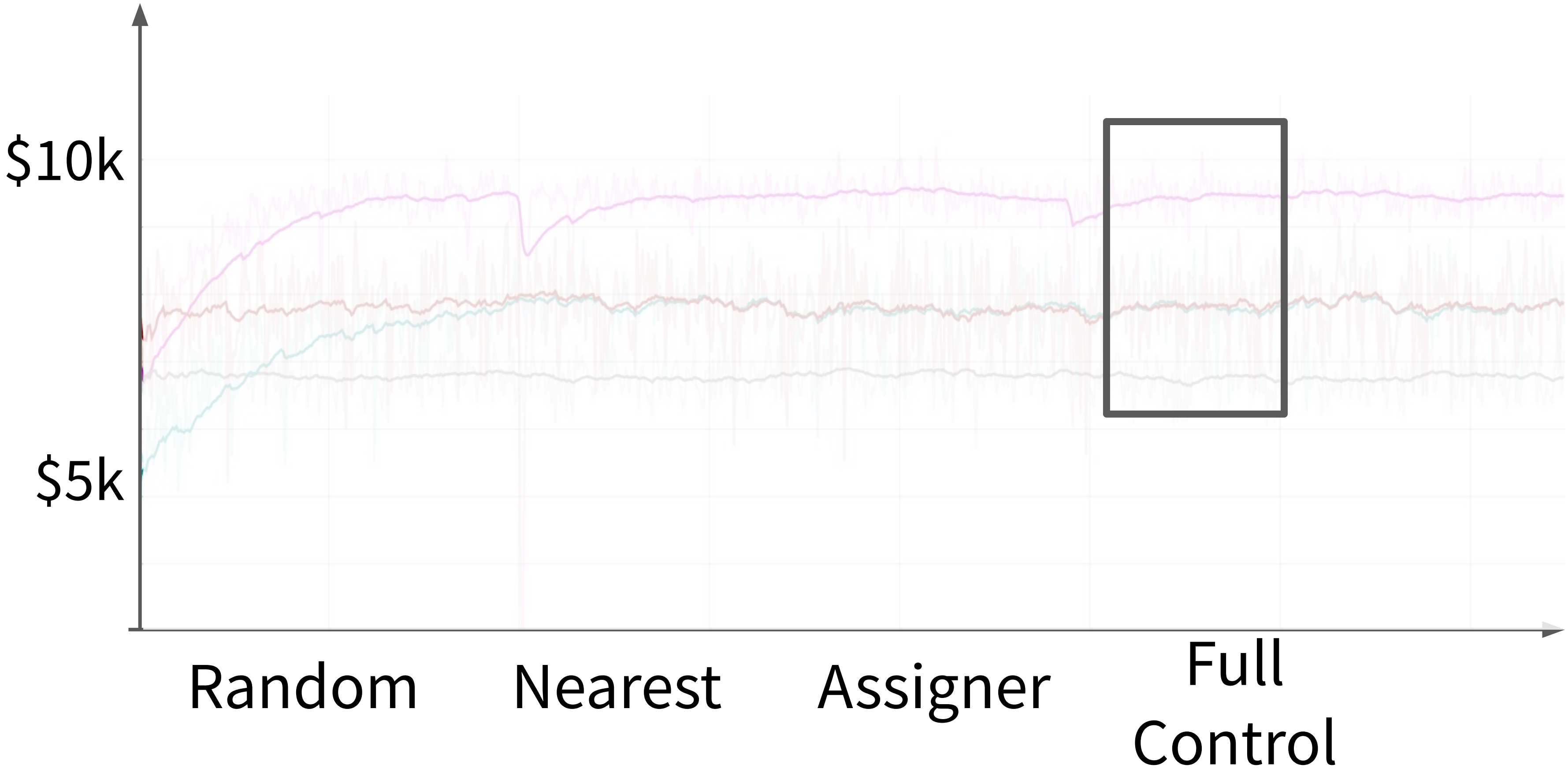
Results (1400/14)



