Reinforcement Learning-Based Autonomous Driving at Intersections in CARLA Simulator

Rodrigo Gutiérrez-Moreno, Rafael Barea, Elena López-Guillén
, Javier Araluce and Luis M. Bergasa

University of Alcalá

Intersections problem

60% of severe traffic injuries in Europe a

29% of all car crashes and 18% of pedestrian fatalities

large amount of information

Developing an agent that allows

safe and reliable decisions is a hard task to implement manually

existing solutions

prediction and collaborations (V2V, V2I)

TTC (tuning parameters have to be adjusted, and this task can be laborious)

Reinforcement learning

Imitation learning

Approach

Curriculum learning

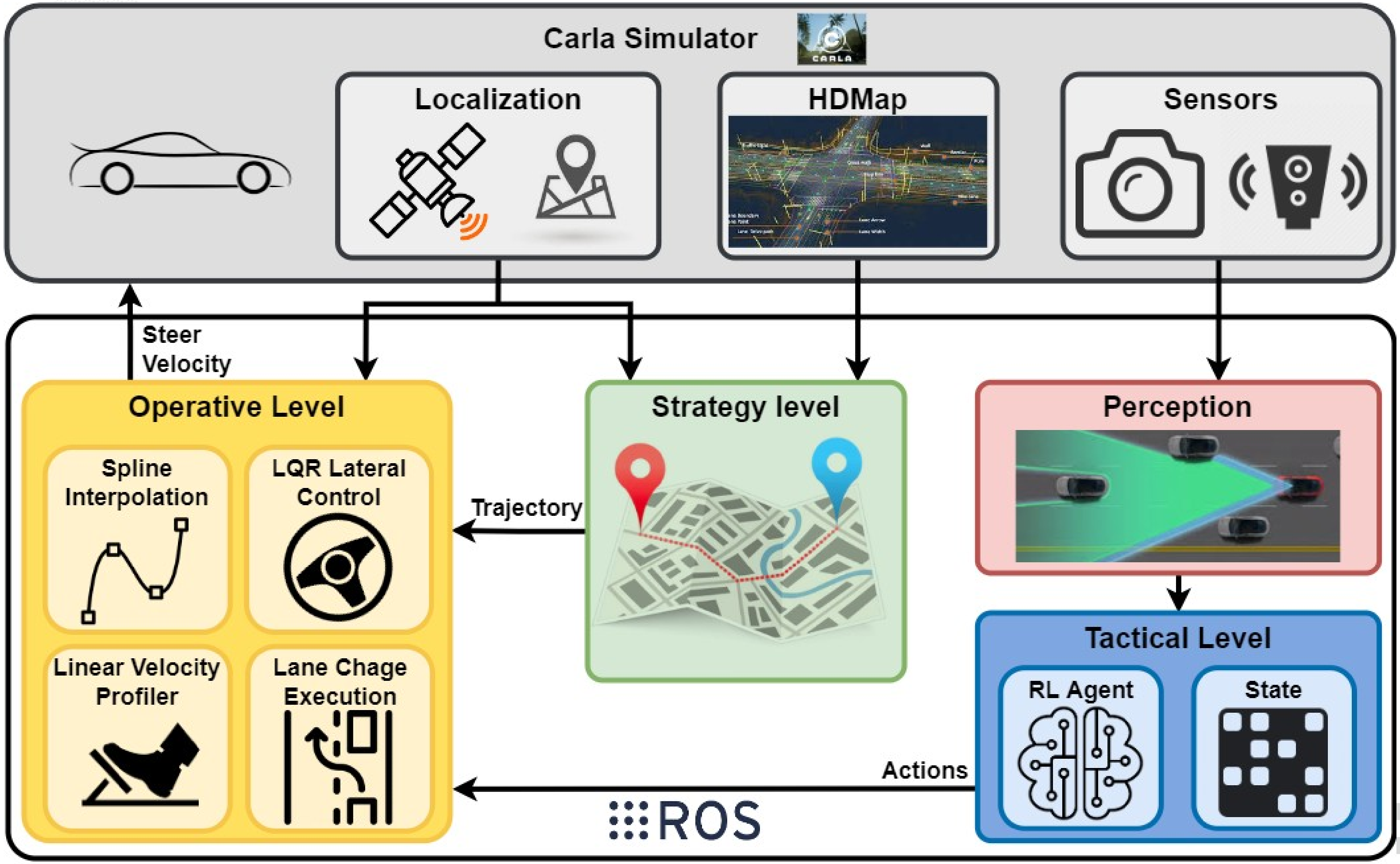
First stage → SUMO
Second stage → Carla

DRL

An execution layer is in charge of the
motion, while a decision-making layer executes the high-level actions

Diferent intersections

No rules
Traffic lights
Stop signal


Carla Simulator
