Safe and Scalable Planning Under Uncertainty for Autonomous Driving

Thesis Defense

Maxime Bouton

January 23rd , 2020

SISL
Stanford Intelligent Systems Laboratory

Motivation

Road fatalities represent about **1.35 millions** of death each year worldwide.

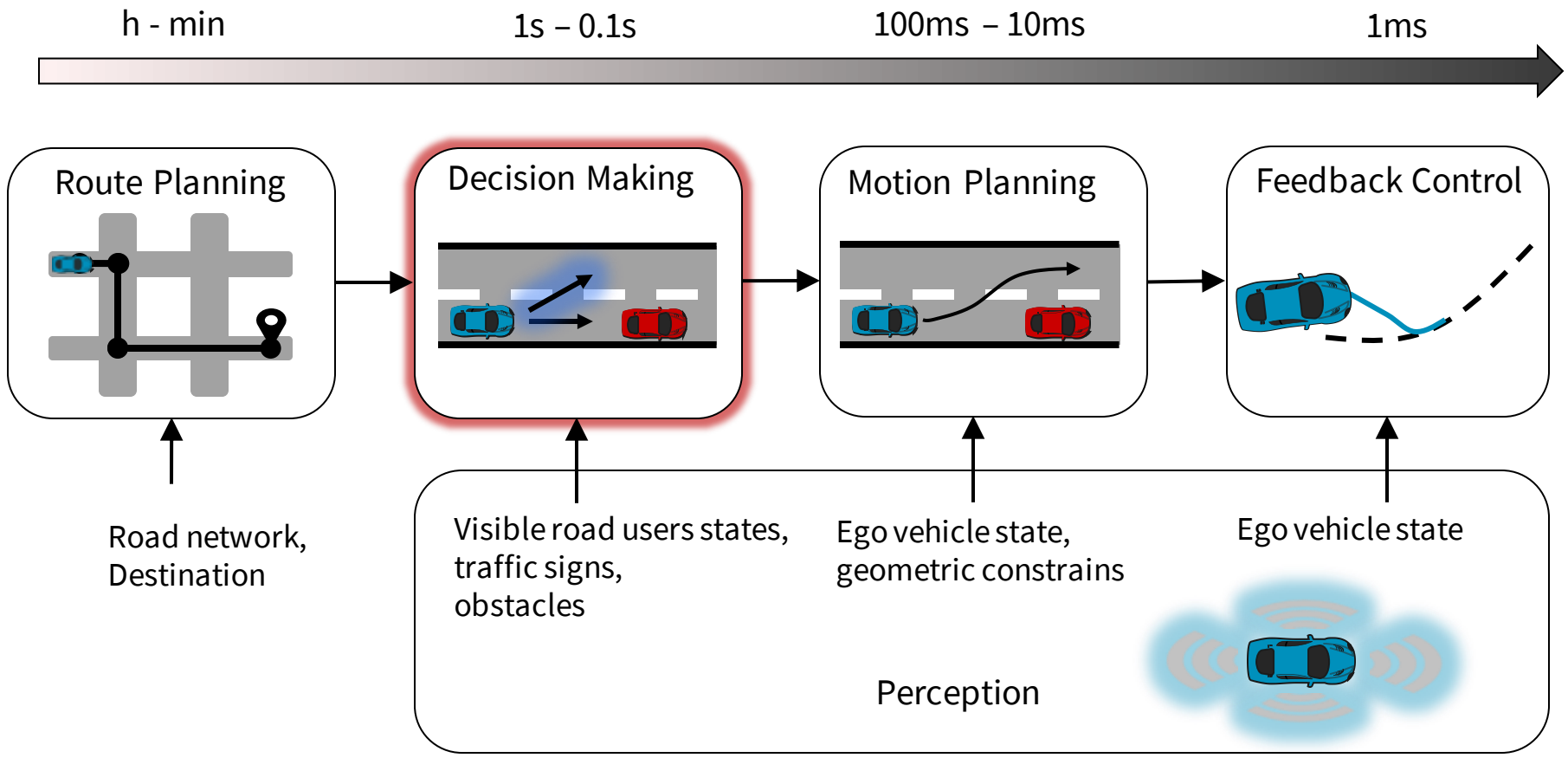
94% of serious crashes are caused by human error.

source: LA Times

Source: IEEE spectrum

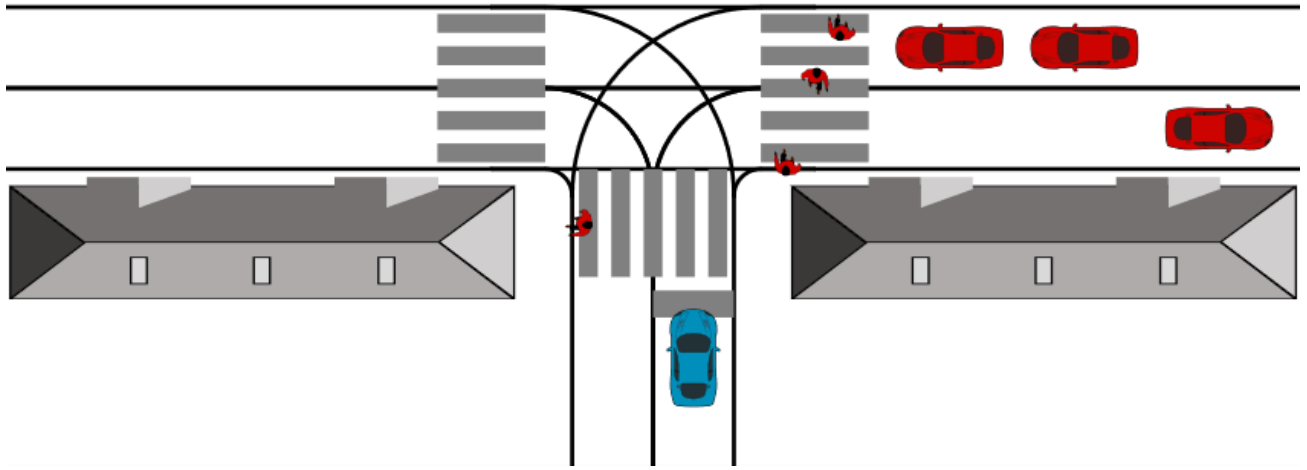
World Health Organization, ‘‘Global status report on road safety,’’ Tech. Rep., 2018.
National Highway Traffic Safety Administration, ‘‘Critical reasons for crashes investigated in the national motor vehicle crash causation survey,’’ Tech. Rep. DOT HS 812 506, 2018.

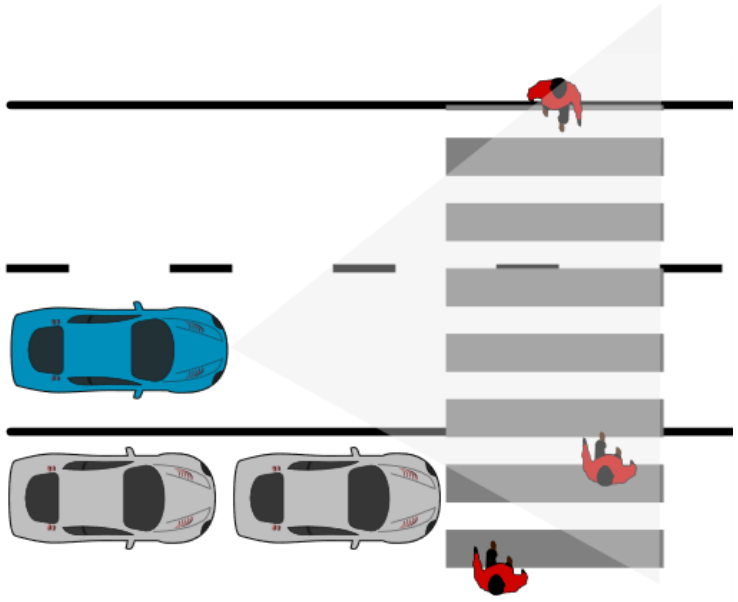
The Autonomy Stack



Decision Making Algorithms

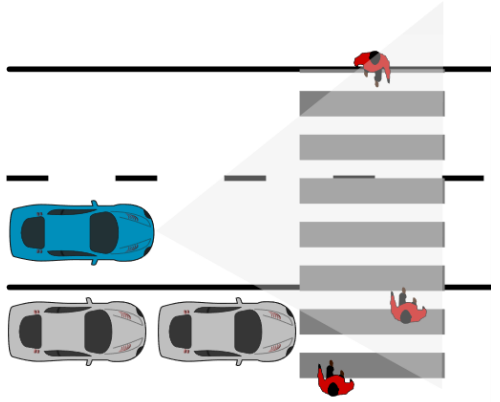
- **Safe**
- **Robust to sensor limitations**
- **Scalable**

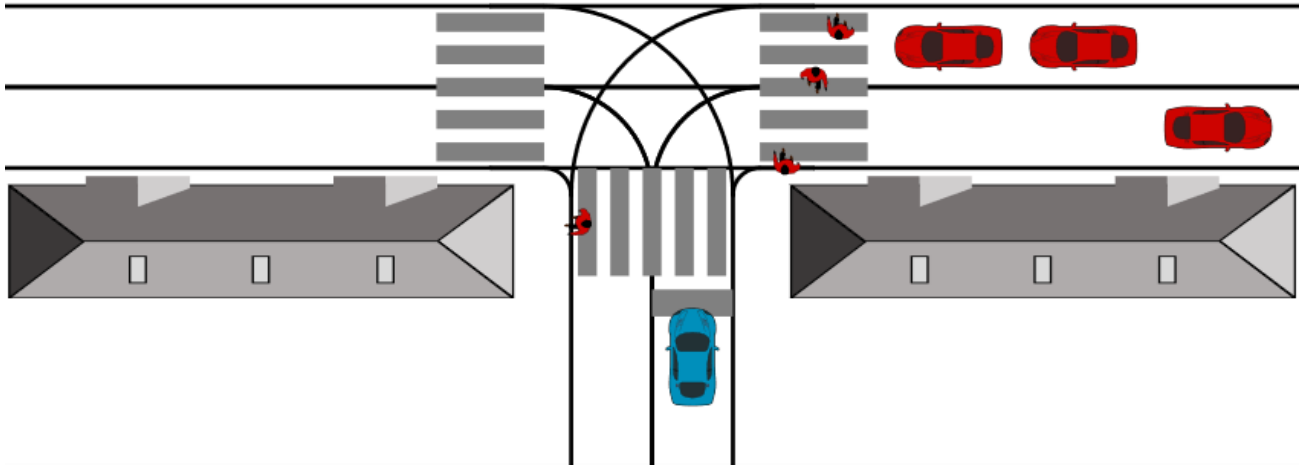




Challenges for Autonomous Vehicles

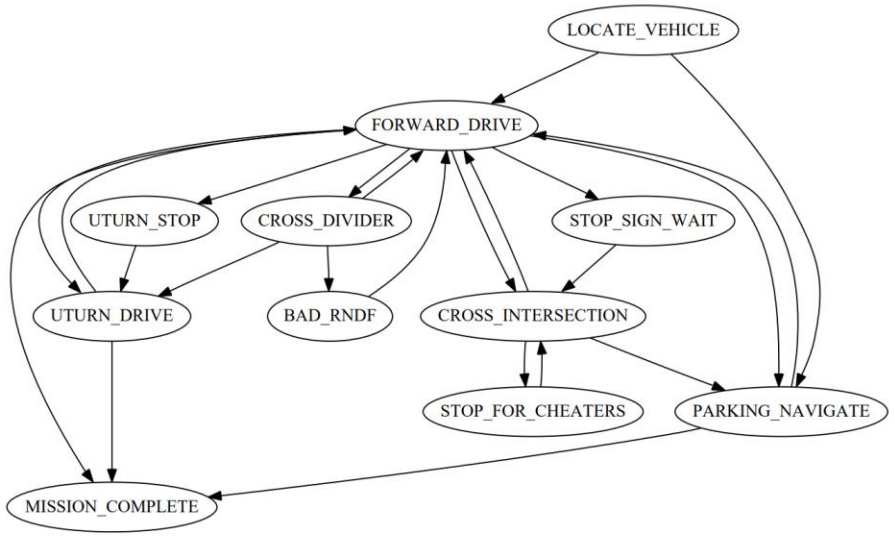
- **Diverse traffic**
- **Crowded**
- **Occluded**
- **Unexpected behaviors**





Source: A. Palffy, J. F. P. Kooij, D. M. Gavrila, "Occlusion aware sensor fusion for early crossing pedestrian detection," in *IEEE Intelligent Vehicles Symposium (IV)*, 2019.

Related Work: Rule-Based / Finite State Machines



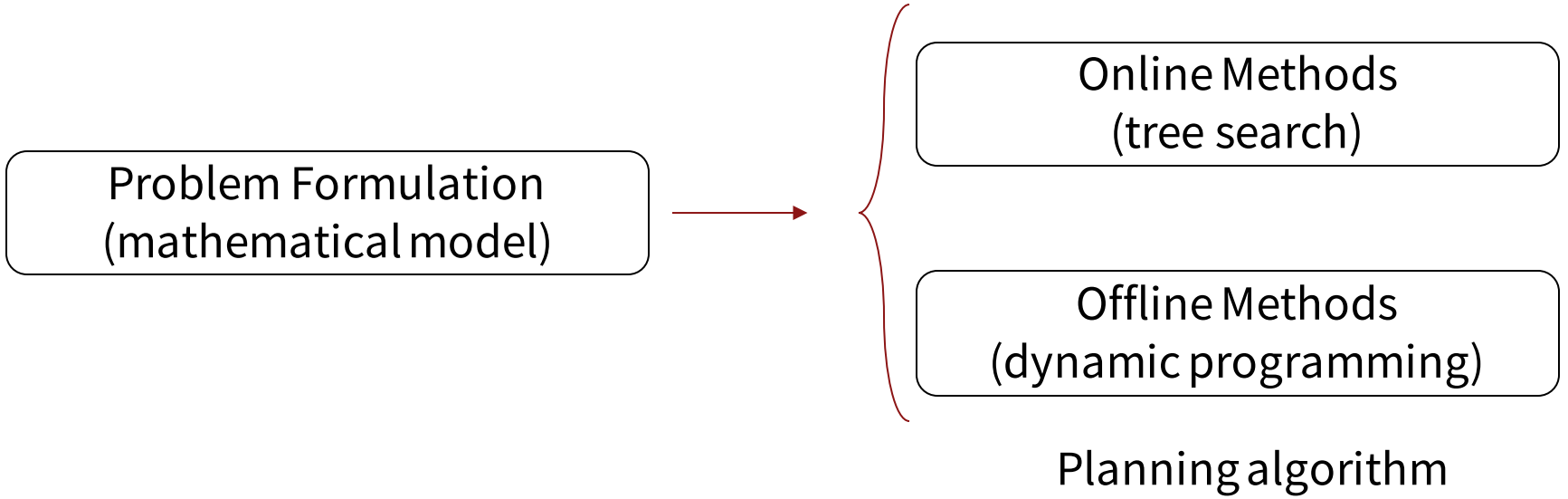
Source: M. Montemerlo et al. "Junior: The stanford entry in the urban challenge." *Journal of field Robotics* 25.9 (2008): 569-597.

Source: L. Fletcher et al. "The MIT–Cornell collision and why it happened." *Journal of Field Robotics* 25.10 (2008): 775-807.

Limitations:

- Requires anticipating every situations
- Difficult to scale to complex scenarios
- Hard to take into account uncertainty (e.g. perception noise)

Related Work: Planning and Optimization

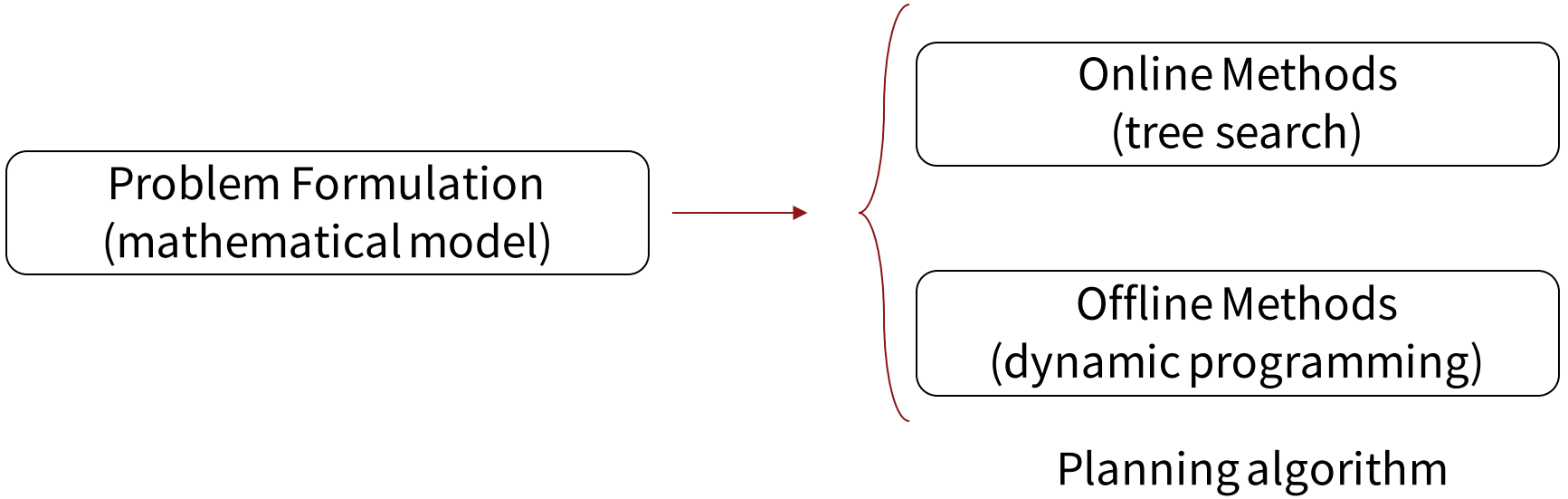


Claim: A model that can capture **sensor uncertainty**, **stochastic behavior**, and **drivers' intention** would lead to good decision strategies.