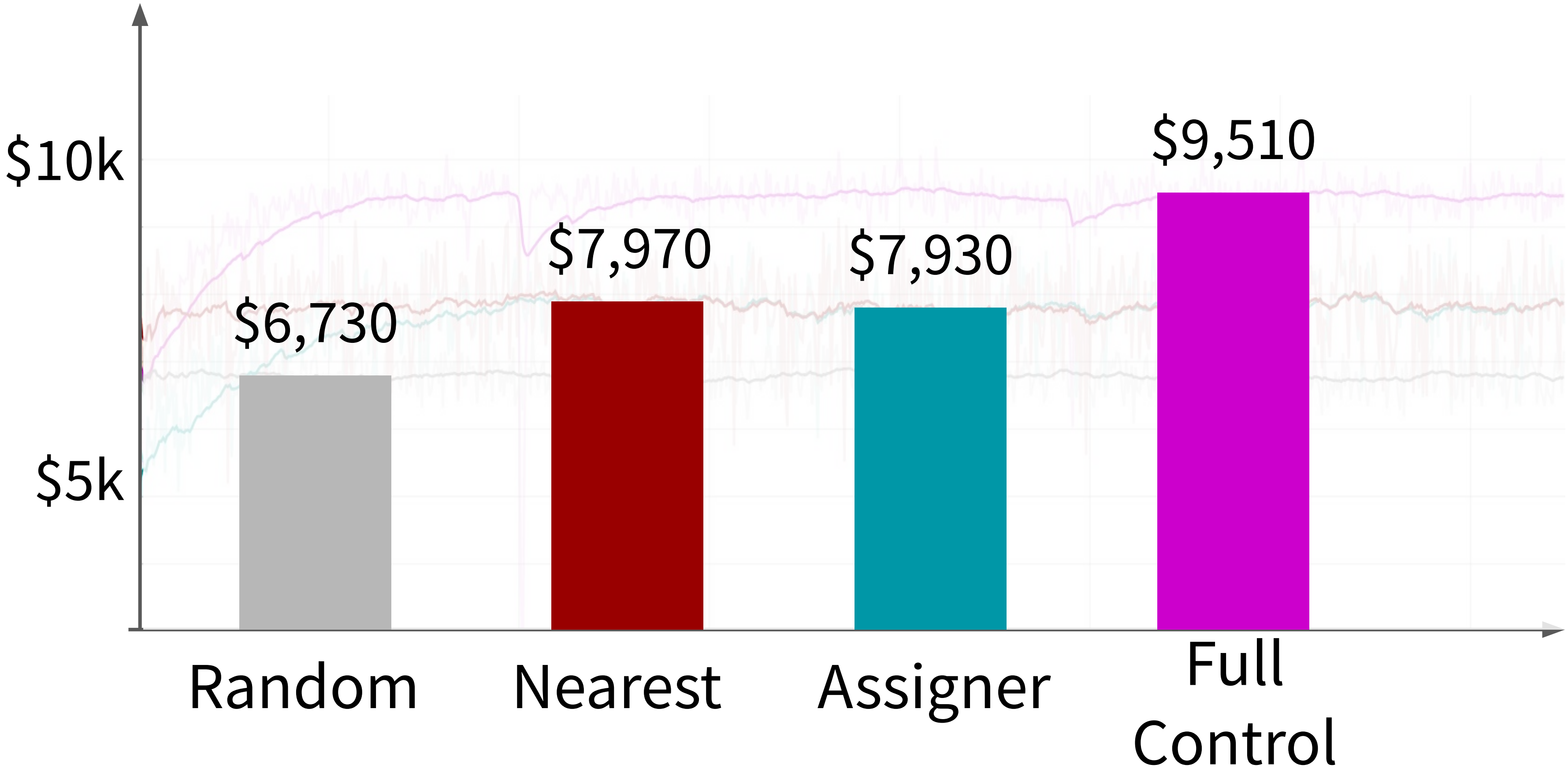
Results (1400/14)