Localization
HDMap
Sensors
Steer
Velocity
Operative Level
Spline Interpolation
LQR Lateral Control
Linear Velocity Profiler
Lane Chage Execution
Trajectory
Strategy level
Perception
Tactical Level
RL Agent
State
Actions
ROS

Approach

Strategy Level

HD map input → road and lanes graph → Dijkstra algorithm → route as way-points → ROS

Tactical Level

state vector to execute a high level action each time step

Operative Level

classic controller performs a smooth interpolation of the way-points using Linear Quadratic Regulator (LQR) techniques && velocity profile is generated

Policy-based method

$$L^{PG}(\theta) = \hat{\mathbb{E}}_t[log\pi_\theta(a_t|s_t)\hat{A}_t]$$

where Et is the expectation, πθ is the policy and Ât is an estimator of the advantage function at a time step t

PPO

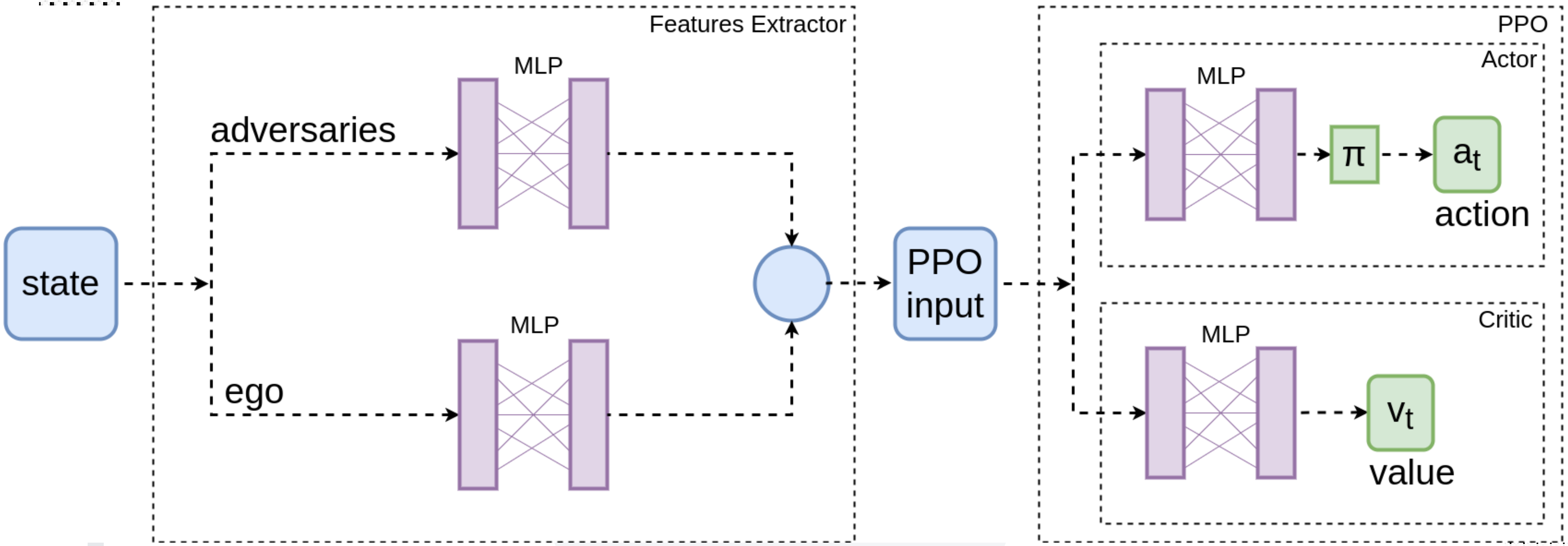
$$L_t^{CLIP+VF+S}(\theta) = \hat{\mathbb{E}}_t[L_t^{CLIP}(\theta) - c_1 L_t^{VF}(\theta) + c_2 S[\pi_\theta](s_t)]$$