Limitations:

- Requires a model
- Computationally expensive

Related Work: Planning and Optimization



A model that can capture **sensor uncertainty**, **stochastic behavior**, and **drivers' intention** would lead to good decision strategies.

Candidate: partially observable Markov decision process (POMDP)

Outline

Introduction

Decision Making for Autonomous Driving

1. Mathematical formulation

Scalability

2. Utility Decomposition

3. Deep Corrections

Safety

4. Safe Reinforcement Learning

5. Model Checking in POMDPs

6. Conclusions

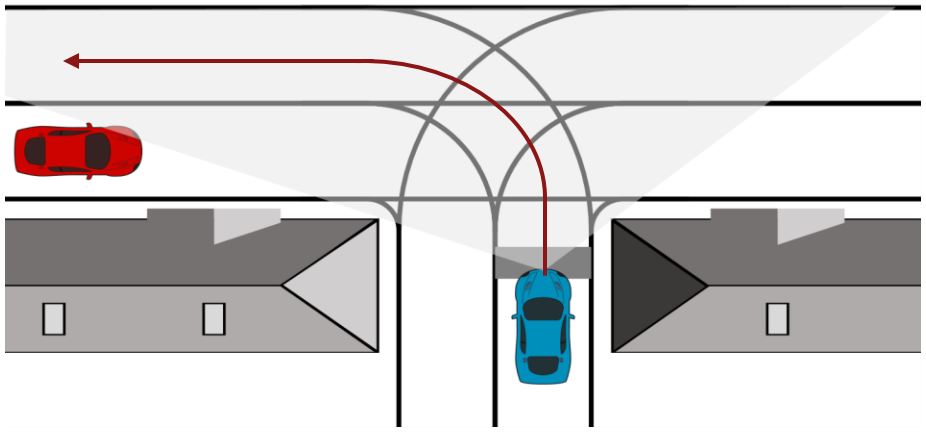
Partially Observable Markov Decision Processes

Mathematical framework for modeling processes with:

- Stochastic evolution
- State uncertainty (sensor noise, intentions)

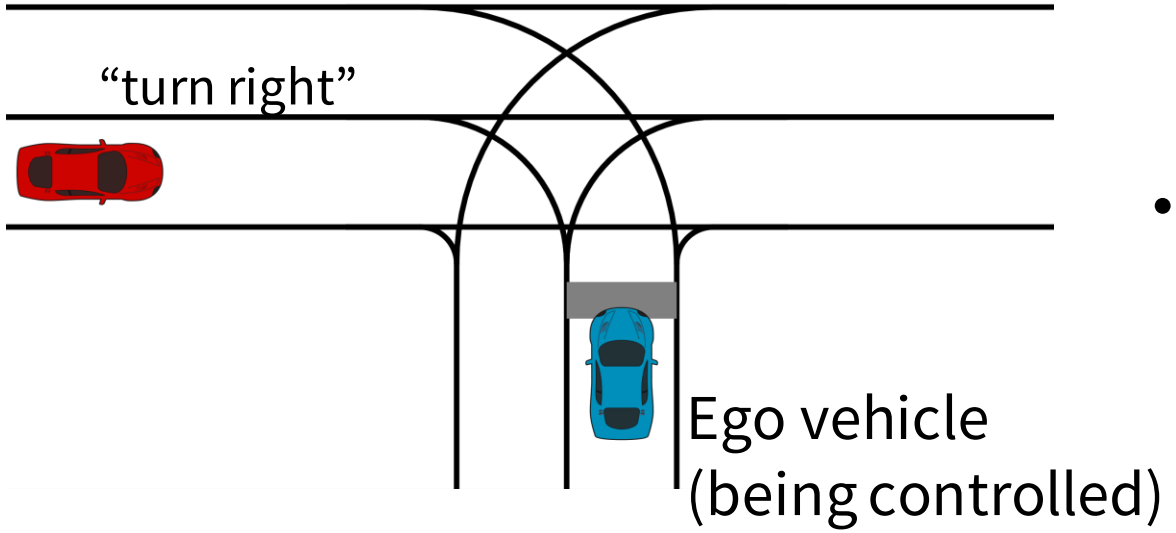
$$(\mathcal{S}, \mathcal{A}, \mathcal{O}, T, O, R, \gamma)$$

Example: left-turn at an occluded intersection:



State

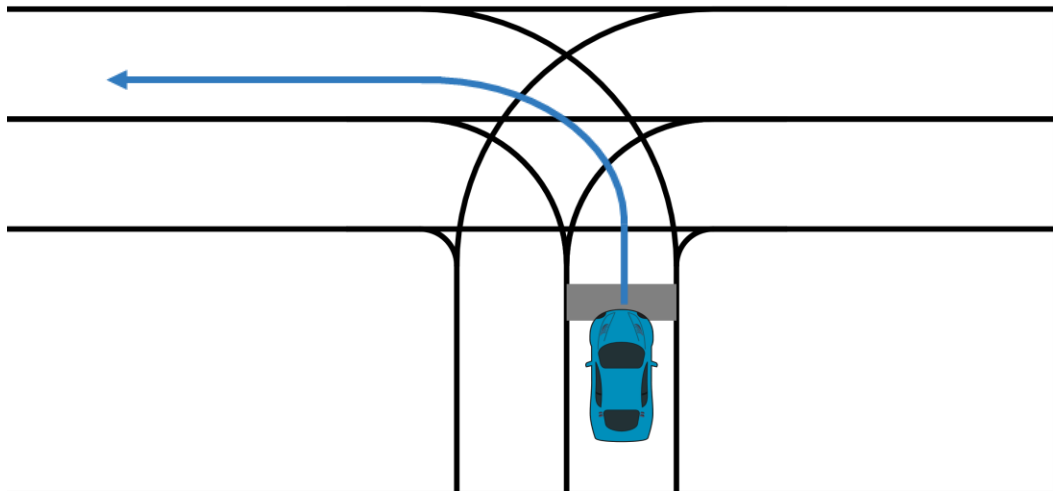
$(S, \mathcal{A}, O, T, O, R, \gamma)$



- Every possible positions and velocities of the ego vehicle and the other car
  $(s_{ego}, v_{ego}, s_{other_1}, v_{other_1})$
- Intention of the other driver: turn left, turn right

Actions

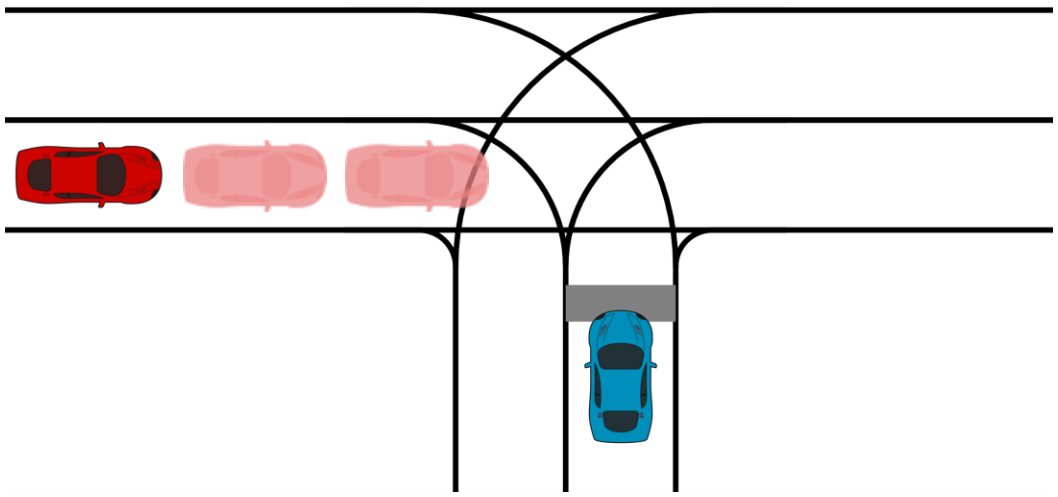
$(S, \mathcal{A}, O, T, O, R, \gamma)$



- $\{-4\ ms^{-2}; -2ms^{-2}; 0ms^{-2}; 2ms^{-2}\}$
- “bead on a wire” , point-mass dynamics

Transition Model

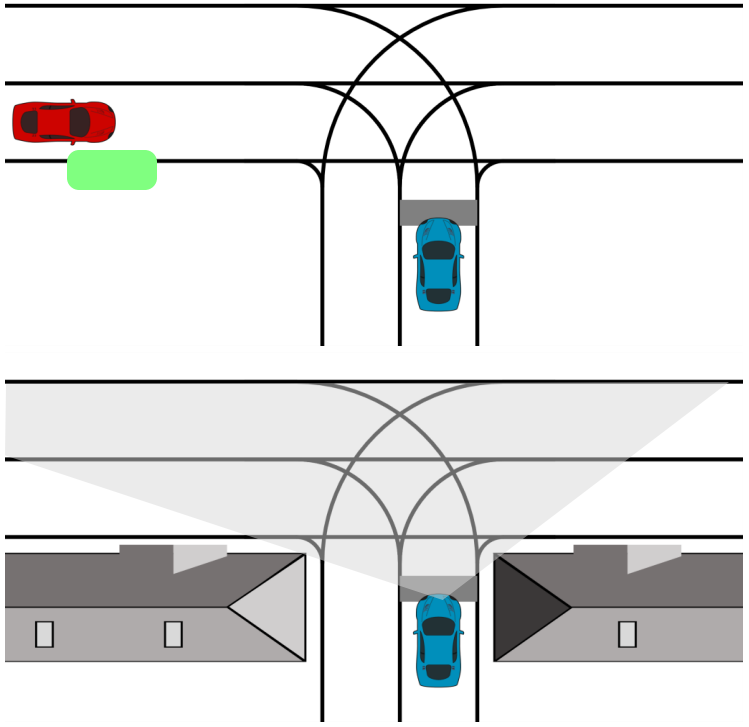
$(S, \mathcal{A}, \mathcal{O}, \mathbf{T}, O, R, \gamma)$



- Probability of transitioning to a next state given the previous state and the action taken.
- Constant velocity with noise
- Probability of appearance

Observations

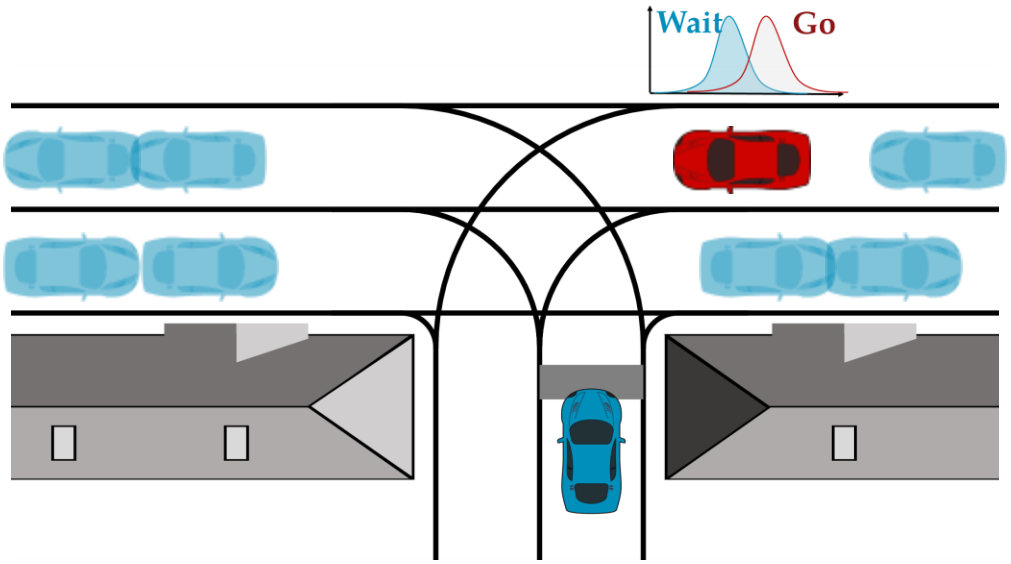
$(S, \mathcal{A}, O, T, o, R, \gamma)$



- Receive noisy measurement of other vehicles position and velocity
- Do not receive observation if there are occlusions

Belief

Since the state is not observable, the agent maintains a **belief** reflecting its internal knowledge of the environment.



A belief is a **distribution** over states

- internal states (intentions)
- partially observed physical states

Policy

Policy: mapping from **beliefs** to **actions**

Find a policy

$$\pi : \mathrm{B} \rightarrow \mathcal{A}$$

that maximizes

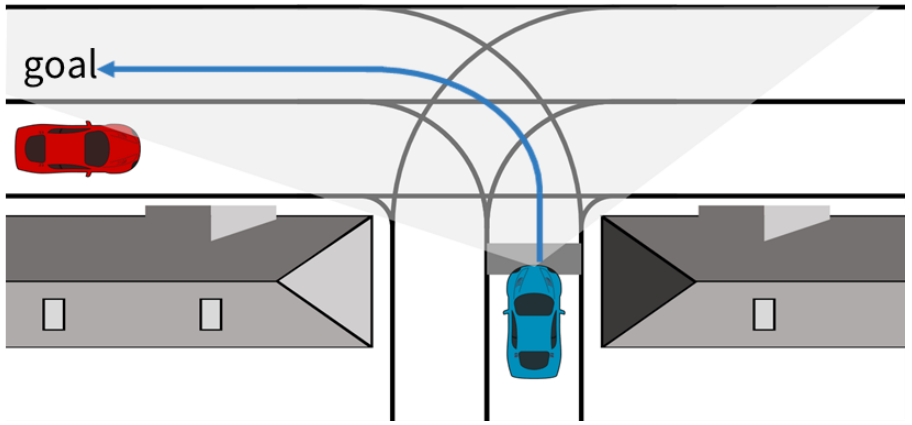
$$E[\sum_{t=0}^{\infty} \gamma^t \, R(s_t, a_t) \mid s_0 \sim b_0]$$

Partially Observable Markov Decision Processes

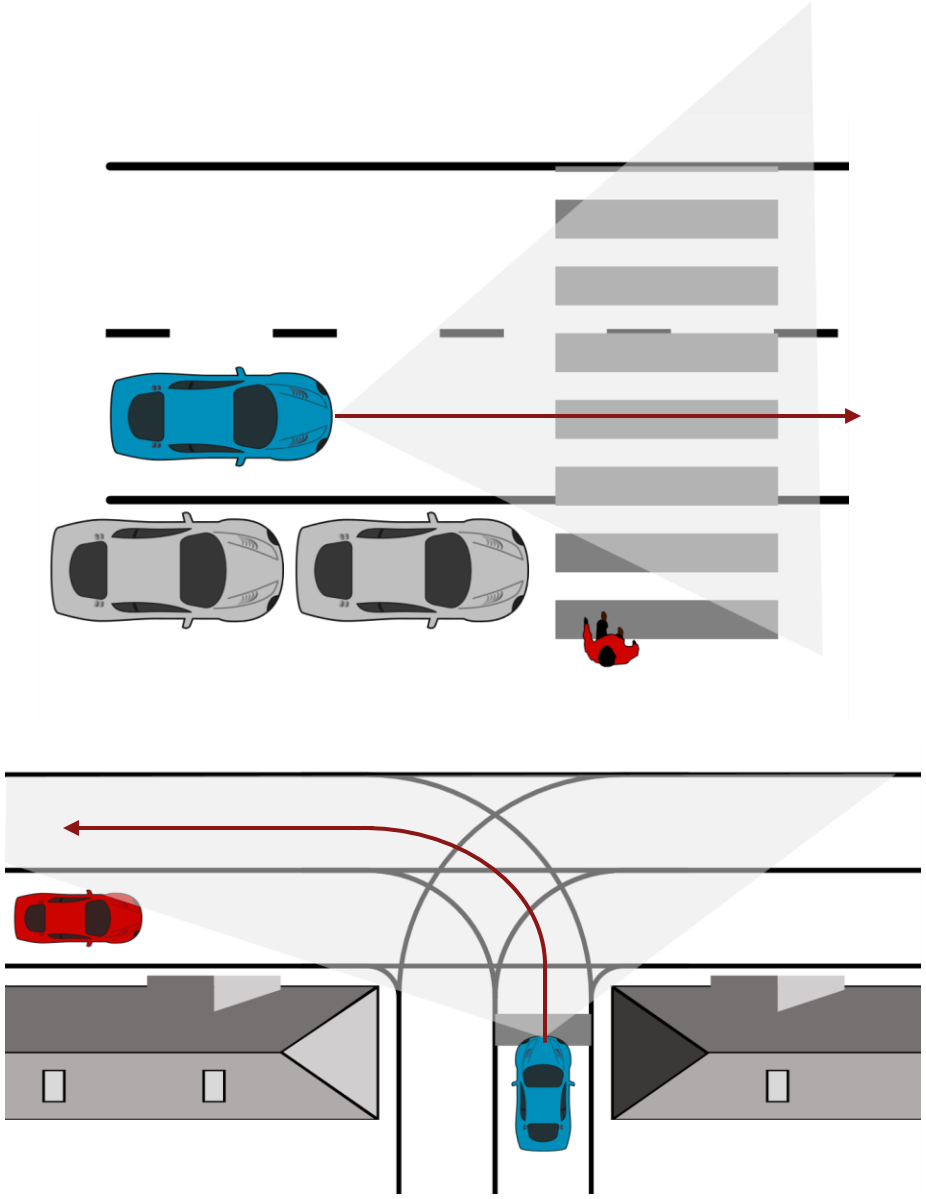
$(\mathcal{S}, \mathcal{A}, \mathcal{O}, T, O, \boldsymbol{R}, \gamma)$

$$R(s,a) = \lambda_{\text{safety}} R_{\text{safety}}(s,a) + \lambda_{\text{efficiency}} R_{\text{efficiency}}(s,a) + \lambda_{\text{comfort}} R_{\text{comfort}}(s,a)$$