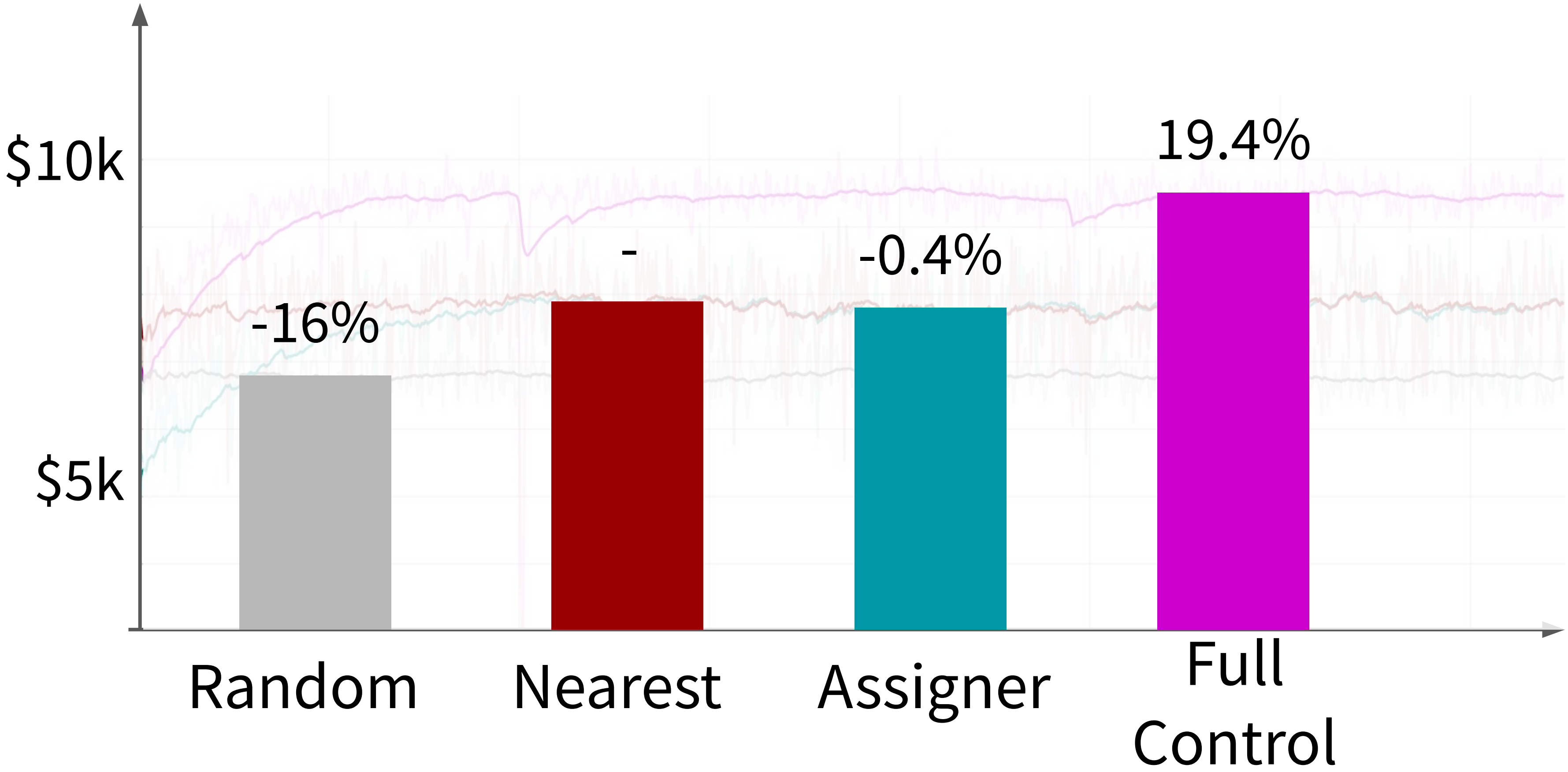
Results (1400/14)