where c1 , c2 are coefficients, S denotes an entropy bonus, and LVF
t (θ) is a squared-error loss

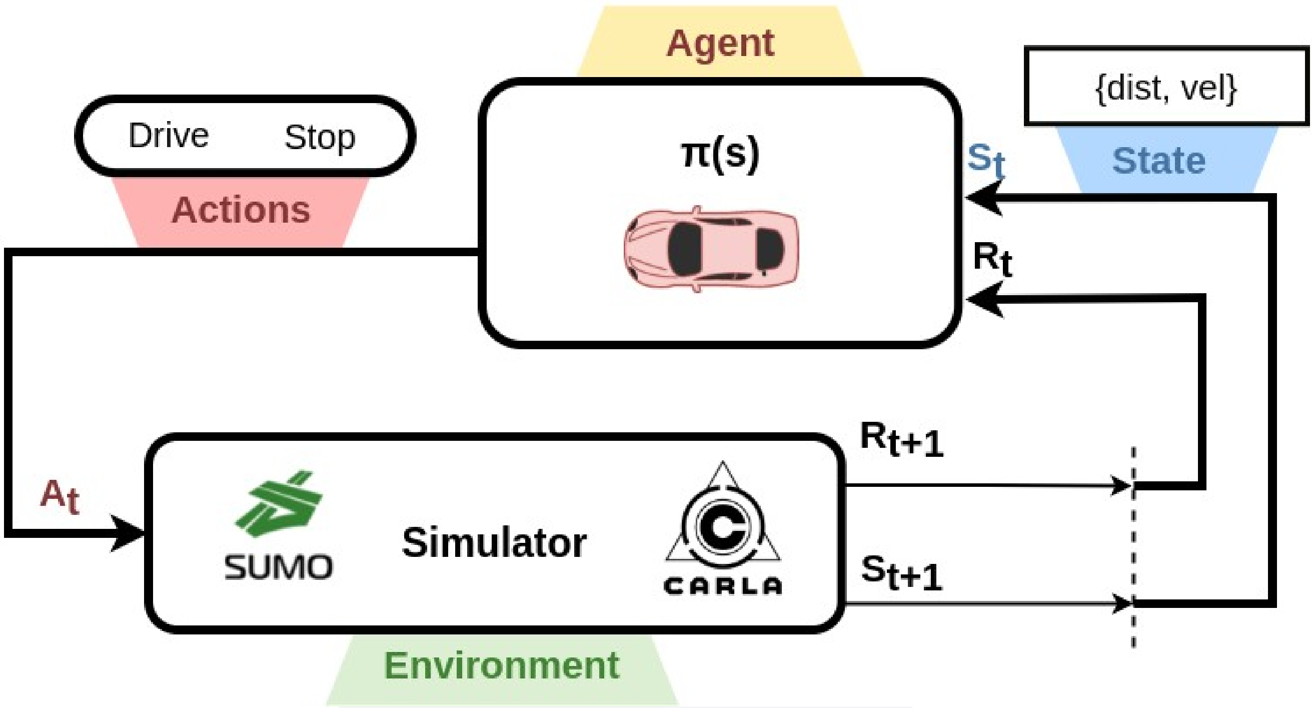
$$L_t^{CLIP}(\theta) = \hat{\mathbb{E}}_t[min(r_t(\theta)\hat{A}_t, clip(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)]$$