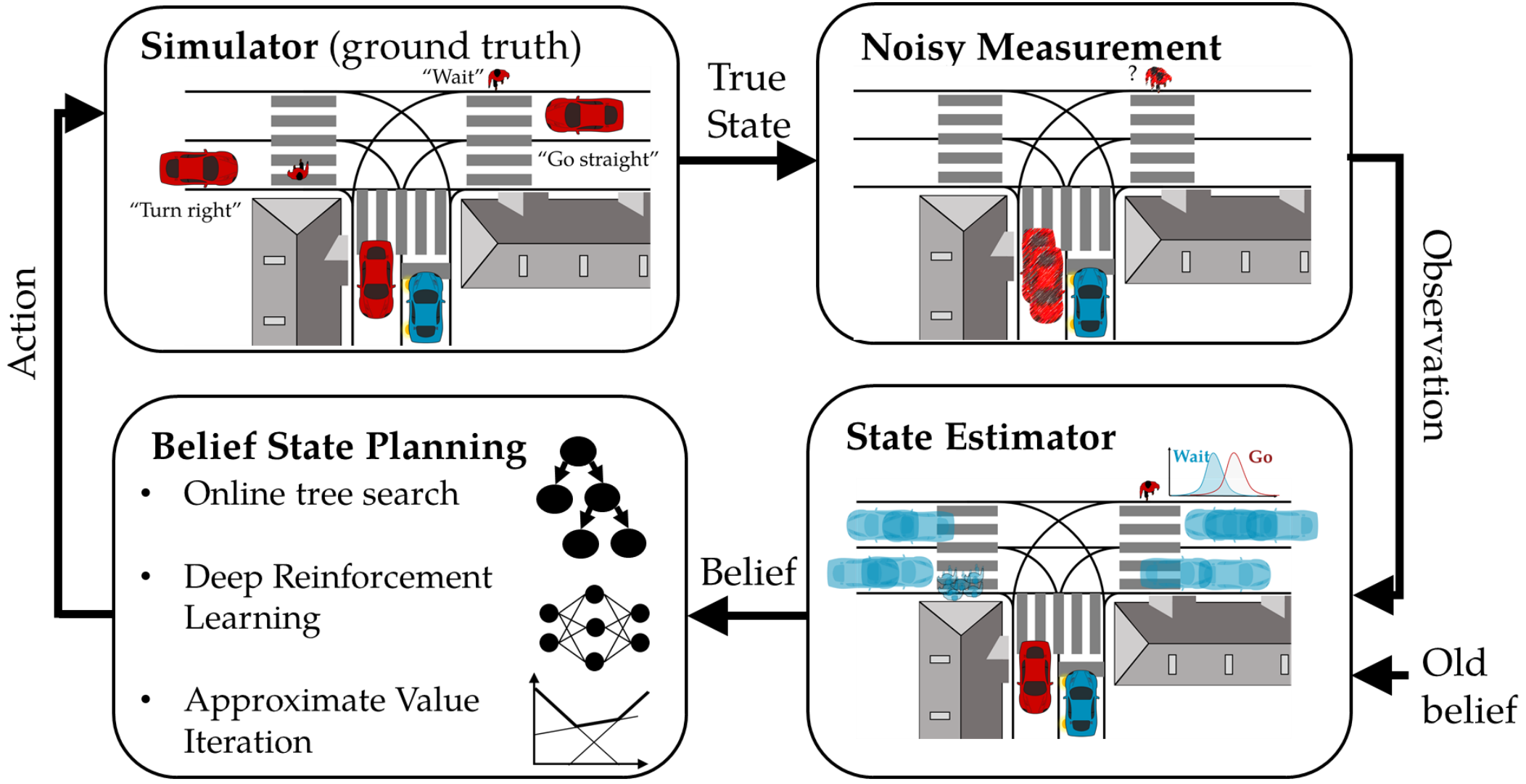
In this work: $R(s,a) = \mathbb{1}_{\text{goal}} - \lambda \mathbb{1}_{\text{collision}}$



Applying the model to different scenarios



Approach Summary



Value Function

$Q^*(b,a)$: average accumulated discounted reward if the agent starts in belief b, takes action a and follows the optimal policy for the remaining steps.

Decision rule: $\pi^*(b) = \arg\max_a Q^*(b,a)$

Bellman equation for POMDPs:

$$Q^{(n+1)}(b,a) = R(b,a) + \gamma \sum_o \Pr(o \mid a,b) \max_{\mathrm{a}} Q^{(n)}(b',a)$$

Immediate reward

Expected future reward

Algorithms for Solving POMDPs

QMDP:

- Up to millions of states
- No information gathering
- Optimistic solution

SARSOP:

- Up to tens of thousands of states
- Information gathering

Tractable

Intractable!

Two users problem:

- 6D Grid
  $(s_{ego}, v_{ego}, s_{other_1}, v_{other_1}, s_{other_2}, v_{other_2})$
- Total number of states:
  200k for the crosswalk
  7M for the intersection

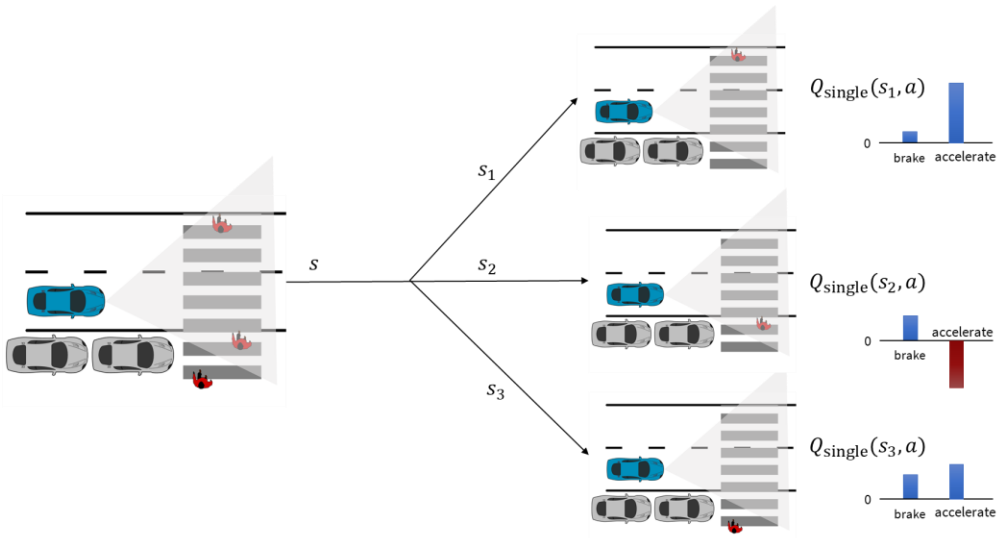
Note: Both require discrete state spaces

1.Introduction

2.Utility Decomposition

3.Deep Corrections
4.Safe Reinforcement Learning
5.Model Checking in POMDPs
6.Conclusion

- Scale decision making algorithms to scenario with multiple traffic participants



Utility Fusion

Approximate the solution to the large problem as a combination of the subproblems:

$$Q^*(b,a) \approx f\left(Q_1^*(b_1,a), \ldots, Q_n^*(b_n,a)\right)$$

Examples:

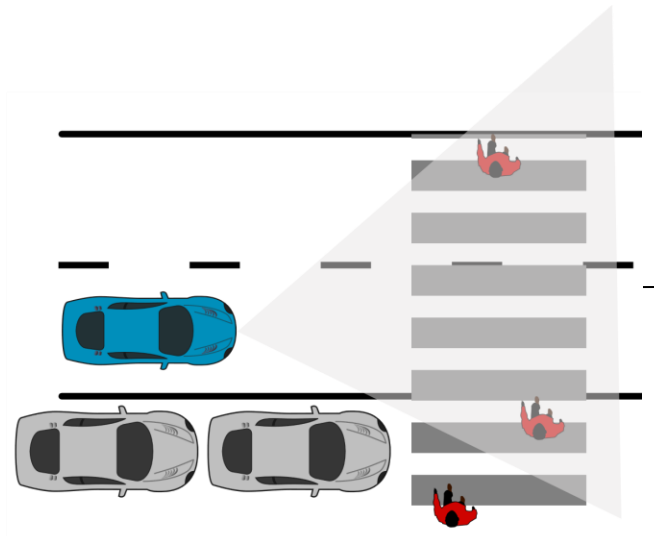
$$Q^*(b,a) \approx \sum_i Q_i^*(b_i,a)$$

or

$$Q^*(b,a) \approx \min_i Q_i^*(b_i,a)$$

Entity Based Decomposition

Evaluating Q_{single} is computationally cheap

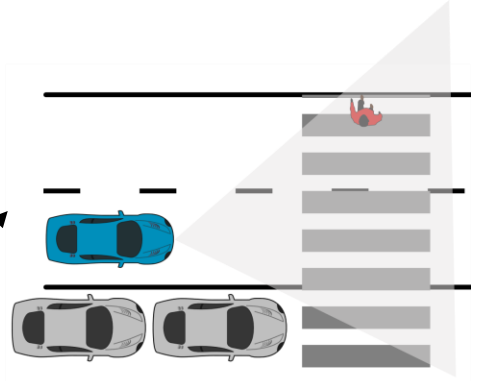


b

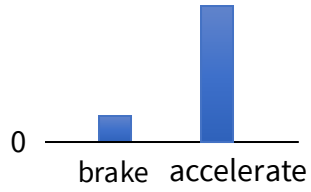
b_1

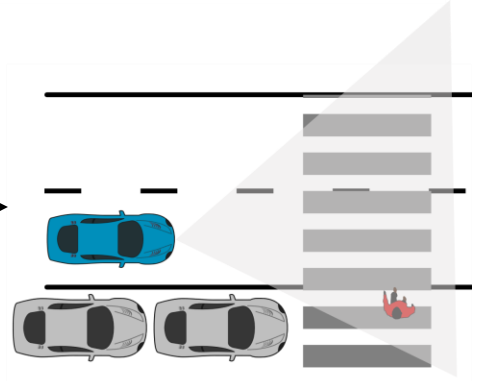
b_2

b_3

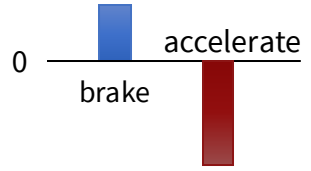


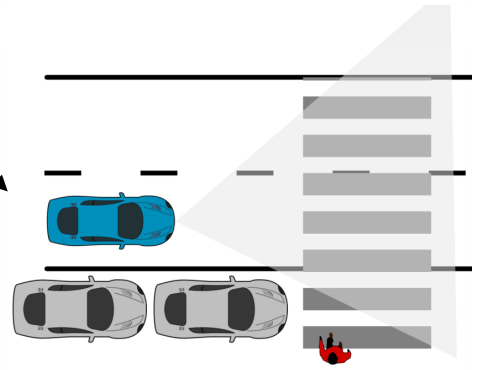
$Q_{\text{single}}(b_1, a)$



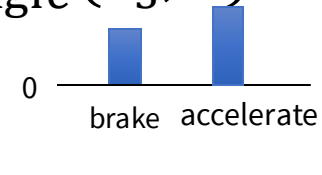


$Q_{\text{single}}(b_2, a)$

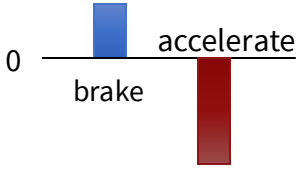




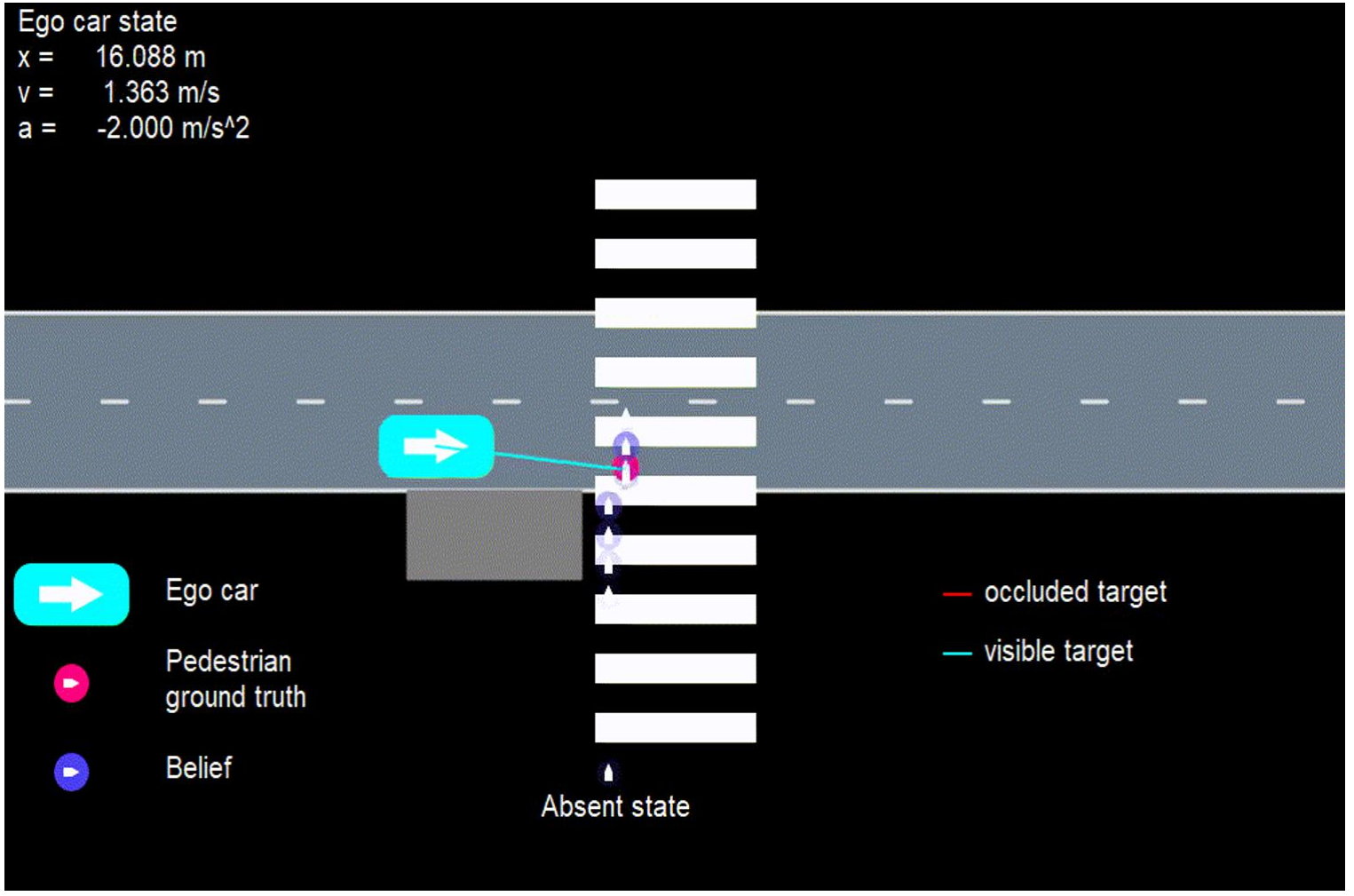
$Q_{\text{single}}(b_3, a)$



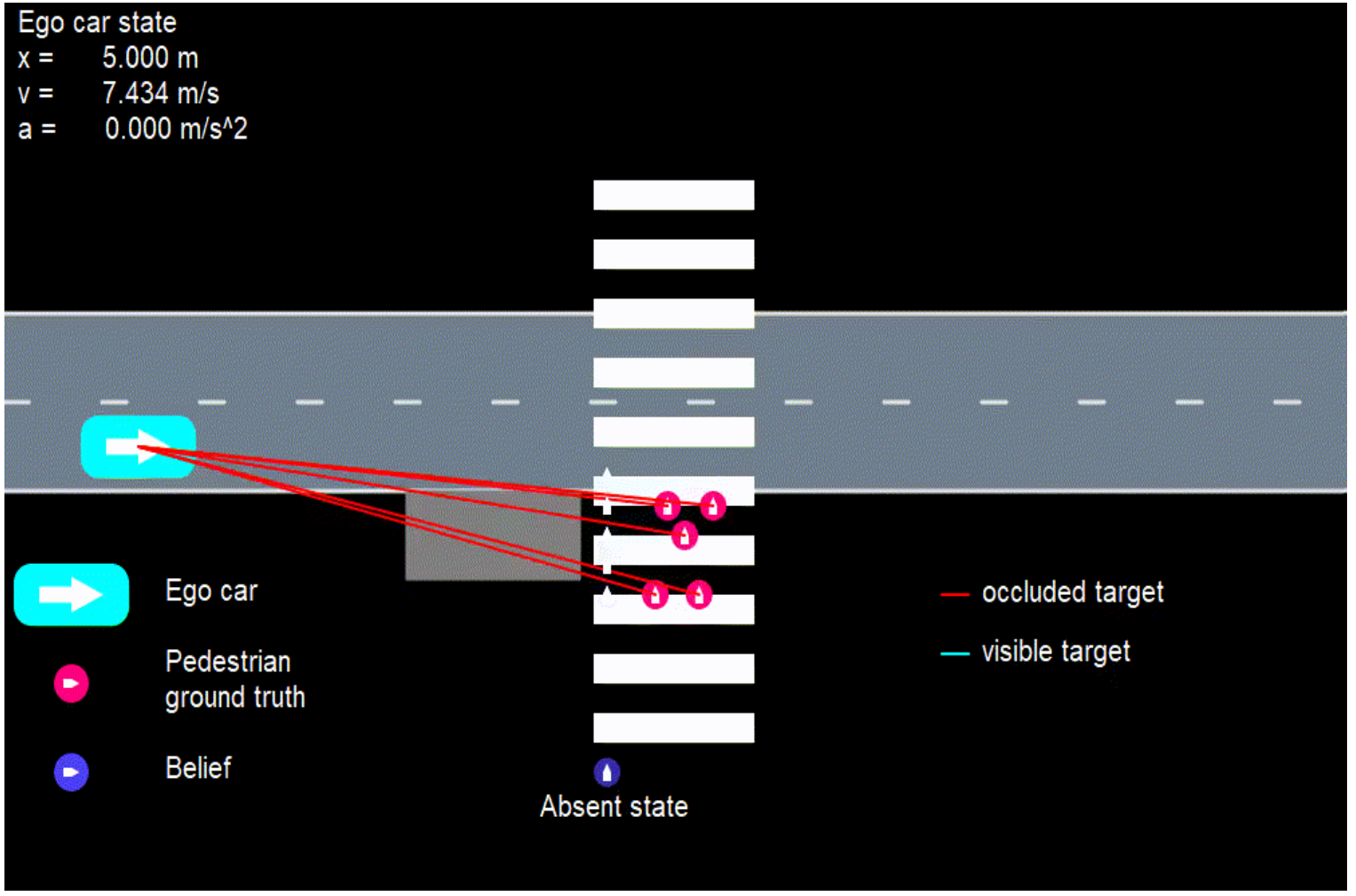
$$a = \text{argmax}_a \min_i (Q_{\text{single}}(b_i, a))$$