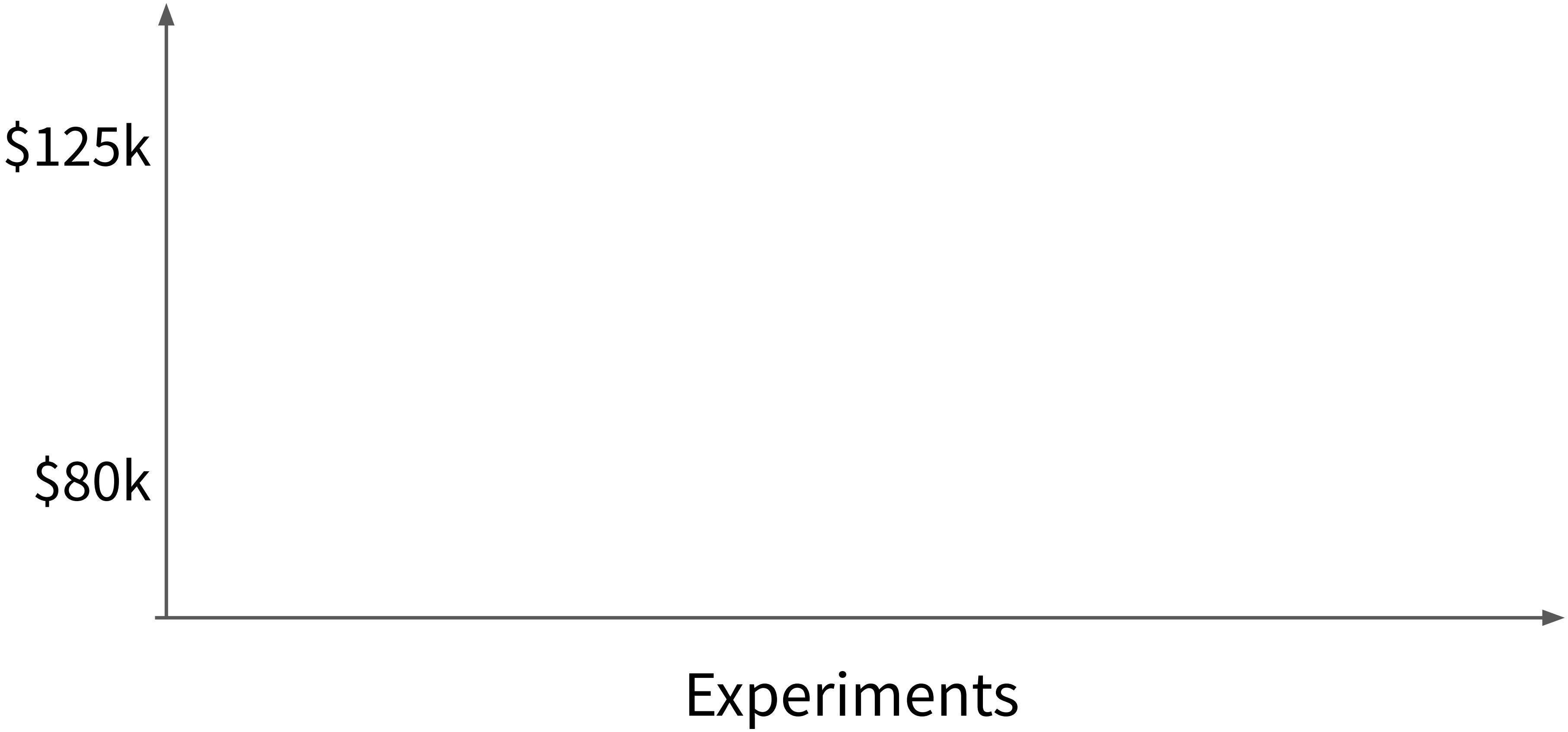
Results (1400/14)



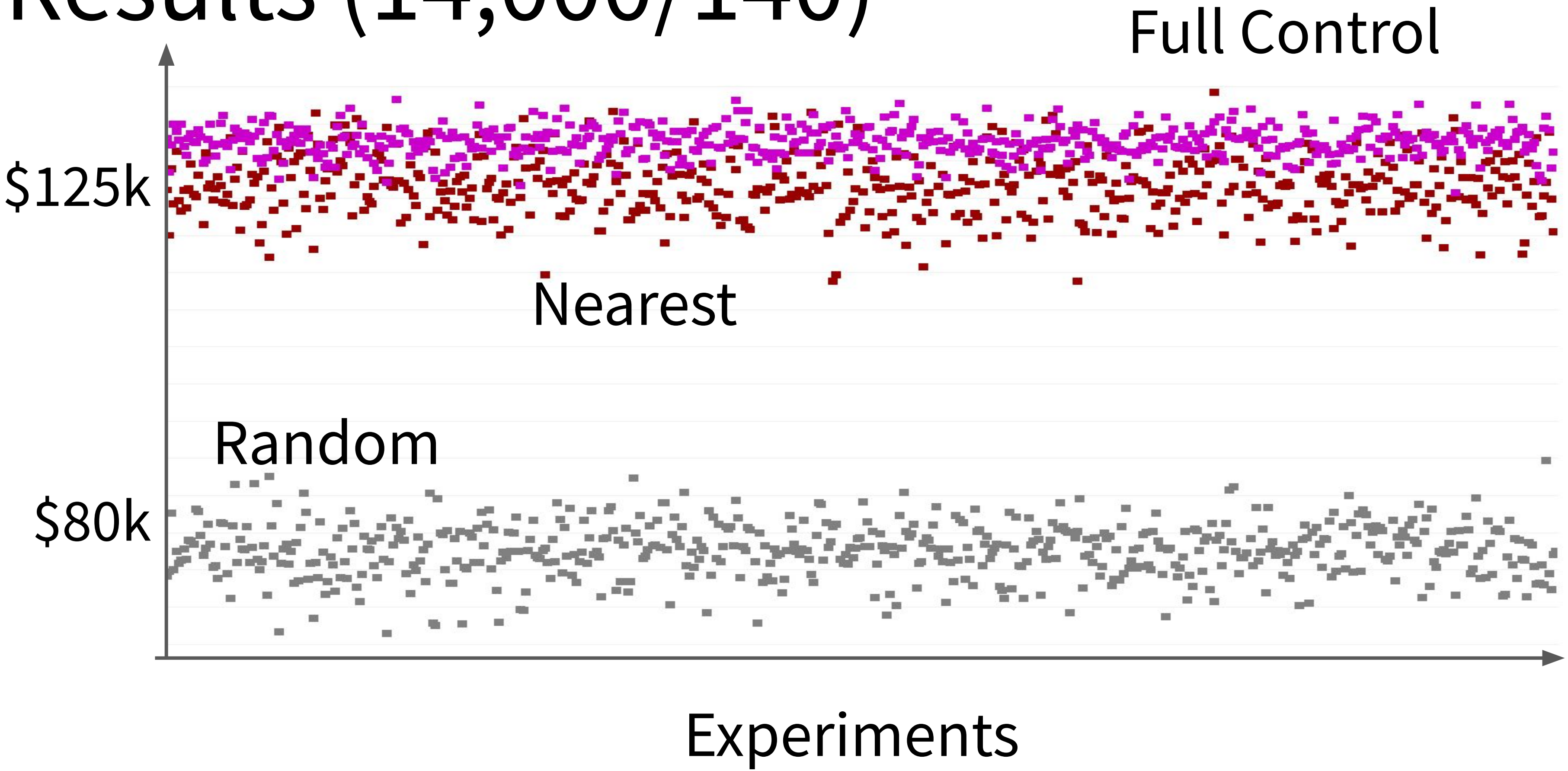
Results (1400/14)