where epsilon is a hyperparameter and rt (θ) is the probability ratio:

$$r_t(\theta) = \pi_\theta(a_t|s_t)/\pi_{\theta_{old}}(a_t|s_t)$$

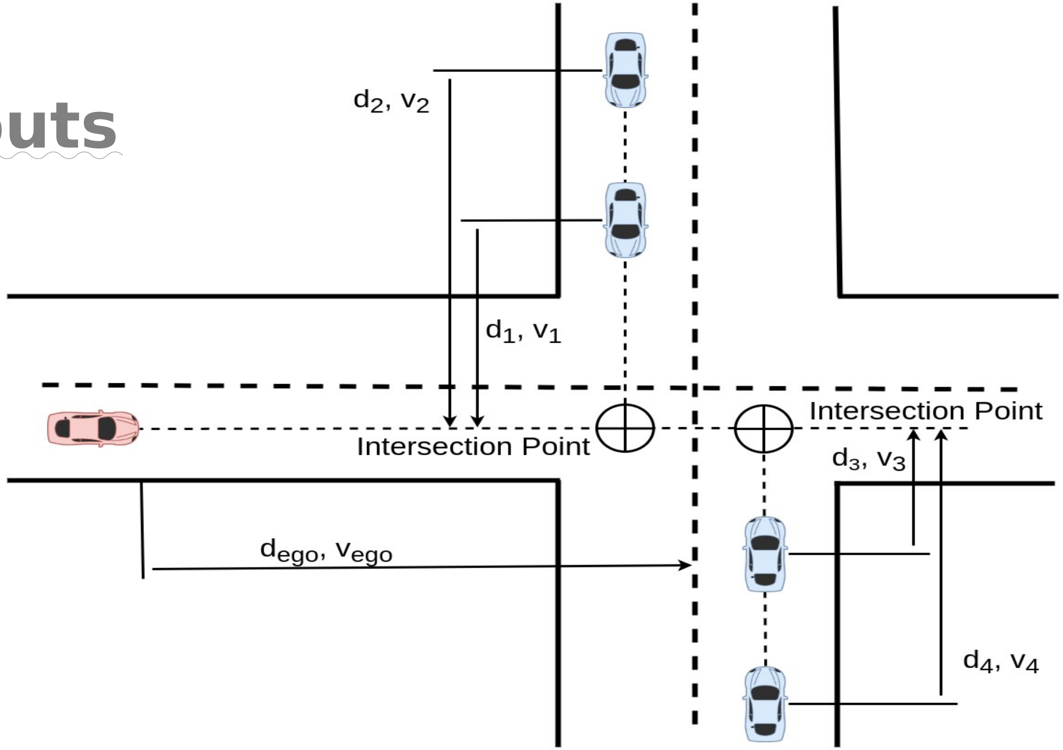


Features Extractor
MLP
adversaries
state
ego
MLP
PPO input
PPO
Actor
MLP
π
a_t
action
Critic
MLP
v_t
value



