Think2Drive: Efficient Reinforcement Learning by Thinking with Latent World Model for Autonomous Driving (in CARLA-v2) Supplementary Material

Qifeng Li*, Xiaosong Jia*, Shaobo Wang, and Junchi Yan†

Shanghai Jiao Tong University, Shanghai 200240, China
{liqifeng, jiaxiaosong, shaobowang1009, yanjunchi}@sjtu.edu.cn
* Equal contributions †Correspondence (partly supported by NSFC 92370201)

A Discription of CARLA v2 Scenarios

CARLA v2 provides 39 corner scenarios which are common in real driving environments. These scenarios can generally be divided into two categories: regular road scenarios and junction scenarios. Here we give a detailed description of each scenario:

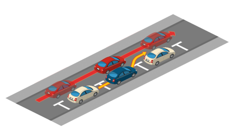
A.1 Regular road scenarios

These scenarios are related to regular road segments

1. ControlLoss

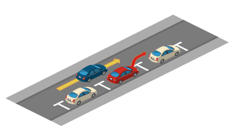
The ego vehicle loses control due to bad conditions on the road and it must recover, coming back to its original lane.

2. ParkingExit



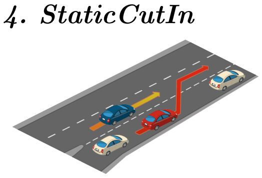
The ego vehicle must exit a parallel parking bay into a flow of traffic.

3. ParkingCutIn



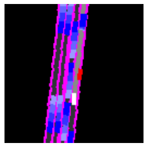
The ego vehicle must slow down or brake to allow a parked vehicle exiting a parallel parking bay to cut in front.

4. StaticCutIn



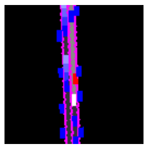
The ego vehicle must slow down or brake to allow a vehicle of the slow traffic flow in the adjacent lane to cut in front. Compared to *ParkingCutIn*, there are more cars in the adjacent lane and any one of them may cut in.

5. ParkedObstacle



The ego vehicle encounters a parked vehicle blocking part of the lane and must perform a lane change into traffic moving in the same direction to avoid it.

6. ParkedObstacleTwoWays



The '*TwoWays*' version of *ParkedObstacle*. The ego vehicle encounters a parked vehicle blocking the lane and must perform a lane change into traffic moving in the opposite direction to avoid it.

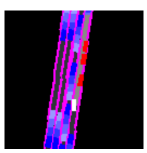
7. Construction

The ego vehicle encounters a construction site blocking and must perform a lane change into traffic moving in the same direction to avoid it. Compared to *ParkedObstacle*, the construction occupies more width of the lane. The ego vehicle has to completely deviate from its task route temporarily to bypass the construction zone.

8. ConstructionTwoWays

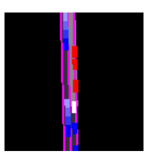
The '*TwoWays*' version of *Construction*.

9. Accident



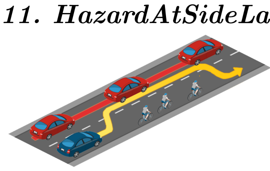
The ego vehicle encounters multiple accident cars blocking part of the lane and must perform a lane change into traffic moving in the same direction to avoid it. Compared to *ParkedObstacle* and *Construction*, these accident cars occupy more length along the lane. The ego vehicle has to completely deviate from its task route for a longer time to bypass the accident zone.

10. AccidentTwoWays



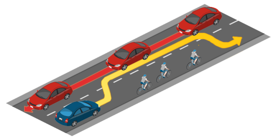
The '*TwoWays*' version of *Accident*. Compared to *ParkedObstacleTwoWays* and *ConstructionTwoWays*, there is a much shorter time window for the ego vehicle to bypass the route obstacles (i.g. accident cars).

11. HazardAtSideLane



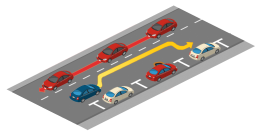
The ego vehicle encounters a slow-moving hazard blocking part of the lane. The ego vehicle must brake or maneuver next to a lane of traffic moving in the same direction to avoid it.

12. HazardAtSideLaneTwoWays



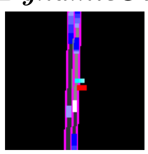
The ego vehicle encounters a slow-moving hazard blocking part of the lane. The ego vehicle must brake or maneuver to avoid it next to a lane of traffic moving in the opposite direction.

13. ***VehiclesDooropenTwoWays***



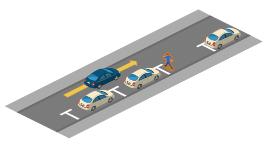
The ego vehicle encounters a parked vehicle opening a door into its lane and must maneuver to avoid it.

14. ***DynamicObjectCrossing***



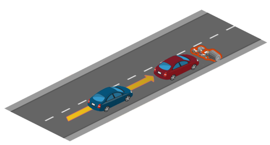
A walker or bicycle behind a static prop crosses the road suddenly when the ego vehicle is close to the prop. The ego vehicle must make a hard brake promptly.