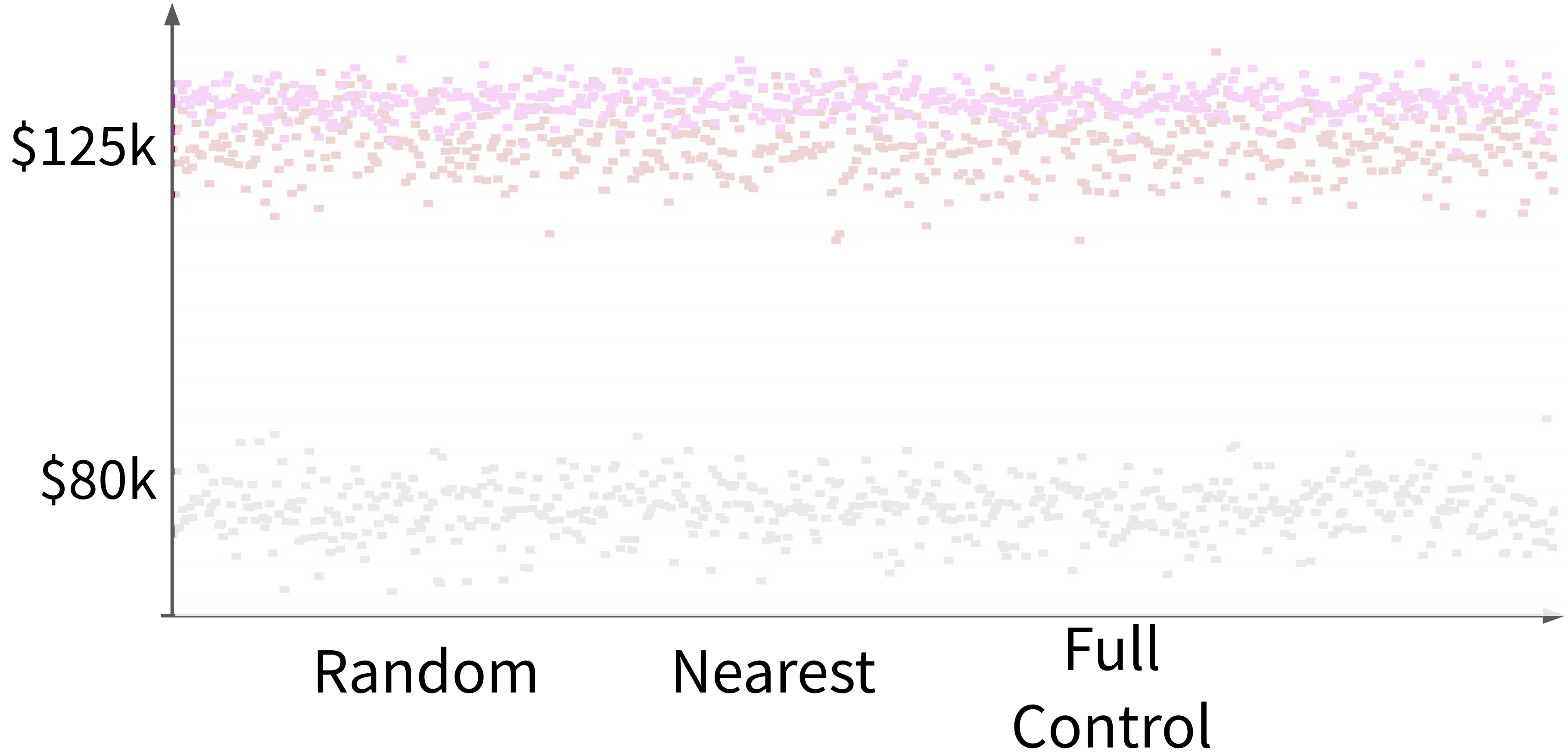
Results (14,000/140)