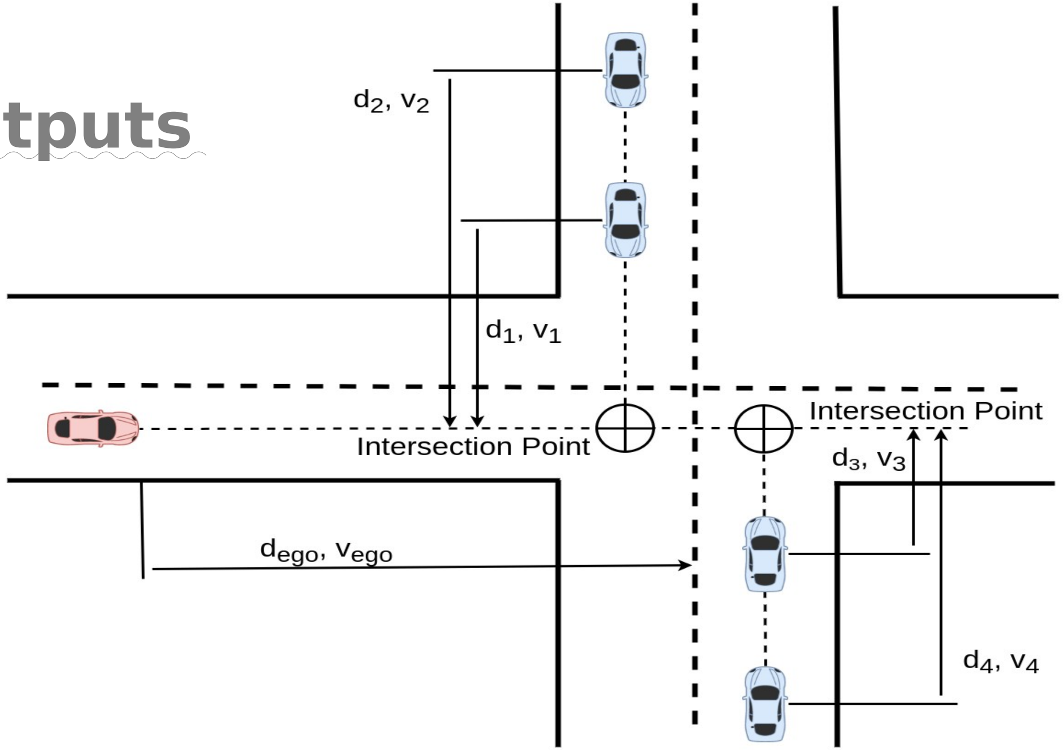
- The perception data are directly obtained from the simulator's ground truth.
- In real applications, these data could be obtained using the fusion of cameras and lidars
- continuous state vector and a set of discrete action
- define the intersection scenarios as an MDP

inputs



- distance to the intersection point and its longitudinal velocity: si = {di , vi } normalized between 0 and 1
- state vector as the collection of the individual states of the two closest vehicles for each lane and the ego vehicle

outputs



- a = {stop, drive}.
- Both actions set a desired velocity, stop sets 0 m/s and drive sets the nominal velocity of 5 m/s.

Rewards

Goal is to drive through the intersection as fast as possible avoiding adversarial vehicles

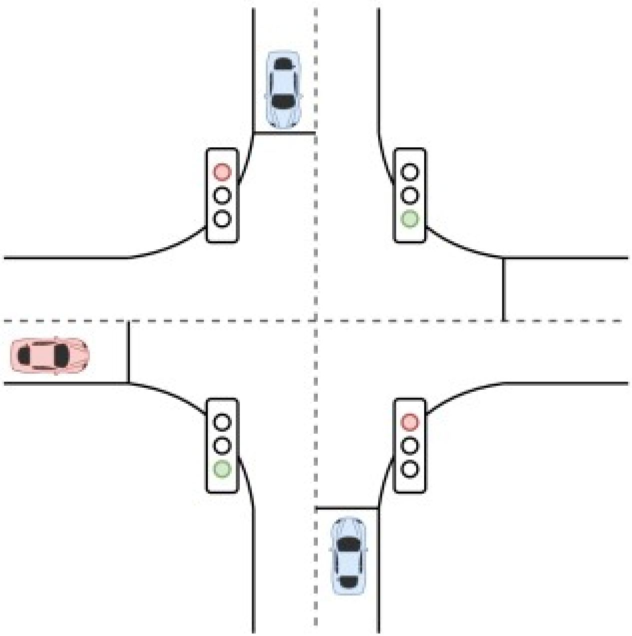
- Negative reward when collision. -2
- Positive reward when reaches the goal. 1
- Cumulative reward based on velocity. K * v
- Negative reward proportional to the duration. 0.2 / t

Intersection scenarios

We present the hypothesis that the RL agent (ego vehicle) is capable of driving only based on the position and speed of the adversarial vehicles