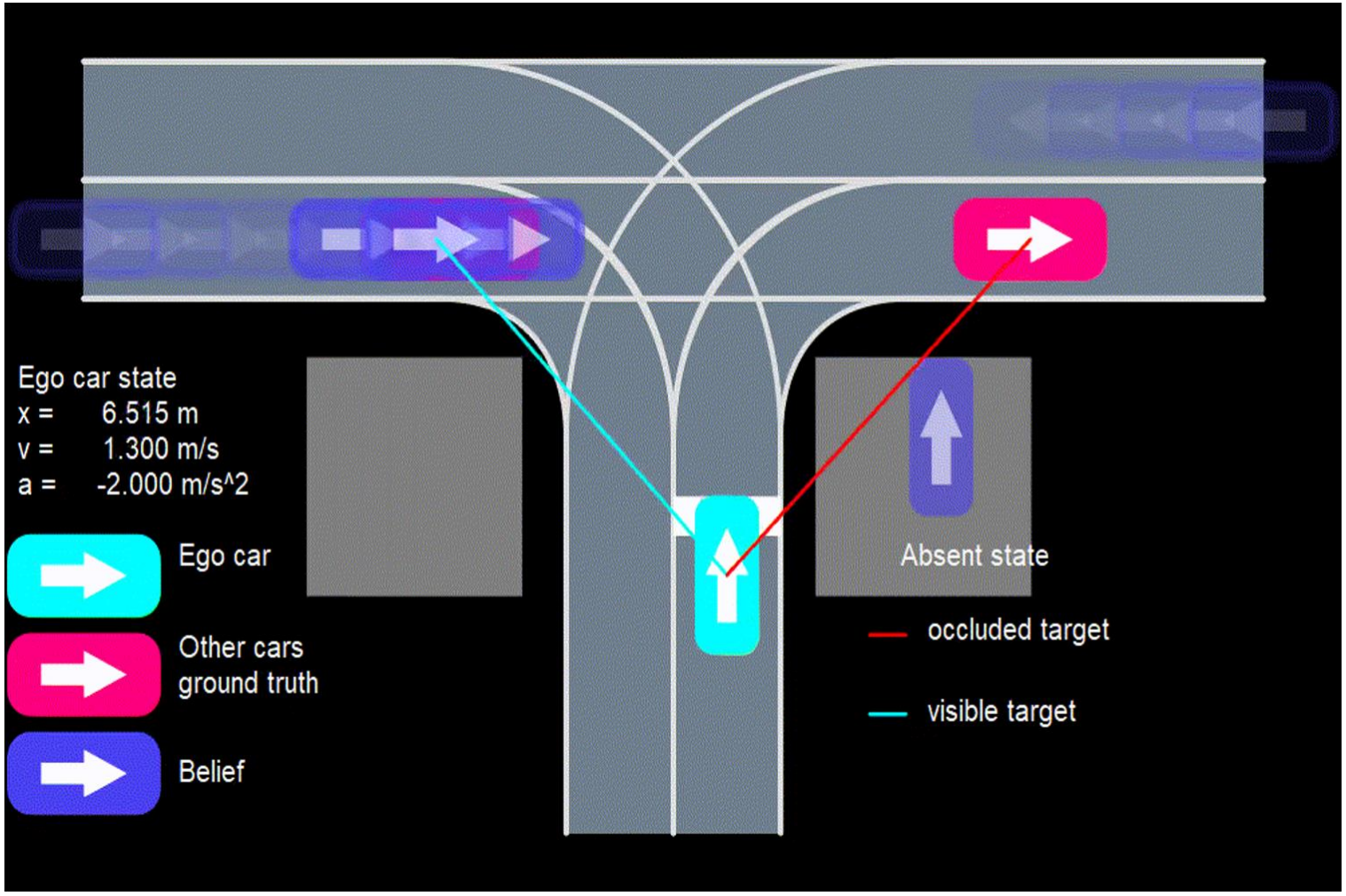
Occluded Crosswalk Demo



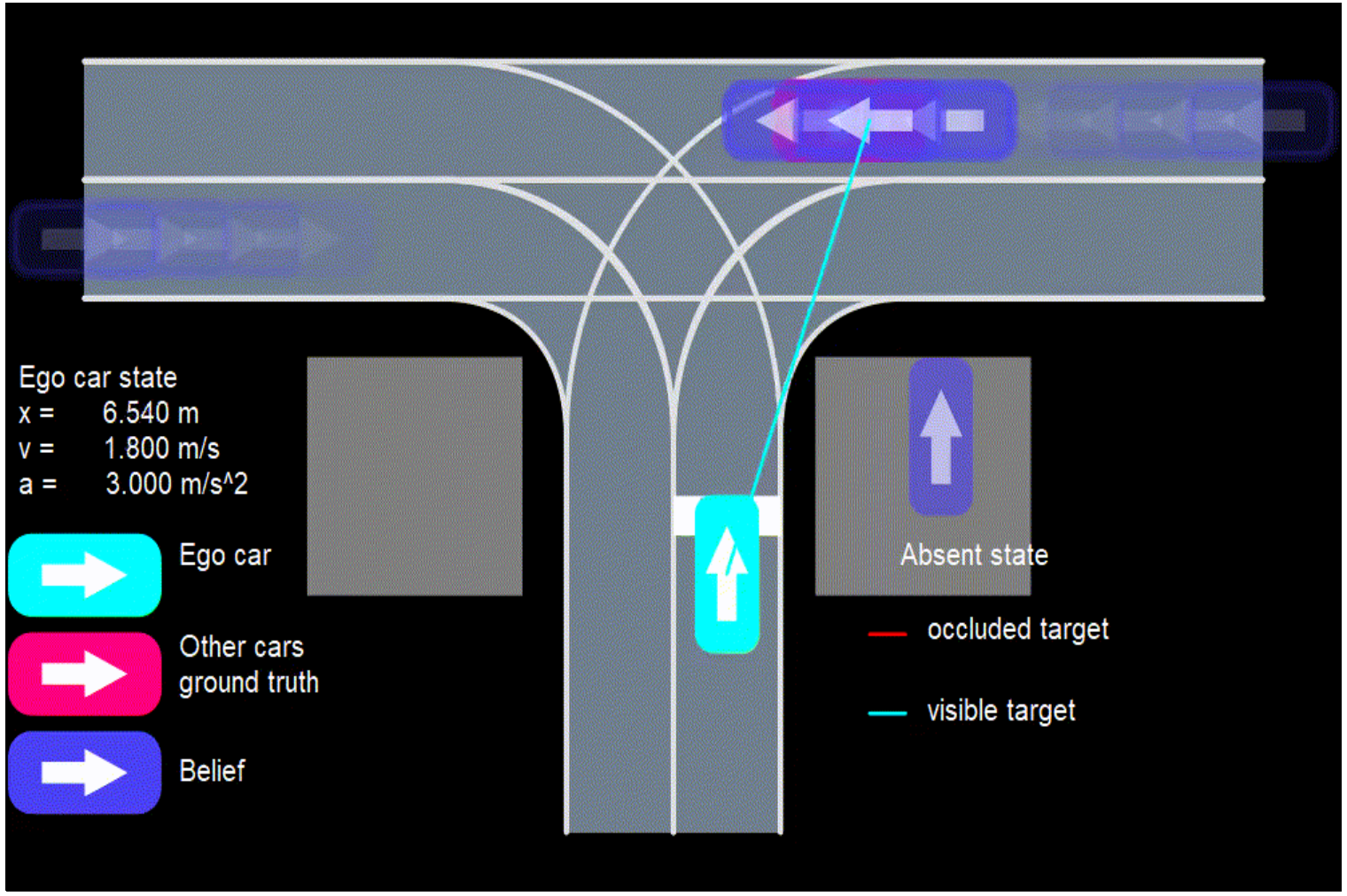
Occluded Crosswalk Demo



Occluded Intersection Demo

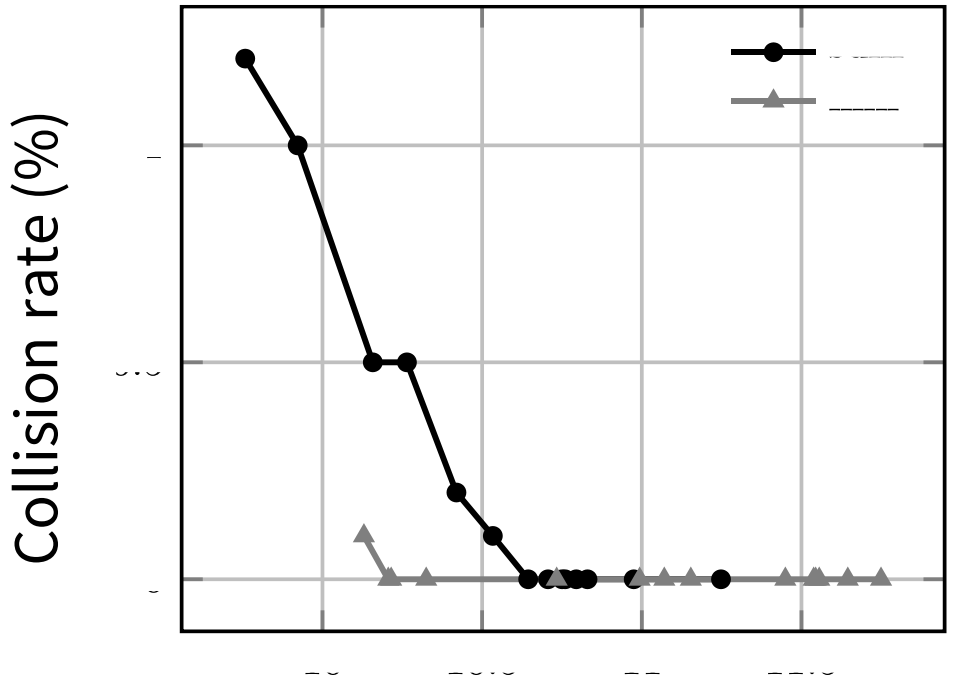


Occluded Intersection Demo

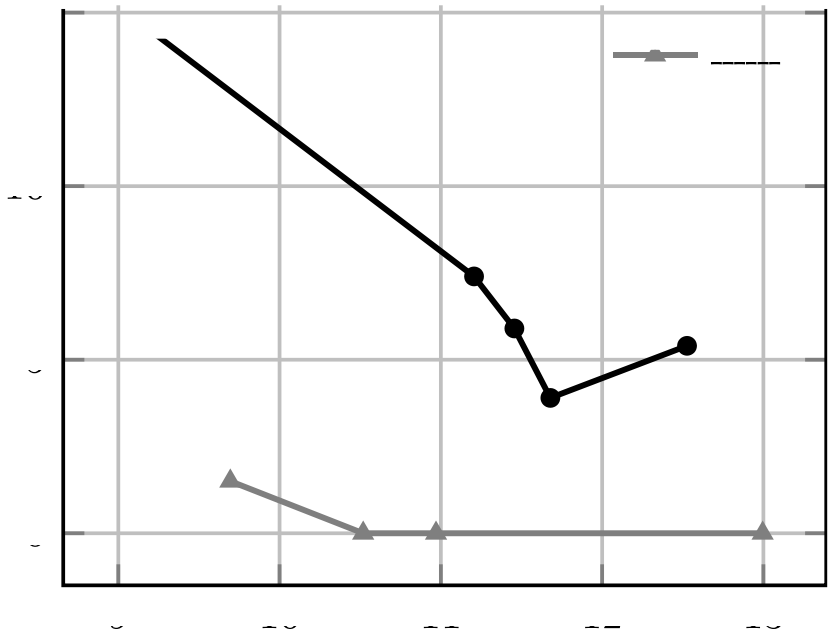


Results

QMDP



SARSOP



- Collision cost is varied
- Reward tuning allows to get safe policies
- The choice of the fusion function matters a lot

Results

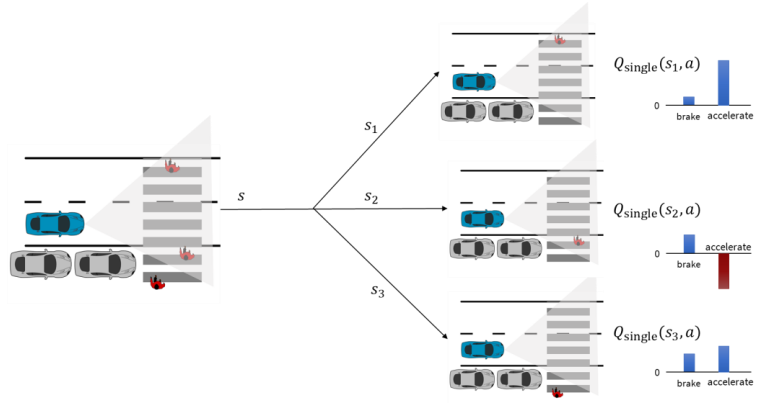
	Collision rate (%)	Time to cross (%)
Crosswalk		
Baseline	0.1 ± 0.04	18.58 ± 5.39
QMDP	0.0 ± 0.0	10.61 ± 3.76
→ SARSOP	0.0 ± 0.0	10.51 ± 4.44
Intersection		
Baseline	0.1 ± 0.04	13.46 ± 3.04
→ QMDP	0.0 ± 0.0	6.20 ± 2.108
SARSOP	0.7 ± 0.83	4.38 ± 0.1342

Discussion

1. POMDP approach can handle sensor occlusions and stochastic behaviors and outperforms rule-based methods.

2. Reward tuning allows to generate different behavior

3. Decomposition methods highly reduce the computational cost of POMDP algorithms



Drawbacks:

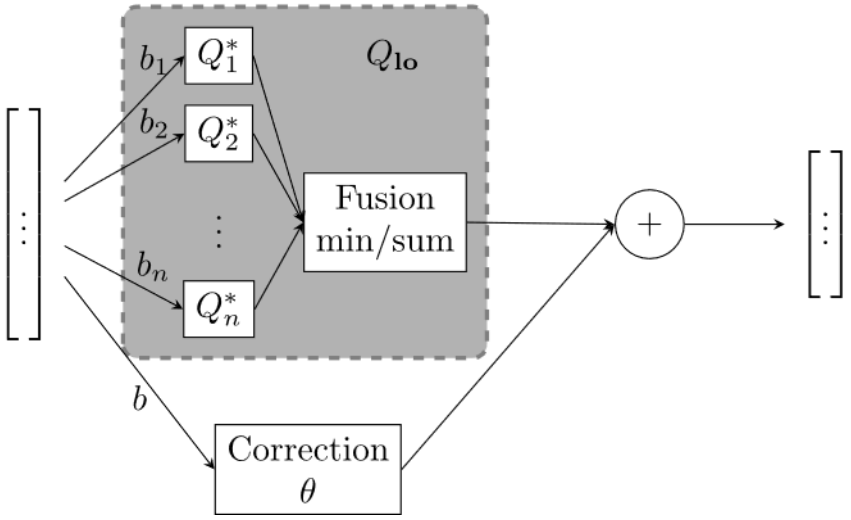
- Rough approximation
- Makes strong assumptions of independence between traffic participants

1.Introduction
2.Utility Decomposition

3.Deep Corrections

4.Safe Reinforcement Learning with Scene Decomposition
5.Model Checking in POMDPs
6.Conclusion

- Learn a **corrective term**, represented by a neural network, to refine the approximation
- Faster learning
- Better performance.