15. ***ParkingCrossingPedestrian***



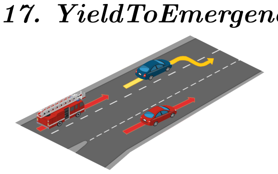
The ego vehicle encounters a pedestrian emerging from behind a parked vehicle and advancing into the lane. The ego vehicle must brake or maneuver to avoid it. Compared to *DynamicObjectCrossing*, the pedestrian is closer to the road and the ego vehicle has to act more timely.

16. ***HardBrake***



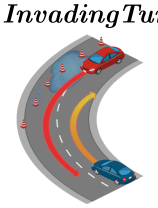
The leading vehicle decelerates suddenly and the ego vehicle must perform an emergency brake or an avoidance maneuver.

17. ***YieldToEmergencyVehicle***



The ego vehicle is approached by an emergency vehicle coming from behind. The ego vehicle must maneuver to allow the emergency vehicle to pass.

18. ***InvadingTurn***

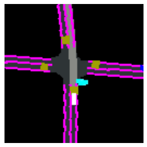


When the ego vehicle is about to turn right, a vehicle coming from the opposite lane invades the ego's lane, forcing the ego to move right to avoid a possible collision.

A.2 Junction scenarios

These scenarios are related to junctions.

1. ***PedestrainCrossing***



While the ego vehicle is entering a junction, a group of natural pedestrians suddenly cross the road and ignore the traffic light. The ego vehicle must stop and wait for all pedestrians to pass even though there is a green traffic light or a clear junction.

2. VehicleTurningRoutePedestrian

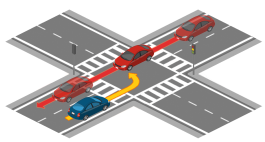
While performing a maneuver, the ego vehicle encounters a pedestrian crossing the road and must perform an emergency brake or an avoidance maneuver.

3. VehicleTurningRoute While performing a maneuver, the ego vehicle encounters a bicycle crossing the road and must perform an emergency brake or an avoidance maneuver. Compared to *VehicleTurningRoutePedestrian*, the bicycle moves faster and the ego has to brake earlier.

4. BlockedIntersection

While performing a maneuver, the ego vehicle encounters a stopped vehicle on the road and must perform an emergency brake or an avoidance maneuver.

5. SignalizedJunctionLeftTurn

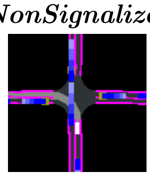


The ego vehicle is performing an unprotected left turn at an intersection, yielding to oncoming traffic.

6. SignalizedJunctionLeftTurnEnterFlow

The ego vehicle is performing an unprotected left turn at an intersection, merging into opposite traffic.

7. NonSignalizedJunctionLeftTurn

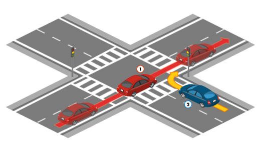


Non-signalized version of *SignalizedJunctionLeftTurn*. The ego has to negotiate with the opposite vehicles without traffic lights.

8. NonSignalizedJunctionLeftTurnEnterFlow

Non-signalized version of *SignalizedJunctionLeftTurnEnterFlow*.

9. SignalizedJunctionRightTurn

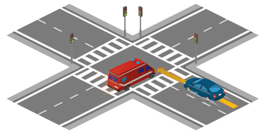


The ego vehicle is turning right at an intersection and has to safely merge into the traffic flow coming from its left.

10. NonSignalizedJunctionRightTurn

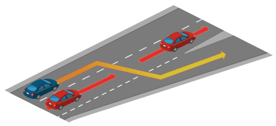
Non-signalized version of *SignalizedJunctionRightTurn*. The ego has to negotiate with the traffic flow without traffic lights.

11. EnterActorFlows



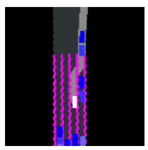
A flow of cars runs a red light in front of the ego when it enters the junction, forcing it to react (interrupting the flow or merging into the flow). These vehicles are 'special' ones such as police cars, ambulances, or firetrucks.

12. HighwayExit



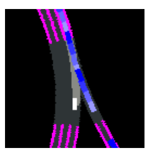
The ego vehicle must cross a lane of moving traffic to exit the highway at an off-ramp.

13. MergerIntoSlowTraffic



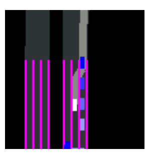
The ego vehicle must merge into a slow traffic flow on the off-ramp when exiting the highway.

14. MergerIntoSlowTrafficV2



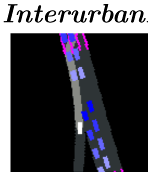
The ego vehicle must merge into a slow traffic flow coming from the on-ramp when driving on highway roads.

15. InterurbanActorFlow



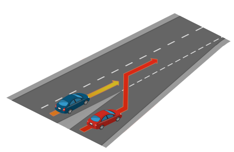
The ego vehicle leaves the interurban road by turning left, crossing a fast traffic flow.