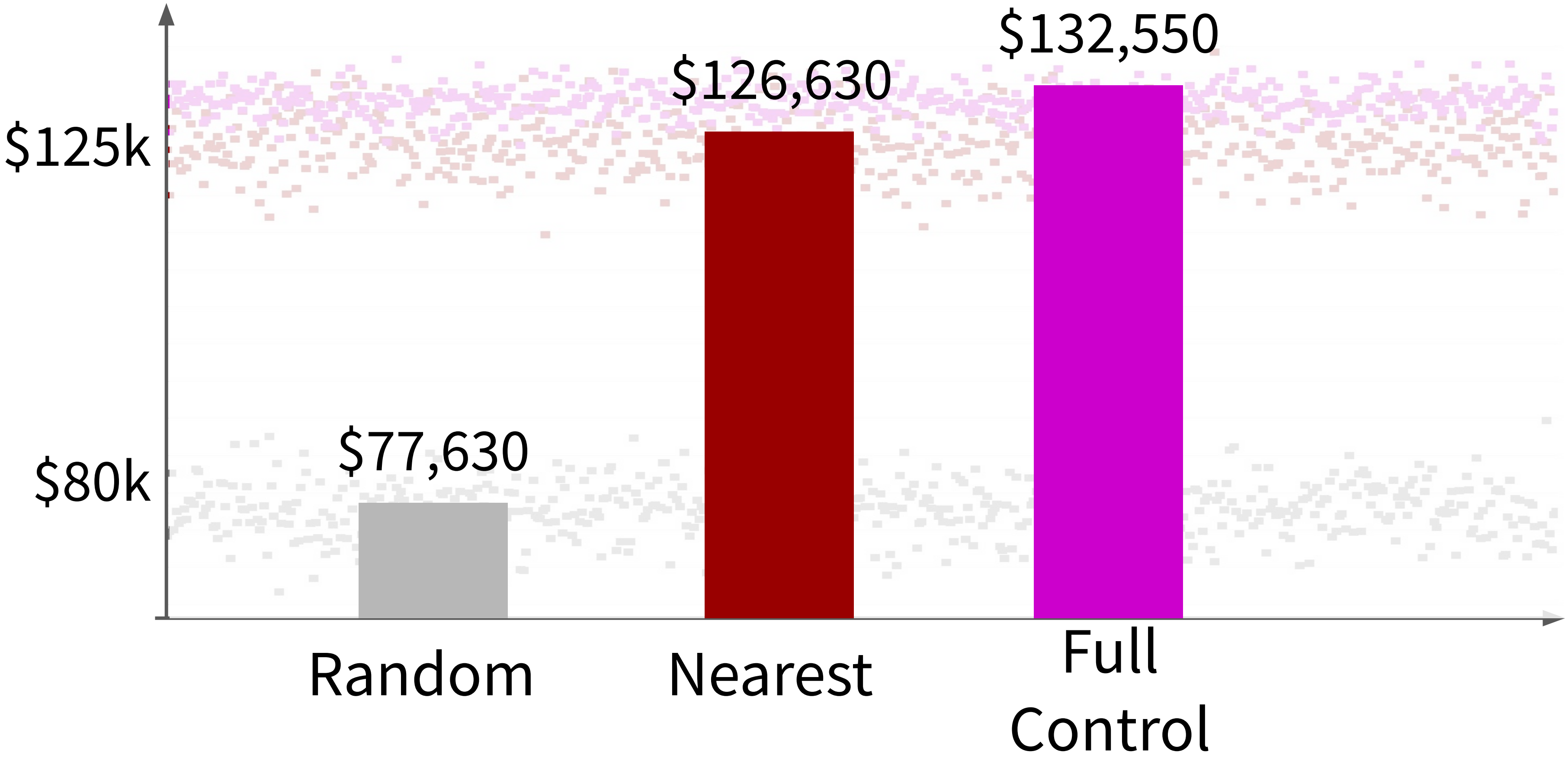
Results (14,000/140)