Utility Decomposition with Deep Corrections

- Utility decomposition performs well empirically ☺
- **Sacrifices optimality** ☹
- Learn a corrective term :

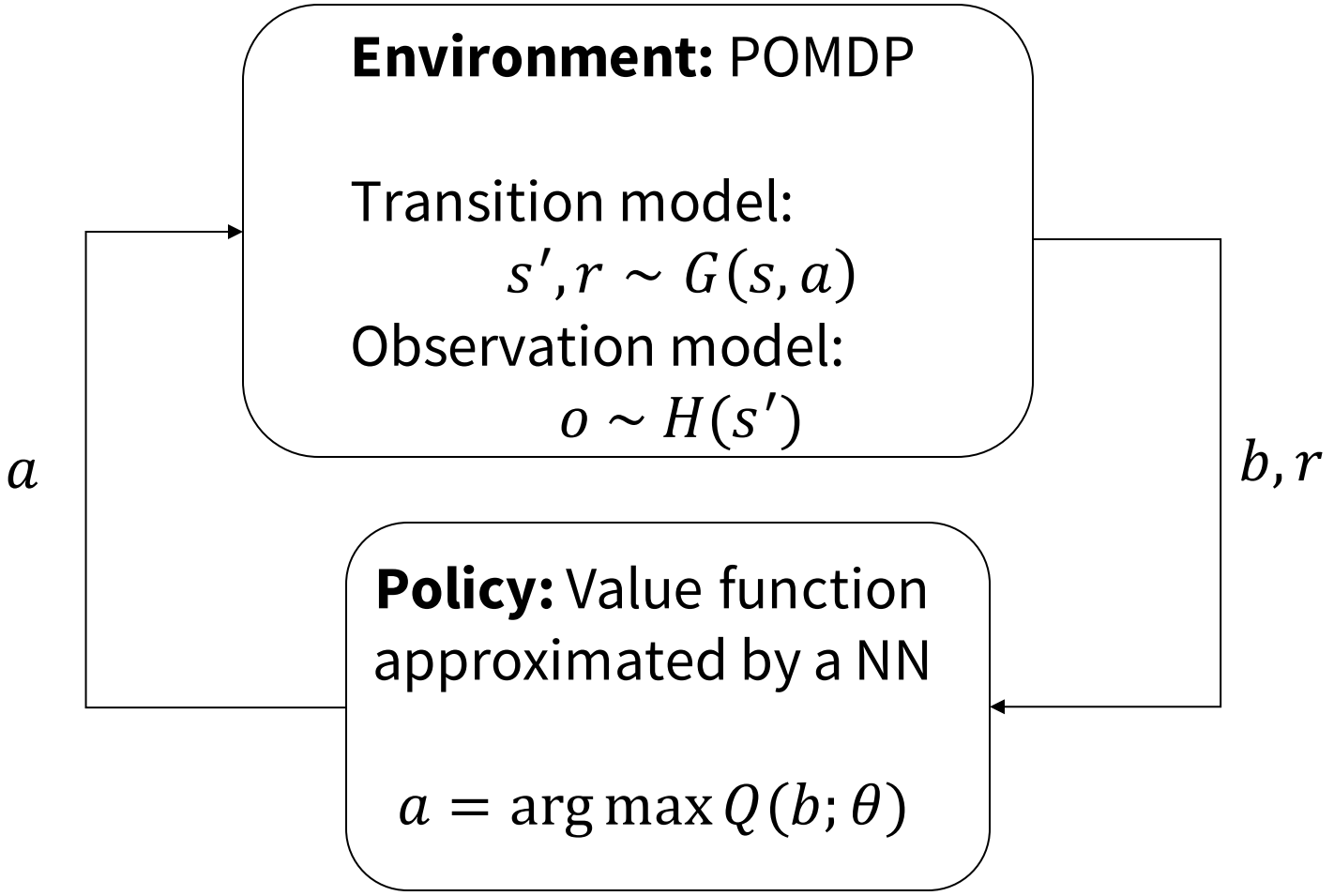
$$Q^*(b,a) \approx Q_{low}(b,a) + \delta(b,a\,;\theta) = \sum_i Q_{single}(b_i,a\,;\theta') + \delta(b,a;\theta)$$

Obtained via utility decomposition or prior knowledge

Learnt using Deep Q learning

Reinforcement Learning

Requires: Black box simulator (model free RL), reward function: $R(s, a)$

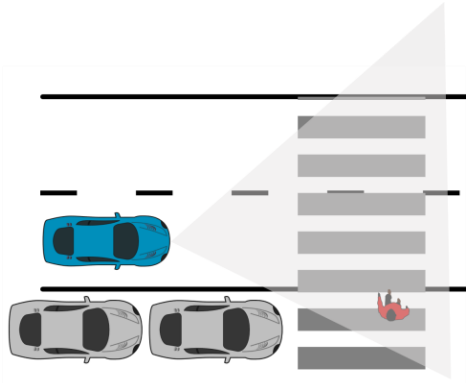


Train the Corrective Factor

$$Q^*(b,a) \approx \sum_i Q_{single}(b_i, a\,; \theta') + \delta(b,a;\theta)$$

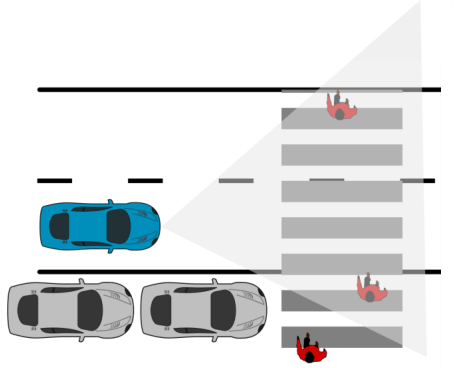
Step 1:

Learn Q_{single} in an environment with one entity



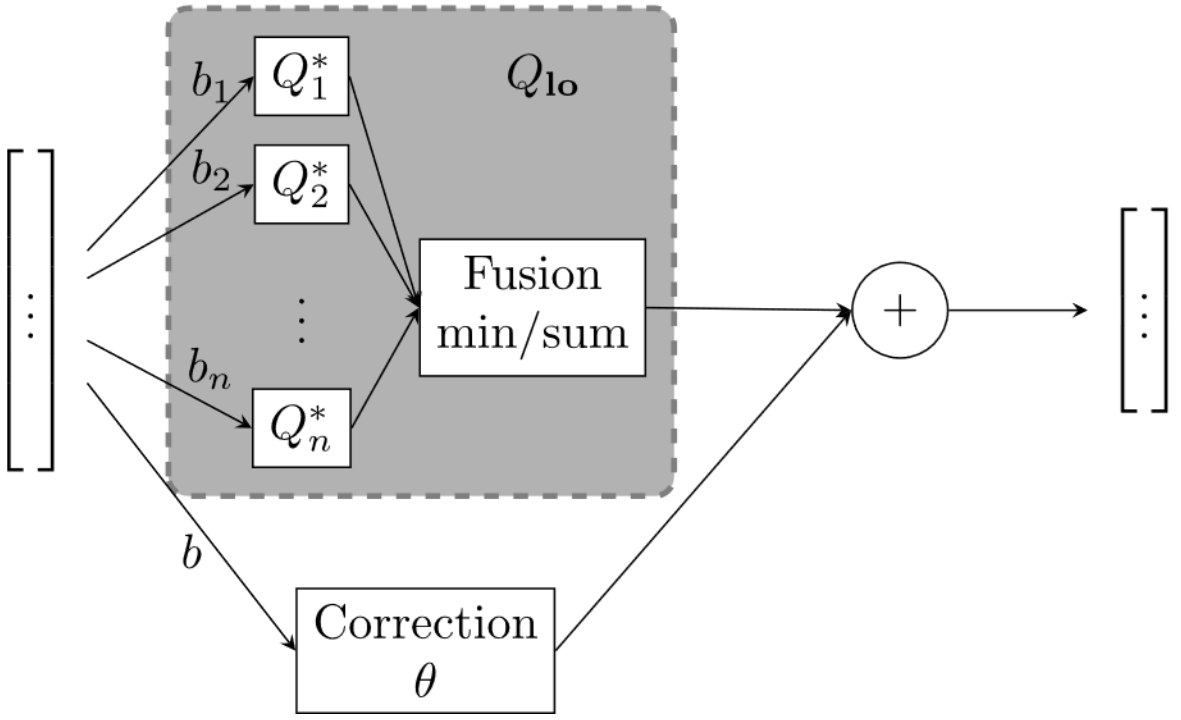
Step 2:

Learn δ in the full environment

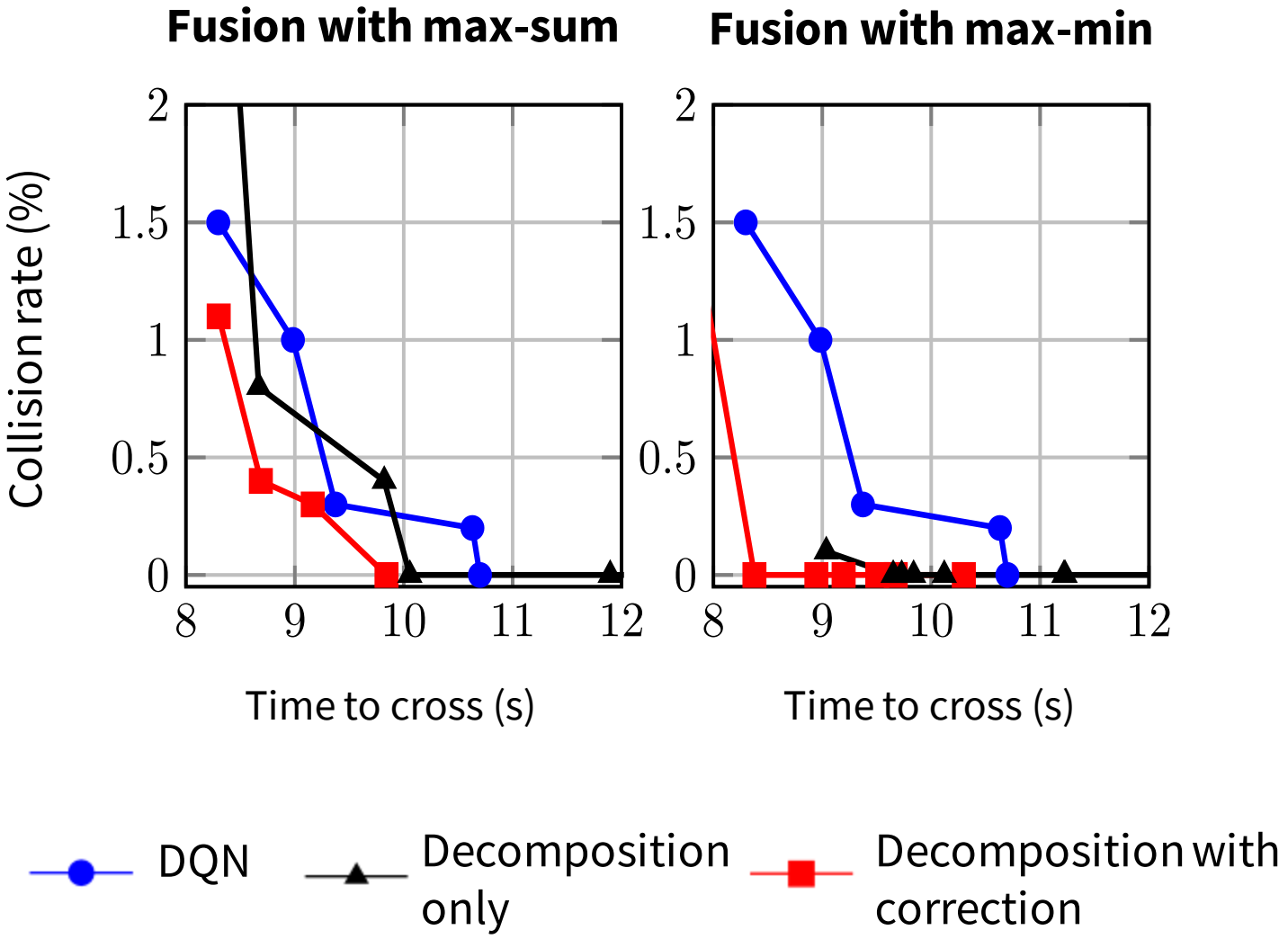


Train the Corrective Factor

- Using deep Q-learning
- Freeze the weights of Q_i
- Propagate the gradients through the correction term only



Results



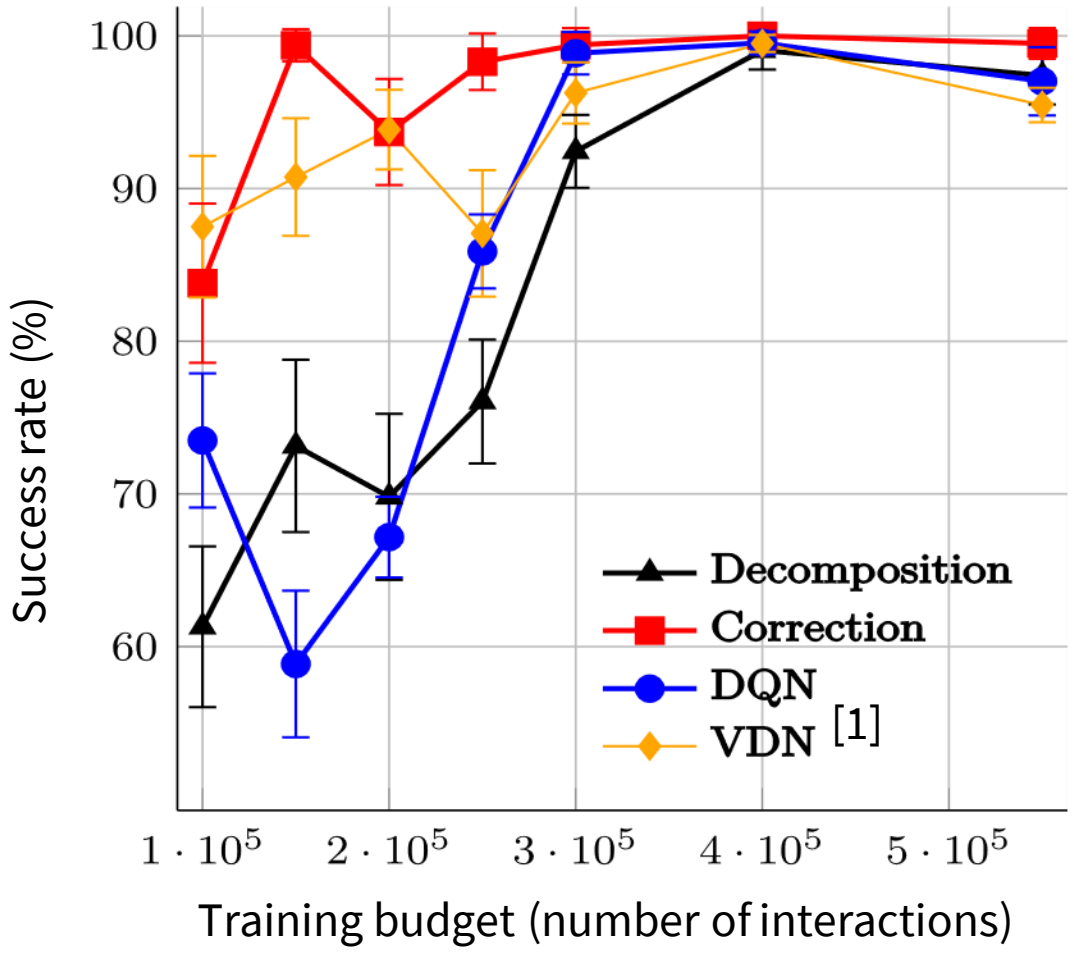
DQN: Solving the full problem with a deep Q network

Decomposition only: Applying utility fusion

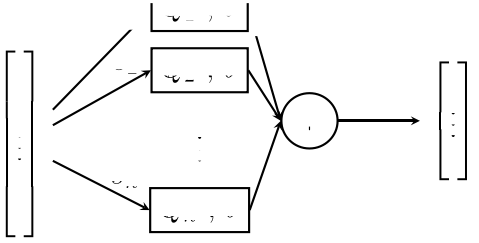
Decomposition with correction: Adding a corrective term to the utility fusion solution

=> **Domination of the correction method**

Results



- The correction function is being trained on **twice as less samples** than the regular DQN policy.
- Converges faster
- **Leveraging prior knowledge allows to significantly reduce the time spent for exploration**



Value decomposition network structure

[1] P. Sunehag, et al, "Value-Decomposition Networks for Cooperative Multi-Agent Learning Based on Team Reward," in AAMAS, 2018.

Discussion

1. Improve the approximation from utility decomposition by **learning a corrective term**

2. Learn the corrective term through deep reinforcement learning

3. Learns faster than other methods

4. Outperforms policies trained from scratch or using decomposition method only

Related Work:

- T. Silver et al, “Residual policy learning”, Arxiv 2018.