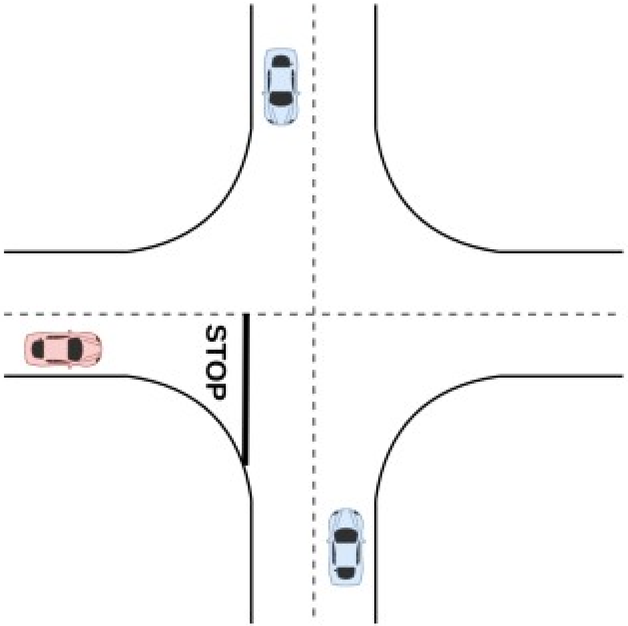
We need to define the scenarios in the two simulators slightly different to obtain similar intersection scenarios. In CARLA, the simulation is slower than in SUMO, so we use a shorter road and generate fewer adversaries, but the ego vehicle faces a similar situation when it approaches the intersection

Intersection scenarios



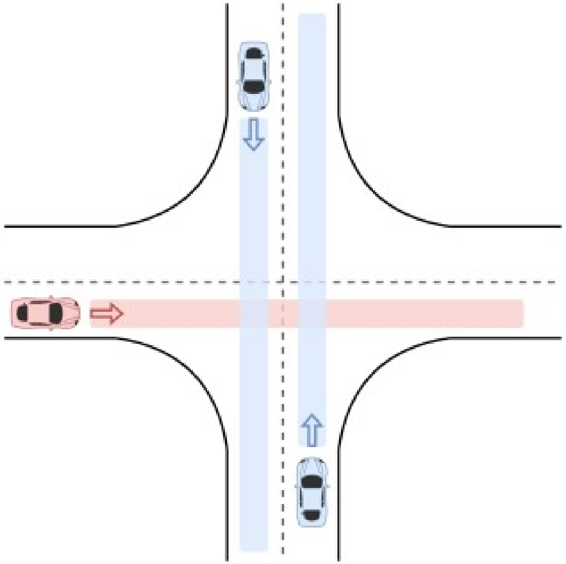
- collisions can be avoided even though the ego vehicle misses the traffic light