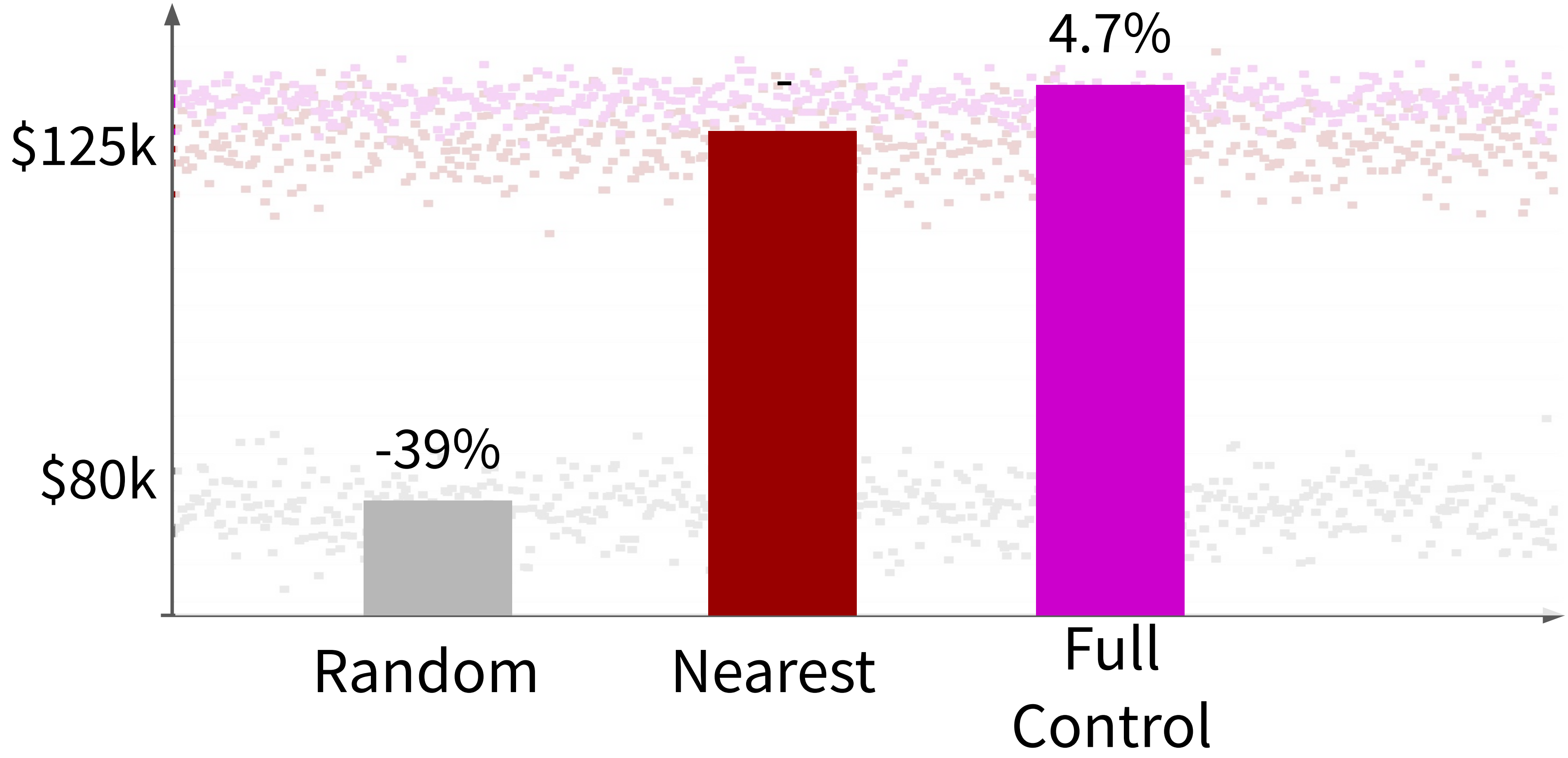
Results (14,000/140)