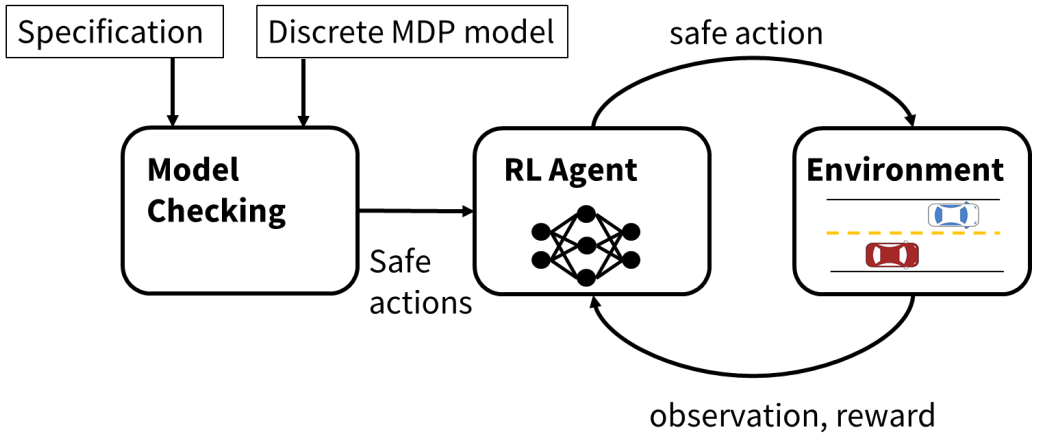
$$\pi_\theta(s) = \pi(s) + f_\theta(s)$$

1.Introduction
2.Utility Decomposition
3.Deep Corrections

4.Safe Reinforcement Learning

5.Model Checking in POMDPs
6.Conclusion

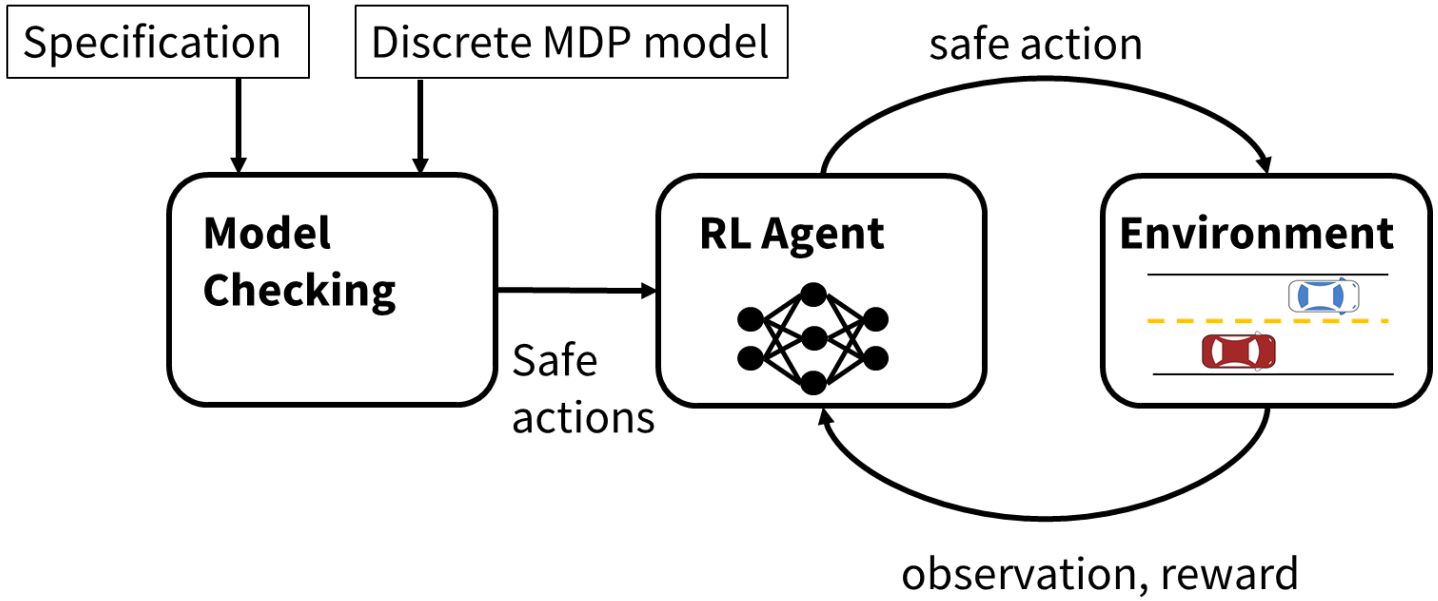
- Rely on a model checker to enhance safety.
- Combine model checking with the decomposition method



Safe Reinforcement Learning using Model Checking

- **Model Checking** allows to verify property of systems with high confidence
  - Given a safety specification
  - Identify actions satisfying the specification using **Model Checking**

- **Constrain** the action space to enforce the satisfaction of the safety specification.
- Train an RL agent in a **higher fidelity** environment



Identifying Acceptable Actions

$P_s(s, a)$: probability of reaching the goal **safely** when taking action a in state s

Given a minimum acceptable probability of success γ, the set of acceptable action is given by:

$$A(s) = \{ a \mid P_s(s, a) > \gamma \}$$

$P_S(s, a)$ can be computed using value iteration (polynomial in the number of states)

Note: Assume full observability and approximate $P_S(b, .) \approx \sum_s b(s) P_S(s, .)$

Constrained Exploration

In state s

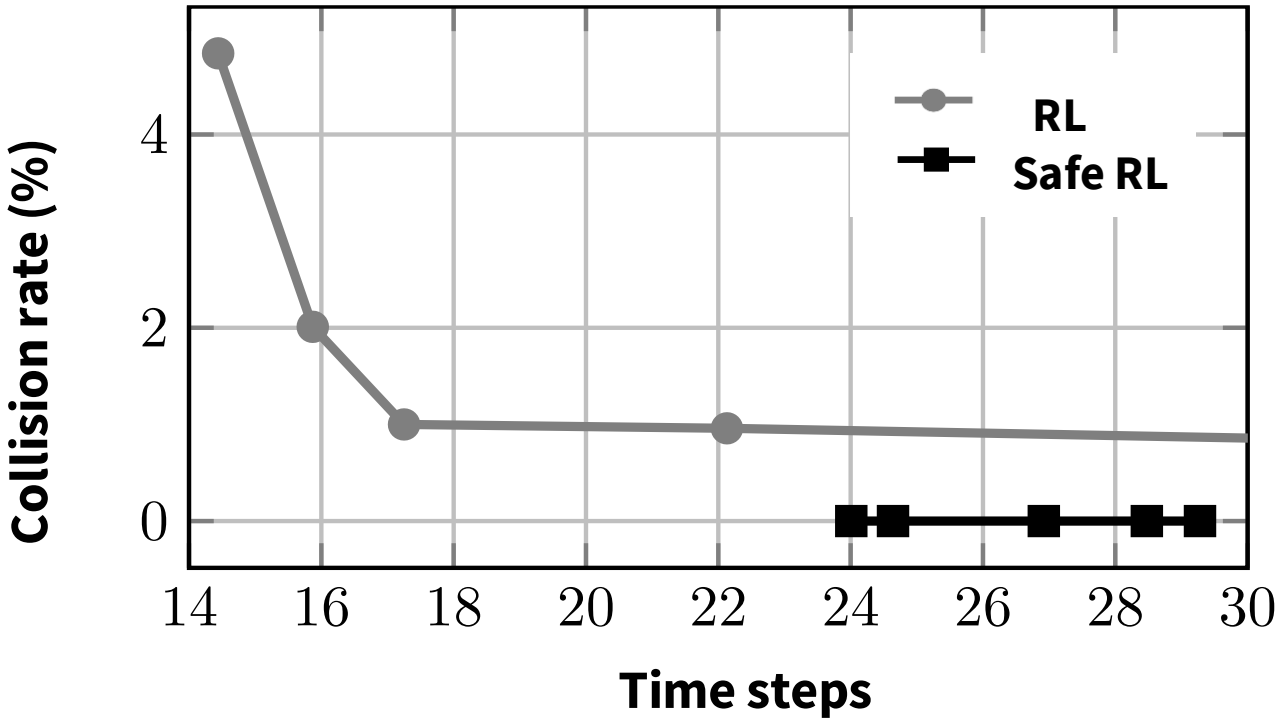
- If $A(s)$ is empty
  - Default to $\pi_{\text{safe}}(s) = \text{argmax}_{\text{a}}\ P_S(s, a)$
- Else
  - Choose action in $A(s)$ using any exploration policy (e.g. ϵ-greedy)

Guaranteed: never prioritize unsafe actions over safe actions

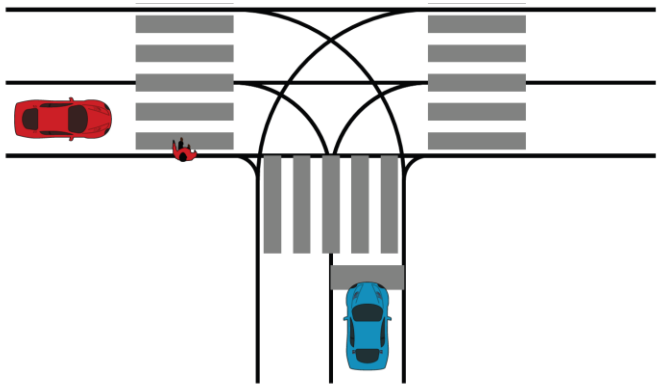
Not guaranteed: the policy has a probability of success of at least γ

Training with constraints

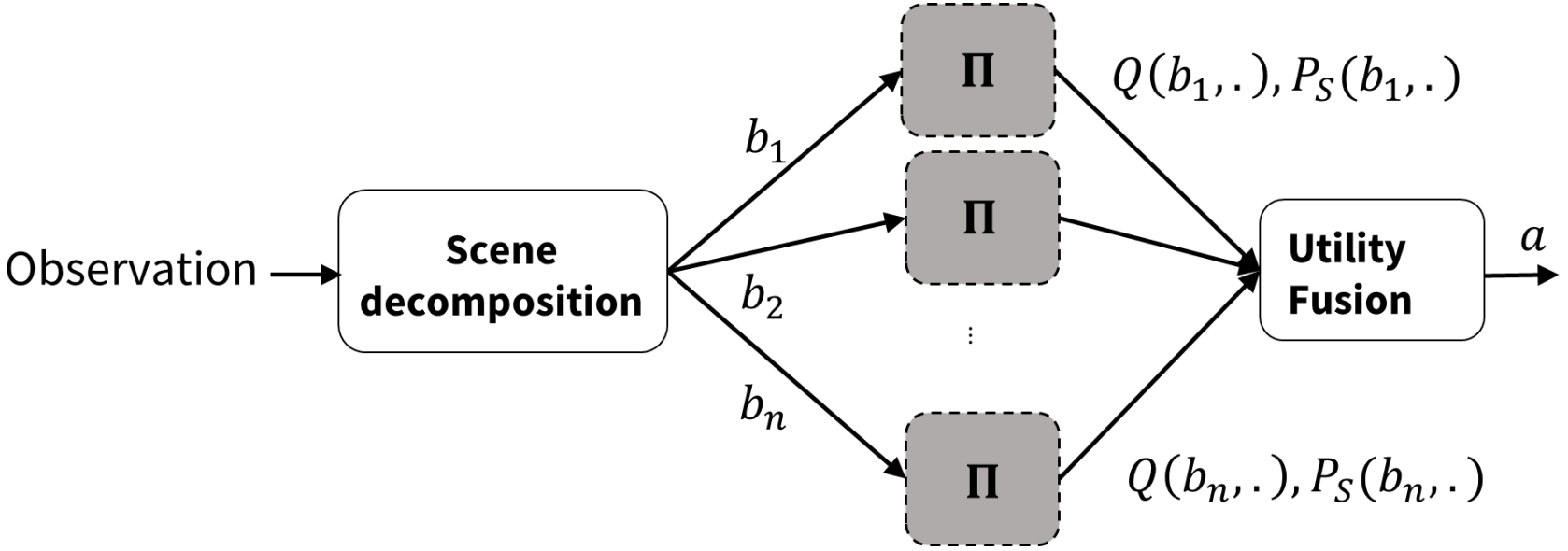
- Our approach moves the Pareto frontier towards safe regions of the operation space
- Decoupling of the two objectives simplify the design process.



Scenario (fully observable):



For each subproblem compute P_s and Q



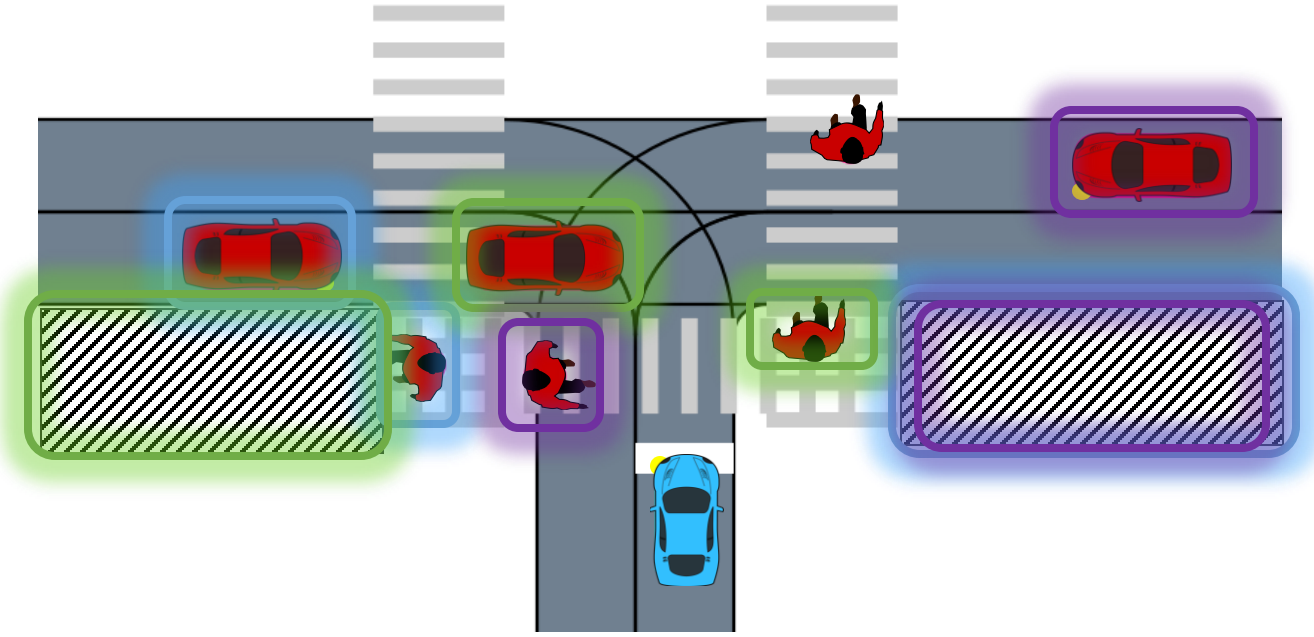
$$P_S(b,a) = \min_i P_S(b_i,a)$$

$$Q(b,a) = \min_i Q(b_i,a)$$

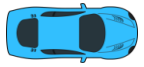
Revisiting the Decomposition Method

Consider a subset of agents instead of considering only one agent

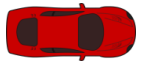
Example of three canonical scenarios in a complex scene.
There are 32 instances of canonical scenarios in this scene ($N_{\text{pedestrians}} \times N_{\text{cars}} \times N_{\text{obstacles}}$)



Canonical Scenarios



Ego Vehicle



Other Vehicle

Pedestrian

Safe Reinforcement Learning with Scene Decomposition

1. Find suitable state decomposition

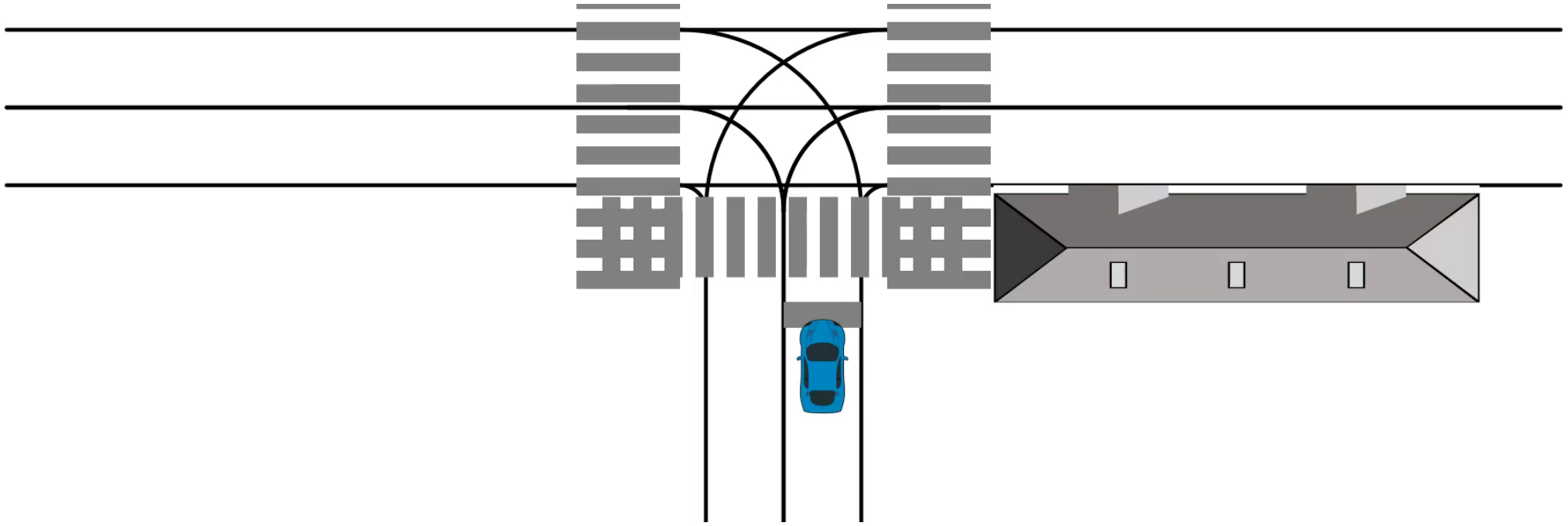
2. Compute the probability of success for the sub-problem