Intersection scenarios



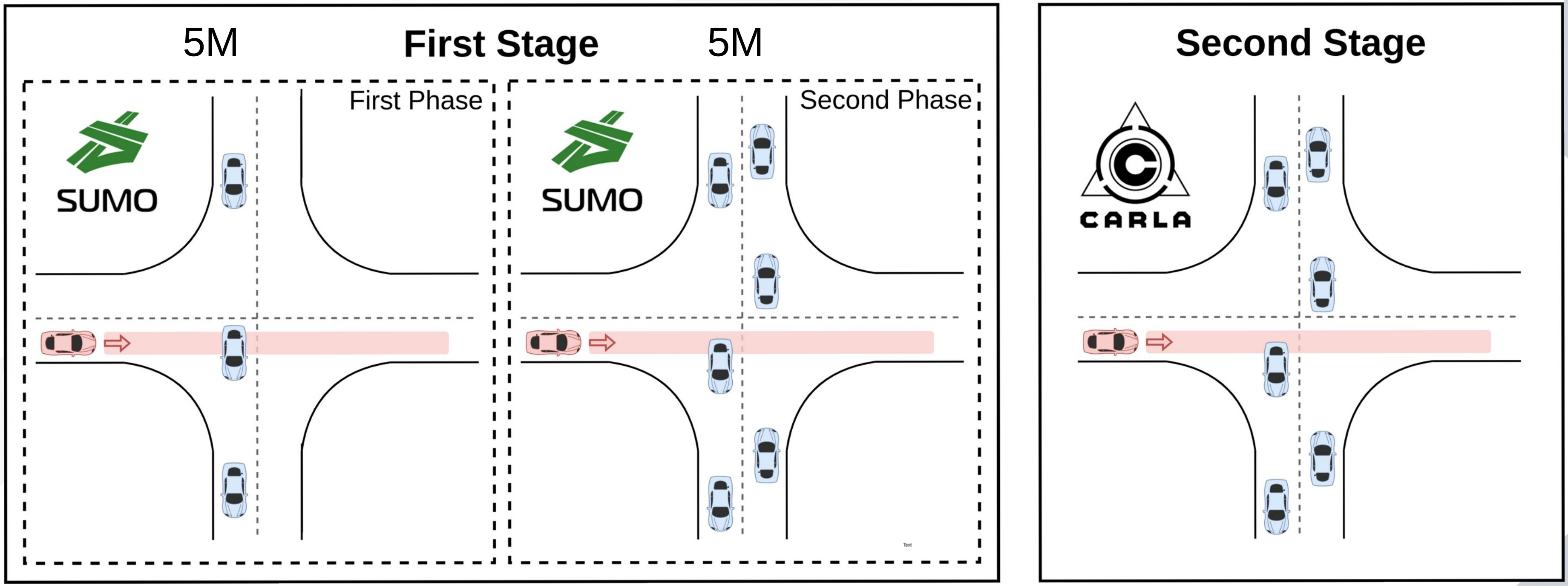
- These adversaries never stop, forcing the ego vehicle to stop and cross when there is a big enough gap

Intersection scenarios



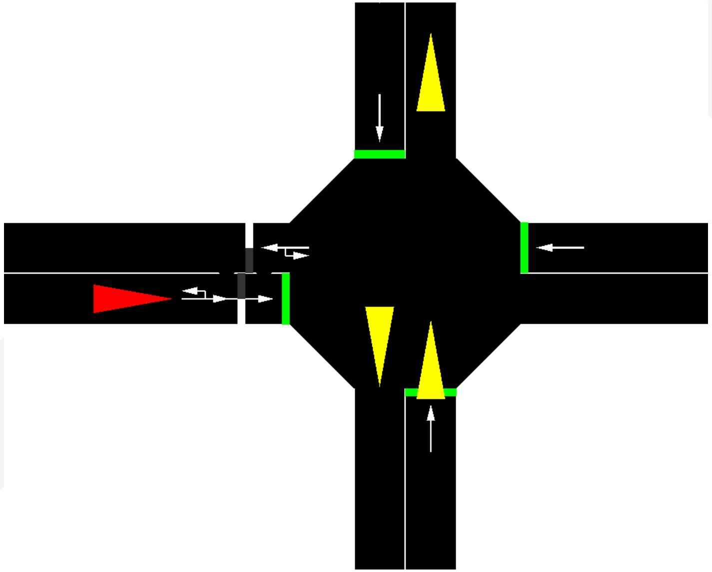
- most difficult scenario
- driver is obliged to yield to vehicles approaching from his right

Experiments



- SUMO does not have sensors neither vehicle dynamics
- SUMO is much faster than CARLA
- SUMO is developed using openAIGym

Experiments



In CARLA, three vehicles are spawned in each lane. These have different behaviors: they can yield, cross or stop randomly.

Evaluation metrics

- $success\ (\%) = end\ reached / n_e$
- $t_{avg} = \sum t_n \ / \ n_e$

Results

Table 2. Evaluation metrics results in SUMO simulator: comparison between algorithms trained with and without features structure.

	Traffic Light		Stop Signal		Uncontrolled		Combination	
	$success$ (%)	t_{avg} (s)	$success$ (%)	t_{avg} (s)	$success$ (%)	t_{avg} (s)	$success$ (%)	t_{avg} (s)
1-PPO	53	112	37	93	23	109	30	105
2-PPO	95	67	78	78	87	63	88	71
1-FEPPO	61	104	48	82	30	111	37	102
2-FEPPO	100	43	90	94	95	55	95	85

Results

Table 3. Evaluation metrics results in CARLA simulator: comparison between the model trained in SUMO (2-FEPPO) and the model trained in CARLA (SUMO + CARLA).