Results (14,000/140)



Results (14,000/140)



Concluding Remarks

Highlights

Deep RL-based policy the current best

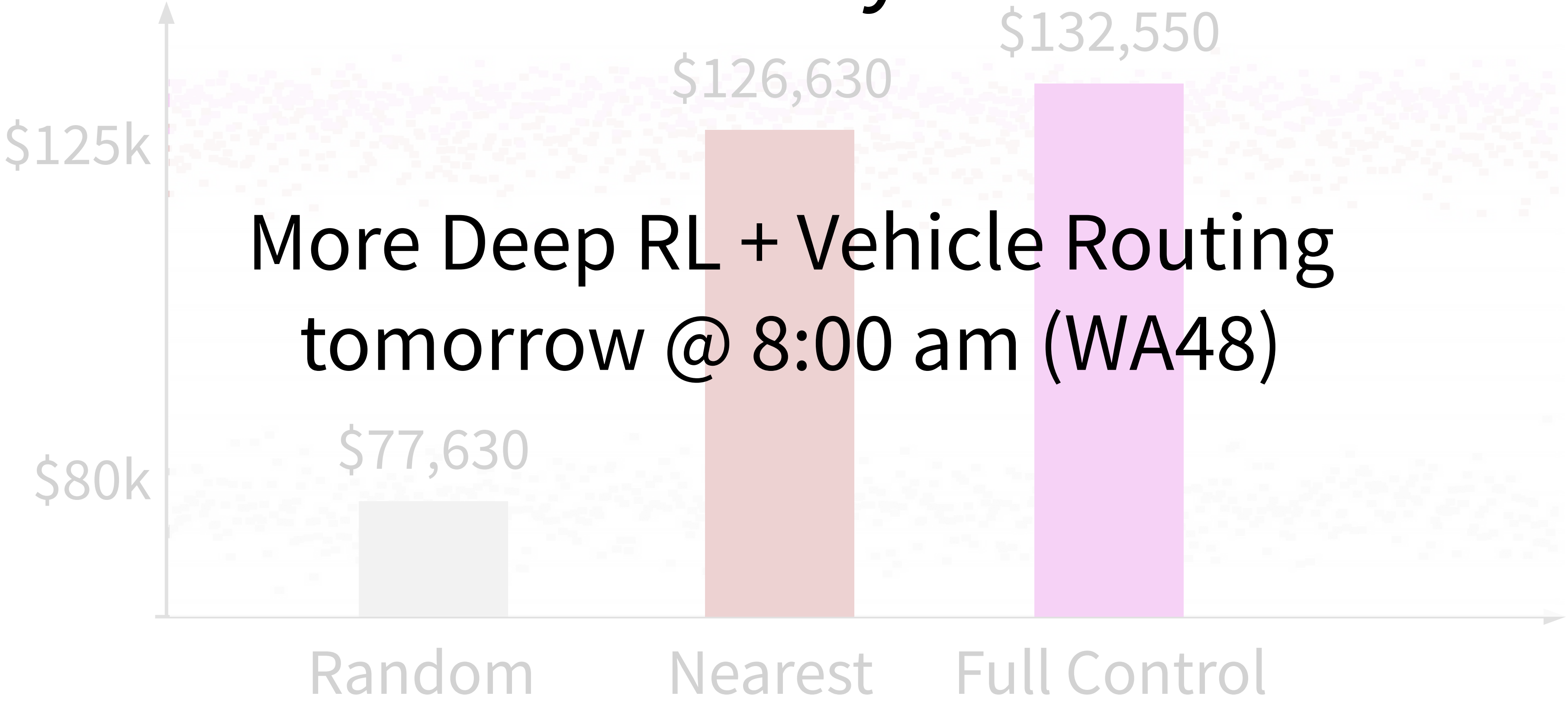
Scalable without retraining

Quick action selection even in large instances (< 0.05 s)

TODOs:

Benchmark improvements, full scale instances

Thank you



More Deep RL + Vehicle Routing
tomorrow @ 8:00 am (WA48)

nicholas.kullman@etu.univ-tours.fr