Model Checking

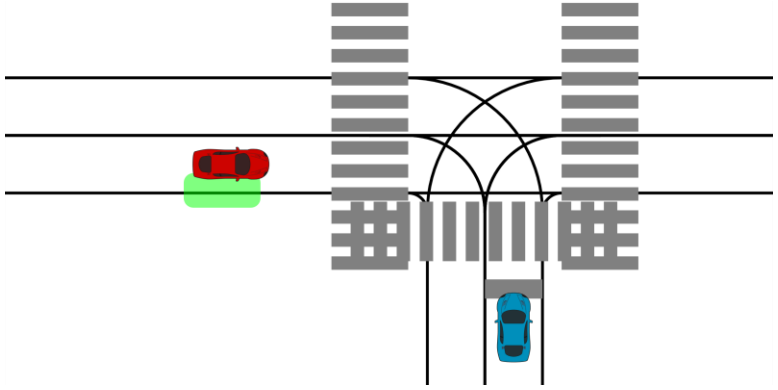
3. Compute the value function for the sub-problem

RL with constrained exploration

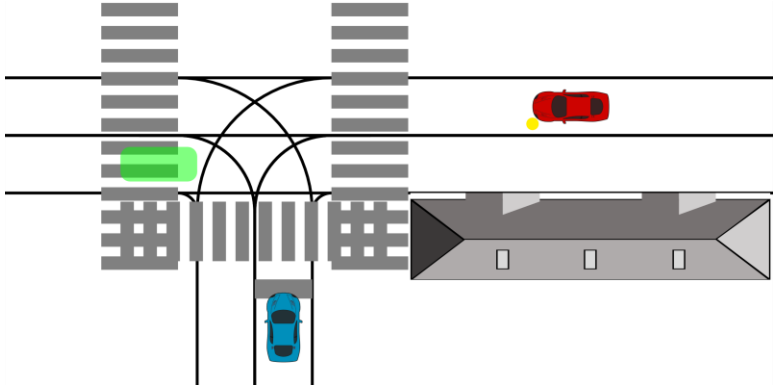
4. Online:
   - Belief update to estimate the state
   - apply a conservative fusion algorithm (min)



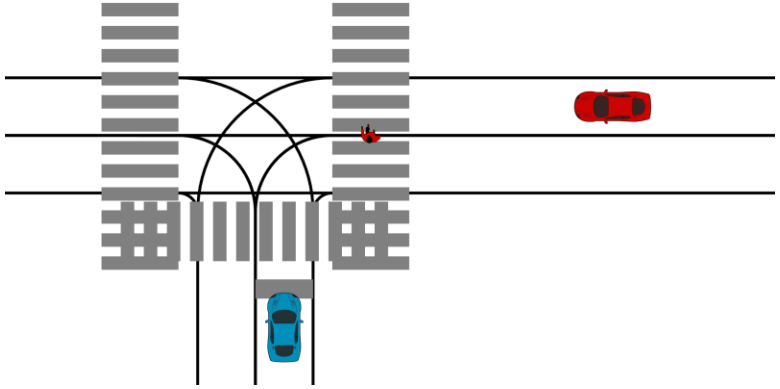
Evaluations Scenarios



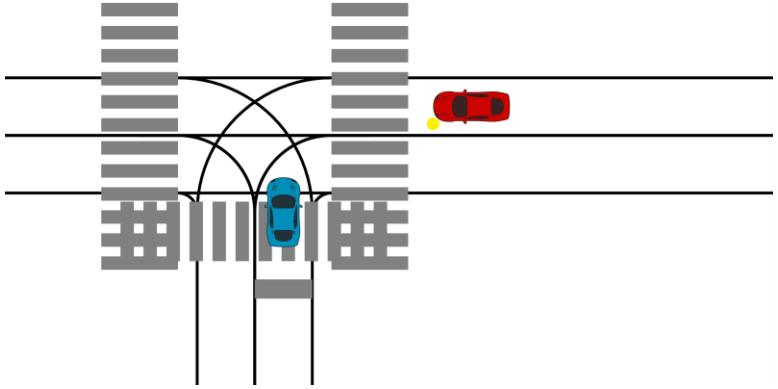
a: only one other car, sensor noise



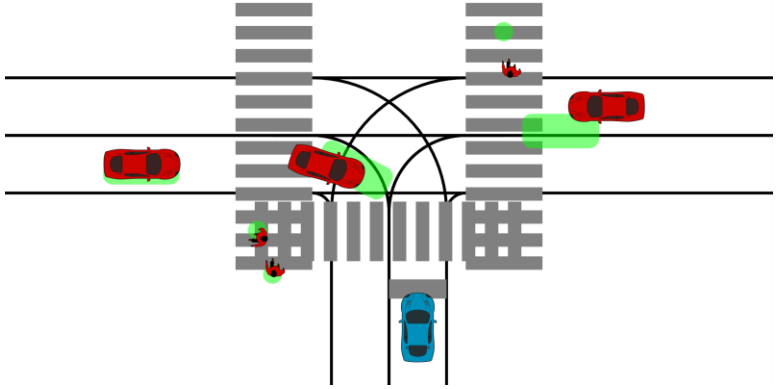
b: only one other car, sensor noise + occlusions



c: car and pedestrian interaction, perfect observation



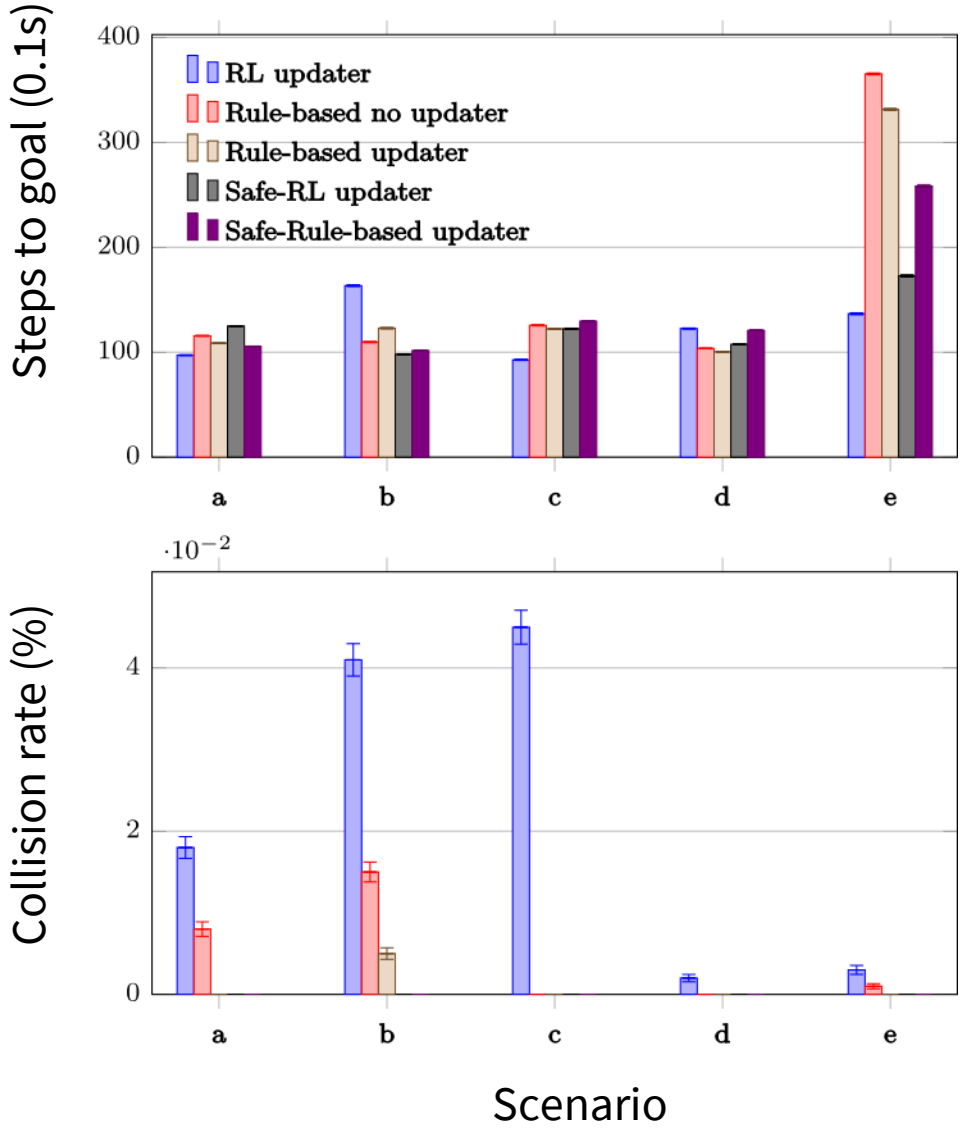
d: car and ego interaction, perfect observation



e: multiple cars, pedestrians, sensor noise + occlusions.

Discussion

1. Rule-based method perform well on simple scenarios
2. Relying on model-checking improves safety
3. Belief state planning improves robustness to sensor uncertainty
4. RL is better for complex scenarios (e)
5. Scene decomposition allows to scale to a large number of traffic participants



1.Introduction
2.Utility Decomposition
3.Deep Corrections
4.Safe Reinforcement Learning

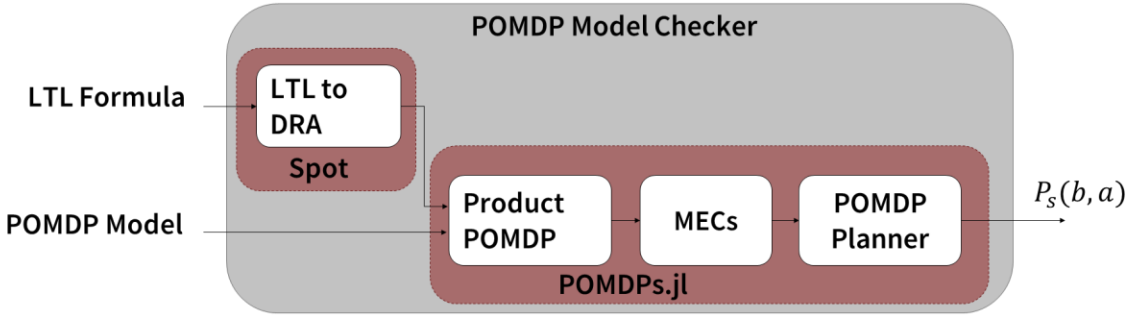
5.Model Checking in POMDPs

6.Conclusion

- Formulate the model checking problem as a **reward maximization** problem

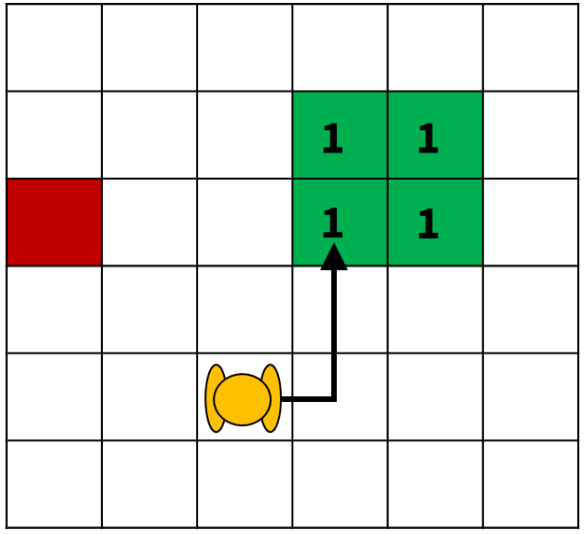
- Use approximate POMDP solvers to solve the problem

Main contribution



Constrained Reachability

What is the probability to reach a given set of states while avoiding some other?



Compute $P_s(s, a)$ for all states and actions

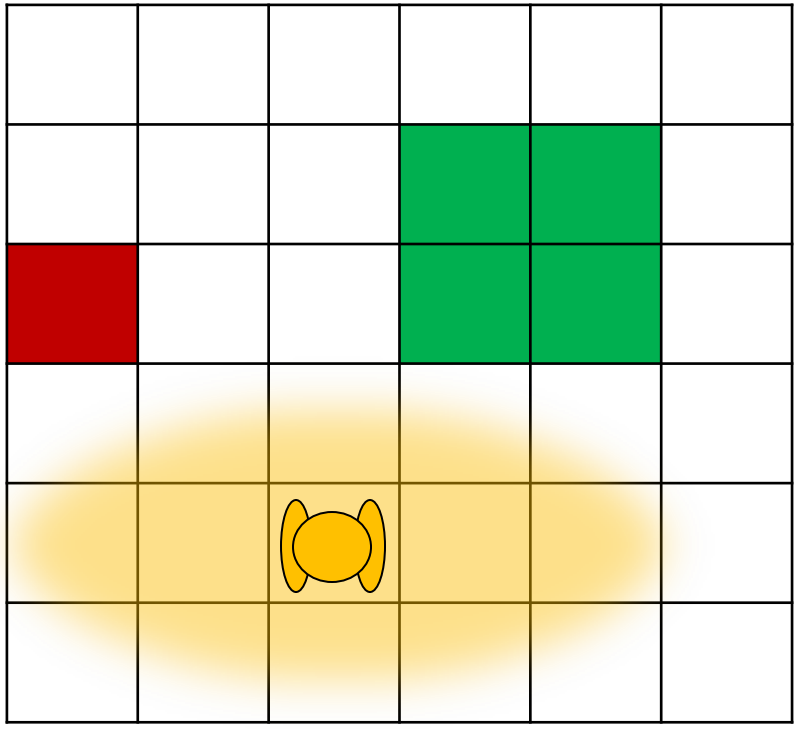
Reward function for reachability:
Desired set B

$$R_{\text{reach}}(s) = 1 \text{ if } s \in B$$
$$R_{\text{reach}}(s) = 0 \text{ otherwise}$$

Add absorbing states:
If $s \in B$, s is absorbing
Avoid set is absorbing

Constrained Reachability in a POMDP

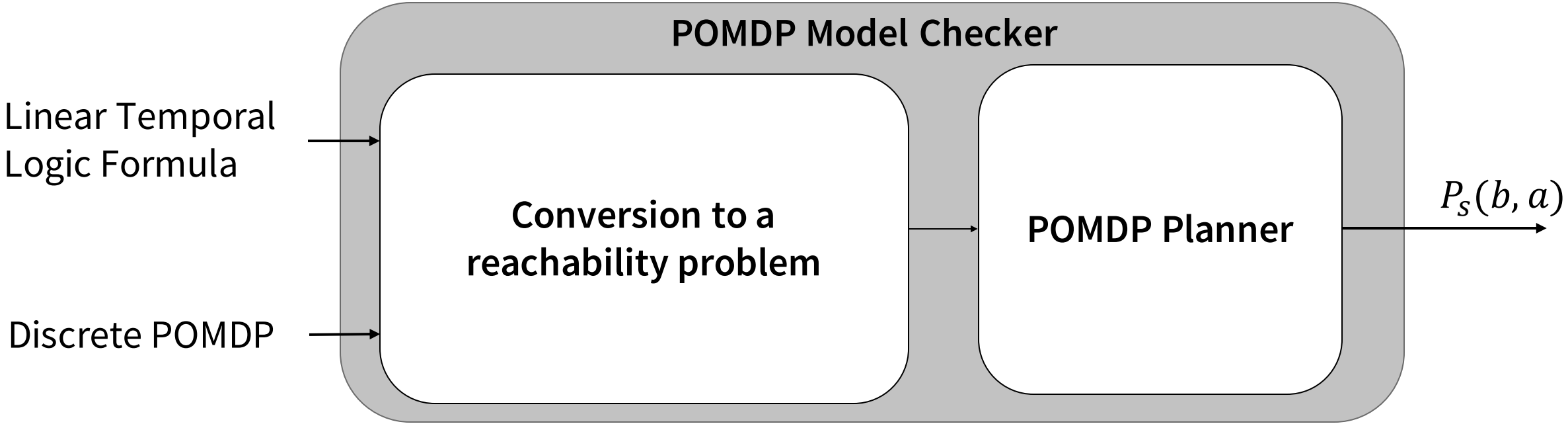
Compute the probability of success from any **belief**: $P_S(b, a)$



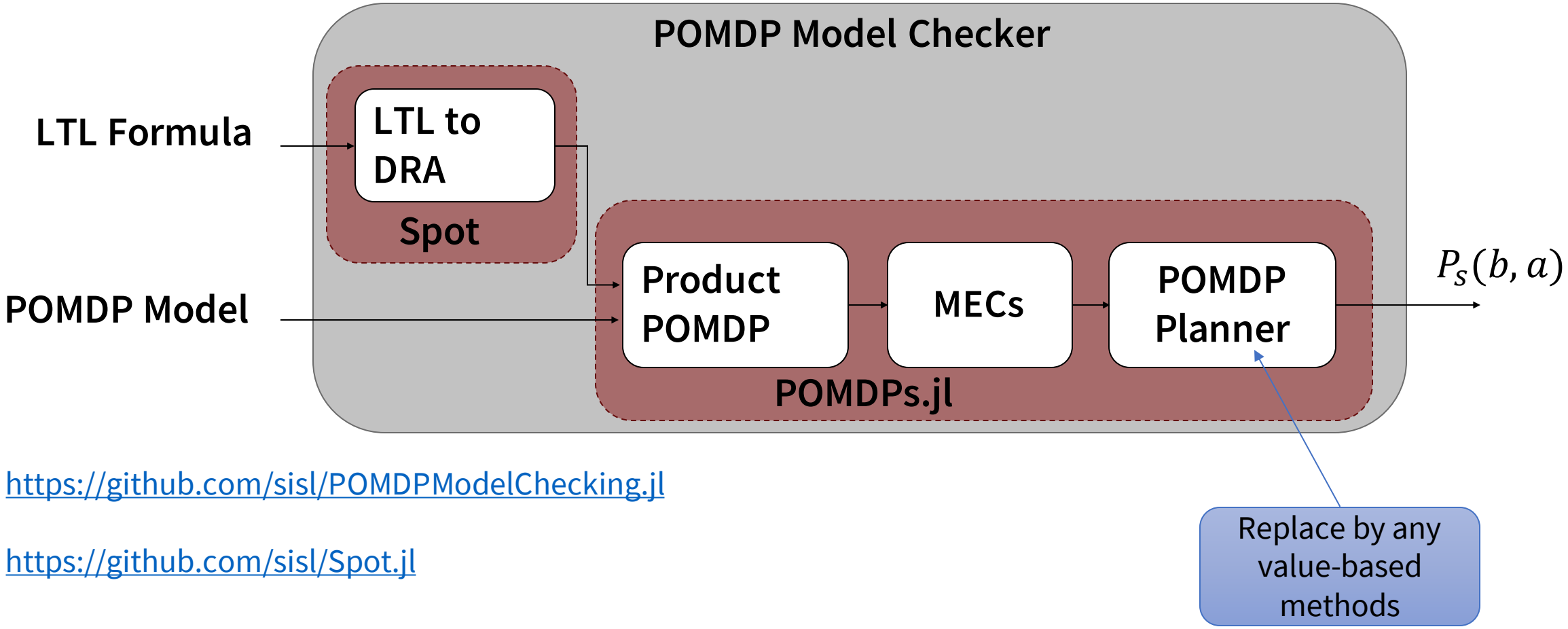
$P_S(b, a)$ is analogous to $Q(b, a)$ in a POMDP where $R = R_{\text{reach}}$

G. Norman, D. Parker, and X. Zou, “Verification and control of partially observable probabilistic systems,” in Real-Time Systems, 2017.