	Traffic Light		**Stop Signal**		**Uncontrolled**		**Combination**	
	$success$ (%)	t_{avg} (s)	$success$ (%)	t_{avg} (s)	$success$ (%)	t_{avg} (s)	$success$ (%)	t_{avg} (s)
SUMO	78	17	35	19	47	19	50	19
CARLA	83	17	70	19	75	16	78	17

Results

Table 4. Comparison of success rate between different approaches.

Architecture	Success Rate (%)
2-FEPPO	95
MPC Agent [24]	95.2
Level-k Agent [28]	93.8
Sc04 Left Turn [30]	90.3

Results

Table 5. Evaluation metrics results in SUMO simulator: ground truth vs. sensor data simulation.

	Traffic Light		**Stop Signal**		**Uncontrolled**		**Combination**	
	$success$ (%)	t_{avg} (s)	$success$ (%)	t_{avg} (s)	$success$ (%)	t_{avg} (s)	$success$ (%)	t_{avg} (s)
Ground Truth	100	43	90	94	95	55	95	85
Sensor Data	96	42	88	92	94	51	91	80

Discussion

- SUMO is faster

Table 6. Simulation time.

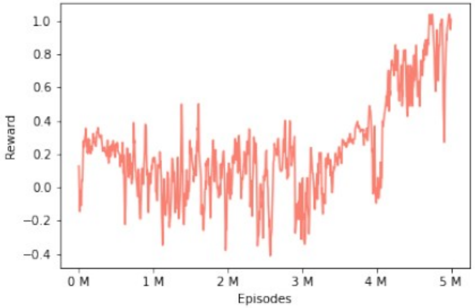
Simulator	No. of Episodes	Time (h)
SUMO	30 k	5
CARLA (estimated)	30 k	1650
SUMO + CARLA	30 k + 1 k	10.5

- Curriculum learning enabled PPO to converge
- The agent sometimes does not follow the rules (cross a red traffic light)

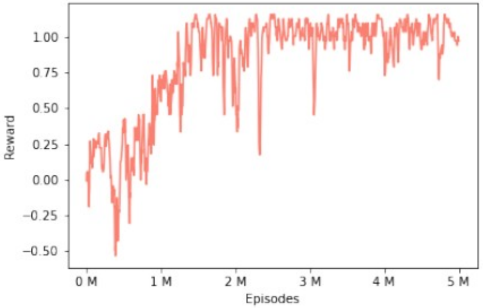
(a) (b) (c) (d)

Discussion

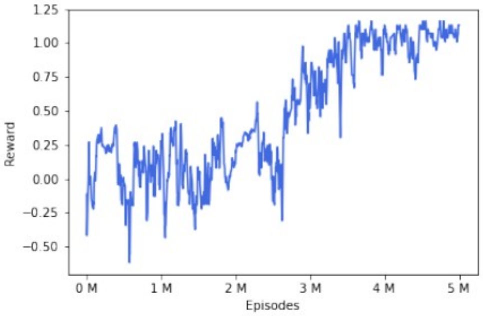
Features extractor improved the 2 stage training performance



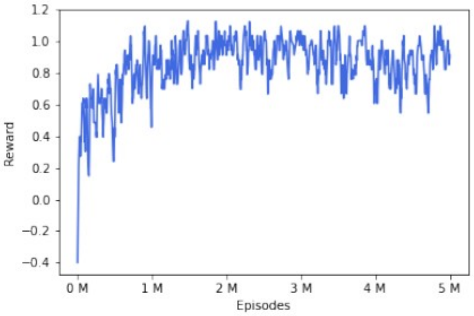
(a)



(b)



(c)



(d)

Future work

- Use sensor data instead of ground truth
- Use this system in a real environment