16. InterurbanAdvancedActorFlow



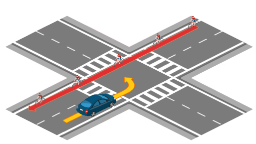
The ego vehicle incorporates into the interurban road by turning left, first crossing a fast traffic flow, and then merging into another one.

17. HighwayCutIn



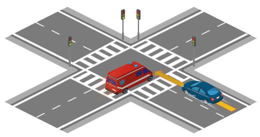
The ego vehicle encounters a vehicle merging into its lane from a highway on-ramp. The ego vehicle must decelerate, brake, or change lanes to avoid a collision.

18. CrossingBicycleFlow



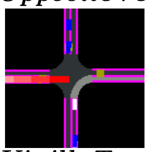
The ego vehicle needs to perform a turn at an intersection yielding to bicycles crossing from either the left.

19. OppositeVehicleRunningRedLight



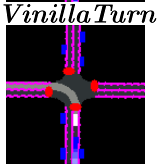
The ego vehicle is going straight at an intersection but a crossing vehicle runs a red light, forcing the ego vehicle to avoid the collision.

20. OppositeVehicleTakingPriority



Non-signalized version of *OppositeVehicleTakingPriority*.

21. VinillaTurn



A basic scenario for the ego vehicle to learn the basic traffic rules, e.g. stop signs and traffic lights.

B Input & Output Representation

Table B.1: Composition of the BEV representation. Each static object occupies 1 dimension out of the $C = 34$, while each dynamic object occupies 4 dimensions out of the $C = 34$.

$C = 34$													
Static($C_s = 1$)						Dynamic($C_d = 4$)							
road	route	ego	lane	yellow line	white line	vehicle	walker	emergency car	obstacle	green traffic light	yellow&red traffic light	stop sign

Table B.2: Dimension of input and output representation. The T temporal masks are from historical time-steps: $[-16, -11, -6, -1]$.

H	W	C	T	M
128	128	34	4	30

Table B.3: The discretized actions. The continuous action space is decomposed into 30 discrete actions, each for specific values of throttle, steer, and brake. Each action is rational and legitimate.

Throttle	Brake	Steer	Throttle	Brake	Steer	Throttle	Brake	Steer
0	1	0	0.3	0	-0.7	0.3	0	0.7
0.7	0	-0.5	0.3	0	-0.5	0	0	-1
0.7	0	-0.3	0.3	0	-0.3	0	0	-0.6
0.7	0	-0.2	0.3	0	-0.2	0	0	-0.3
0.7	0	-0.1	0.3	0	-0.1	0	0	-0.1
0.7	0	0	0.3	0	0	0	0	0
0.7	0	0.1	0.3	0	0.1	0	0	0.1
0.7	0	0.2	0.3	0	0.2	0	0	0.3
0.7	0	0.3	0.3	0	0.3	0	0	0.6
0.7	0	0.5	0.3	0	0.5	0	0	1

C Simulator Execution Mode

We boost CARLA running efficiency via asynchronous reloading and parallel execution which could avoid the waiting time caused by the long preparation time of starting a new route, as shown in Fig. C.1. For each 100K steps of execution, it can reduce about 1 day time cost.

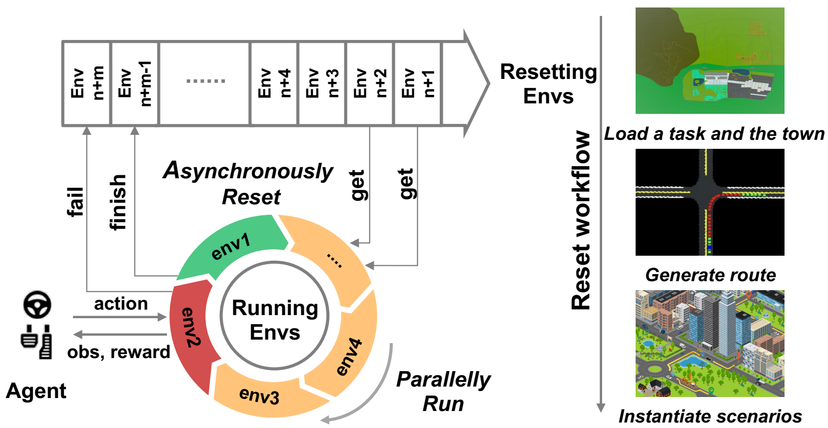


Fig. C.1: Procedure of the Wrapped RL Environment. The right shows the workflow of each reset. In large maps like town12 and town13, the entire reset workflow takes about 1 minute.

D Hyper-parameters

Table D.1: Hyper-parameters of the neural network. The CNN encoder maps the 128×128 BEV to the 4×4 feature map by 4×4 convolutional kernel with stride 2. The flattened feature maps, concatenated with the state features output by the MLP encoder, are then input to the RSSM which consists of GRU cells and a few dense layers. The structure of the decoder inverts that of the encoder and outputs the reconstructed mask and state vector. Please refer to DreamerV3 [16] for details about network.

GRU recurrent units	CNN multiplier	Dense hidden units	MLP layers	Parameters
512	96	512	5	104M