Model Checking in POMDP overview



Implementation

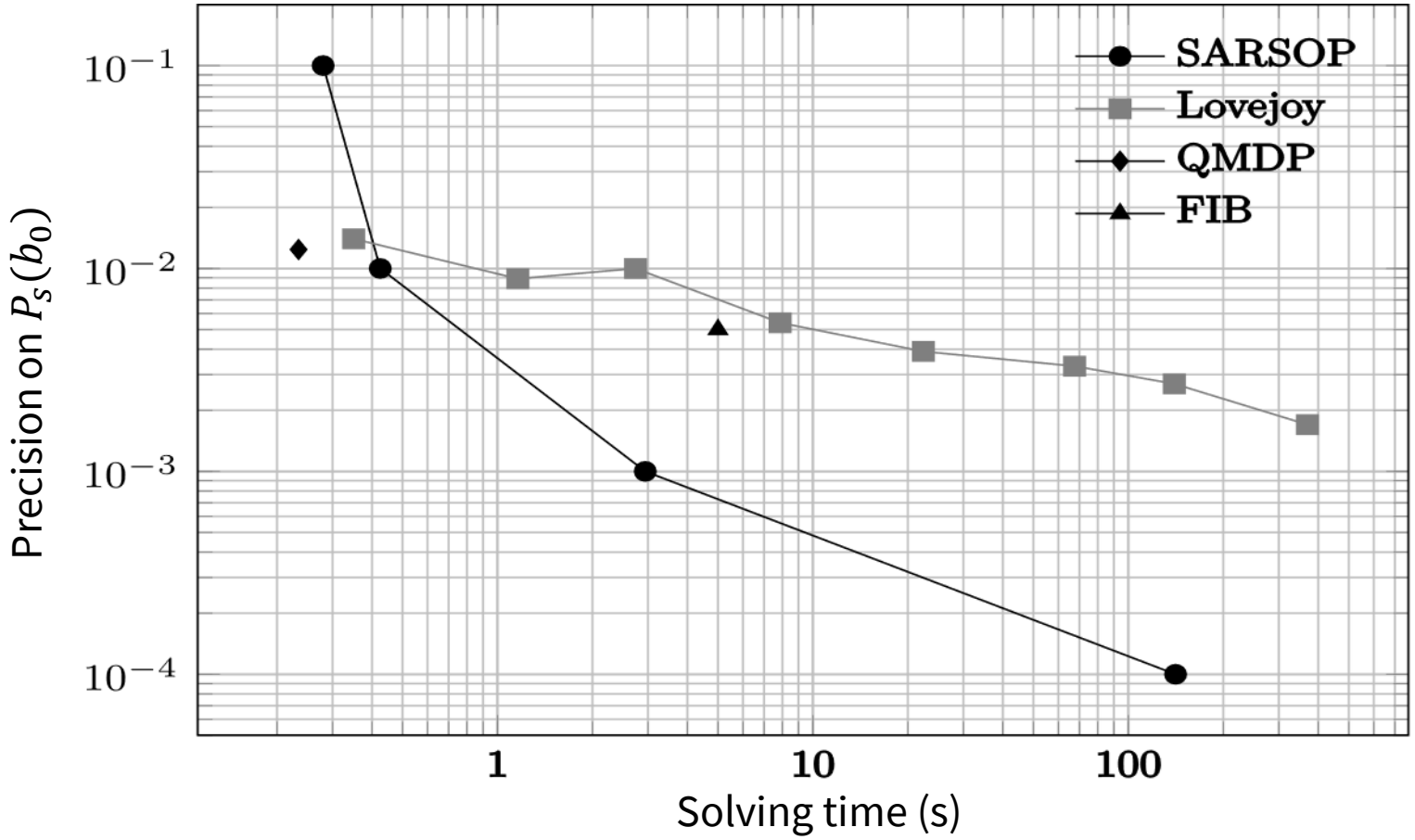


https://github.com/sisl/POMDPModelChecking.jl

https://github.com/sisl/Spot.jl

A. Duret-Lutz, A. Lewkowicz, A. Fauchille, T. Michaud, E. Renault, and L. Xu, "Spot 2.0 – a framework for LTL and ω-automata manipulation," in Automated Technology for Verification and Analysis (ATVA), ser. Lecture Notes in Computer Science, 2016.
M. Egorov, Z. N. Sunberg, E. Balaban, T. A. Wheeler, J. K. Gupta, and M. J. Kochenderfer, "POMDPs.jl: A framework for sequential decision making under uncertainty," Journal of Machine Learning Research, 2017.

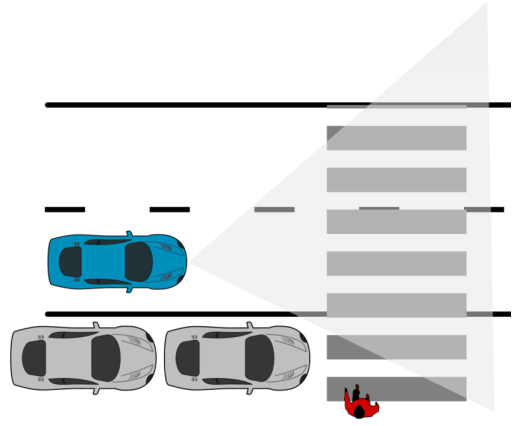
Results

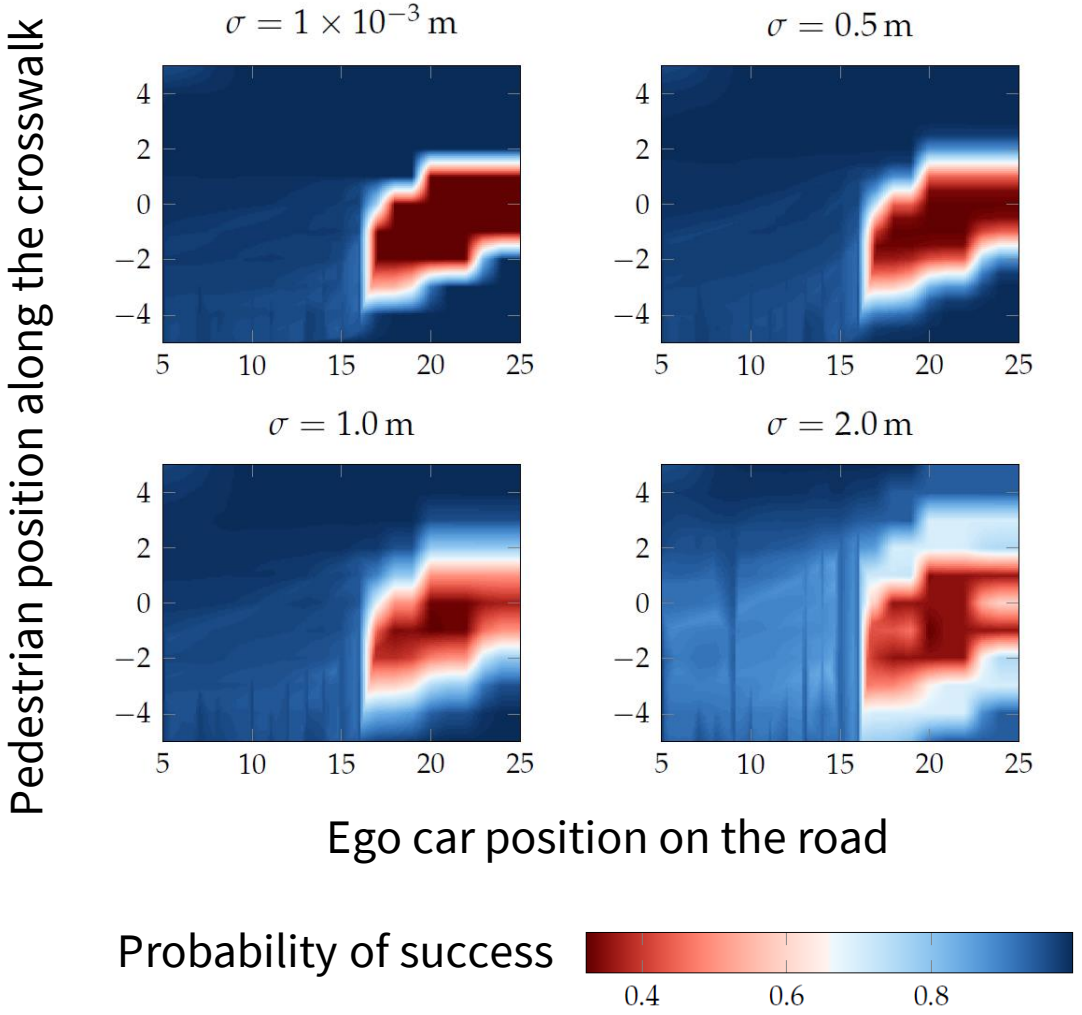


- Better precision than previous work
- Solve model checking in larger POMDPs that were intractable before

Occluded Crosswalk

- Probability of success for every **belief**
- As **uncertainty** increases, the probability of success varies
- It takes into account the possibility that in the future we will have a better observation.





Discussion

1. Model Checking problems can be solved as reachability problem in POMDPs
2. POMDP planners like SARSOP can be used for model checking
3. This methodology provides state-of-the-art performance
4. Limited to discrete models for now

6. Conclusion

Conclusions

1. **Belief state planning** is robust to sensor occlusions and perception uncertainty. Policies tend to outperform rule-based methods.

2. **Decomposition methods** improves the scalability of POMDP solvers. They can be augmented using deep reinforcement learning through the **deep corrections** technique.

3. **Model checking** can help improving safety of reinforcement learning policies.

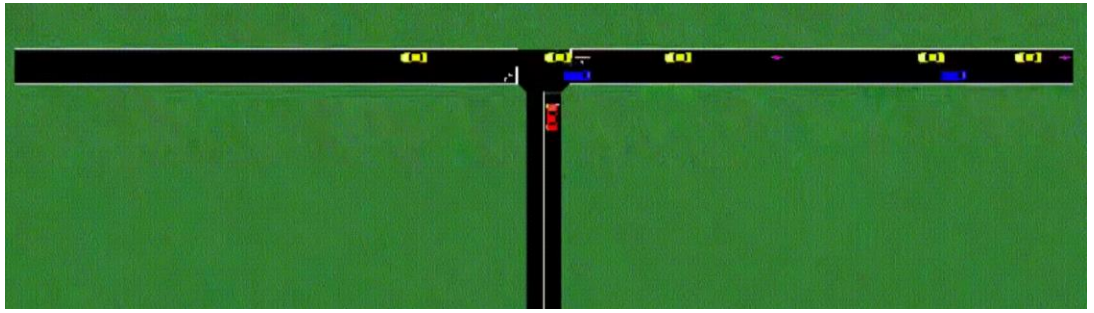
4. **POMDP Planners** can solve model checking problems.

Future Work

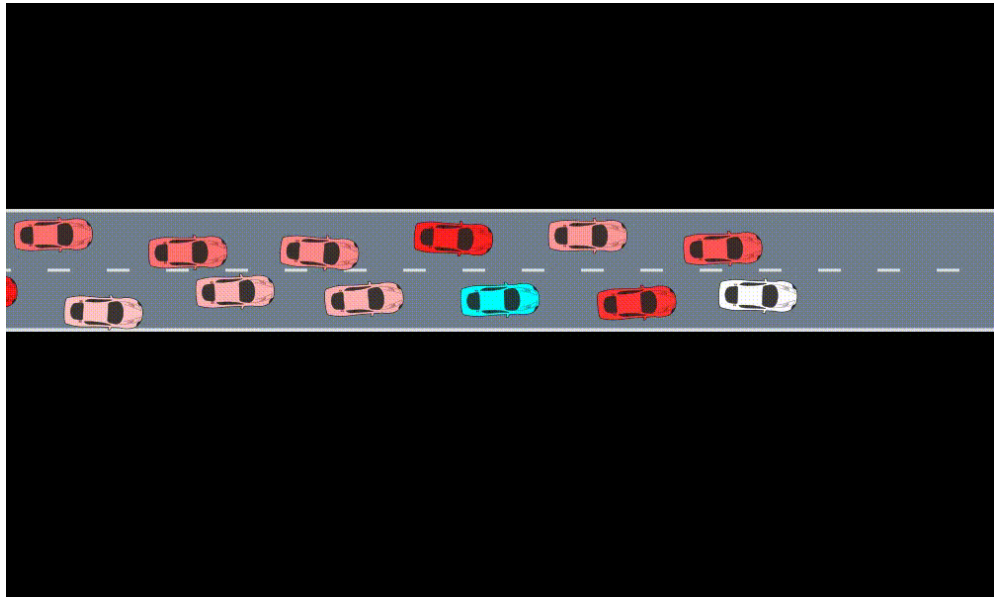
- **Real-world testing**
  - POMDPs in the real world: K. Wray et al. "Online decision-making for scalable autonomous systems." *International Joint Conference on Artificial Intelligence*. 2017.
- **Multi-agent Reinforcement Learning**
  - Learning cooperative policies
- **Combining model-based planning with data driven models**
  - Model based approaches could benefit from using more sophisticated models

Other works

- Online planning for navigating intersections



- Learning to merge in dense traffic



Publications

Online planning and modeling:

- M. Bouton, A. Cosgun, and M. J. Kochenderfer, "Belief state planning for autonomously navigating urban intersections," in *IEEE Intelligent Vehicles Symposium (IV)*, 2017.

Decomposition Methods:

- M. Bouton, K. D. Julian, A. Nakhaei, K. Fujimura, and M. J. Kochenderfer, "Decomposition methods with deep corrections for reinforcement learning," *Autonomous Agents and Multi-Agent Systems*, vol. 33, iss. 3, p. 330–352, 2019.
- M. Bouton, A. Nakhaei, K. Fujimura, and M. J. Kochenderfer, "Scalable decision making with sensor occlusions for autonomous driving," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018.
- M. Schratter, M. Bouton, M. J. Kochenderfer, and D. Watzenig, "Pedestrian collision avoidance system for scenarios with occlusions," in *IEEE Intelligent Vehicles Symposium (IV)*, 2019.

Safe Planning:

- M. Bouton, J. Karlsson, A. Nakhaei, K. Fujimura, M. J. Kochenderfer, and J. Tumova, "Reinforcement learning with probabilistic guarantees for autonomous driving," in *Workshop on Safety Risk and Uncertainty in Reinforcement Learning, Conference on Uncertainty in Artificial Intelligence (UAI)*, 2018.
- M. Bouton, A. Nakhaei, K. Fujimura, and M. J. Kochenderfer, "Safe reinforcement learning with scene decomposition for navigating complex urban environments," in *IEEE Intelligent Vehicles Symposium (IV)*, 2019.

Interaction-aware planning:

- M. Bouton, A. Nakhaei, K. Fujimura, and M. J. Kochenderfer, "Cooperation-aware reinforcement learning for merging in dense traffic," in *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2019.

Model Checking

- M. Bouton, J. Tumova, M. J. Kochenderfer, "Point-Based Method for Model Checking in Partially Observable Markov Decision Processes", in *AAAI Conference on Artificial Intelligence (AAAI)*, 2020.

Sponsor


HRI
Honda
Research
Institute USA


WASP
WALLENBERG AI,
AUTONOMOUS SYSTEMS
AND SOFTWARE PROGRAM

Contributions

Modeling Autonomous Driving Problems

Online Planning at Intersections

Learning to Merge in Dense Traffic

Decomposition methods

Deep Corrections

Safe Planning

- Reinforcement learning with probabilistic guarantees
- Model checking in partially observable environments