Supplementary Material: Conditional Affordance Learning for Driving in Urban Environments

Axel Sauer[1] **Nikolay Savinov**[1] **Andreas Geiger**[1,2]

[1]Computer Vision and Geometry Group, ETH Zürich
[2]Autonomous Vision Group, MPI for Intelligent Systems and University of Tübingen

Abstract: This **supplementary document** provides further implementation details of our CAL agent in Section 1, a detailed description of our ground truth acquisition process in Section 2 and additional experiments in Section 3.
The **supplementary video** shows several navigation examples and visualizes the attention of our agent for different affordance indicators over time.

1 Implementation Details

In this section, we list the value ranges and results of our hyperparameter search. We also provide additional details about our longitudinal control algorithm and the PID tuning procedure.

1.1 Hyperparameter Search

Table 1 shows each hyperparameter and its range of values for the random search described in the main paper. We initialize the network with randomly sampled parameters from the respective ranges. Table 2 shows the parameters of the best performing task blocks after optimization.

Table 1: **Hyperparameter Search.** Value ranges for each hyperparameter.

Hyperparameter Acronym	Layer type $type$	Number of nodes $nodes$	Dropout amount p	Sequence length seq	Dilation value dil
Range of values	*Dense, GRU, LSTM,* *temporal convolution*	$[10, 200]$	$[0.25, 0.75]$	$[1, 20]$	$[1, 3]$

Table 2: **Best task block parameters.** We optimized the layer $type$, the number of $nodes$, the dropout ratio p, the sequence length seq, and the dilation value dil.

Task	$type$	$nodes$	p	seq	dil
Red light	GRU	185	0.27	14	2
Hazard stop	Temp. convolution	160	0.68	6	1
Speed sign	Dense	160	0.55	1	1
Vehicle distance	GRU	160	0.38	11	1
Relative angle	Temp. convolution	100	0.44	10	1
Center distance	Temp. convolution	100	0.44	10	1

2nd Conference on Robot Learning (CoRL 2018), Zürich, Switzerland.

1.2 Controller

Longitudinal Control. The states of the longitudinal controller are illustrated in Figure 1.

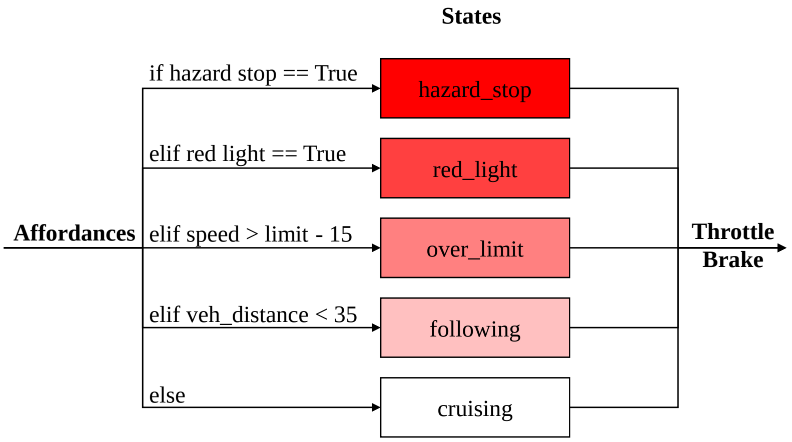

Figure 1: **States of Longitudinal Controller.** The states are ordered in descending importance from top-to-bottom as indicated by the color intensity. All state are mutually exclusive.

The throttle and brake values for the states *over_limit*, *red_light*, and *hazard_stop* are as follows:

- **over_limit**: This state is activated, if the agent is driving more than 15 km/h faster than the speed limit v^*. This situation typically occurs when entering a low-speed zone from a high-speed zone. To decelerate quickly, we set the throttle value to zero and calculate the brake value depending on the current speed $v(t)$:

$$brake = 0.3 \cdot \frac{v(t)}{v^*(t)} \tag{1}$$

  As an example, driving with full speed (90 km/h) into a 30 km/h zone yields $brake = 0.9$, hence applying the brakes almost fully.

- **red_light**: If the prediction probability P for the class *red_light*, i.e., the actual softmax output, is higher than a threshold P_{rl}, the controller switches to the state *red_light*. The throttle is set to zero and the brake is applied with:

$$brake = 0.2 \cdot \frac{v(t)}{30} \tag{2}$$

  We empirically found that a threshold of $P_{rl} = 0.9$ reduces false positives while still being able to reliable stop in front of red lights. Note that we use a smaller multiplier (0.2) compared to the $over\_limit$ state as red lights typically occur in 30 km/h zones. We consider the current speed $v(t)$ to gradually slow down the car in front of red lights.

- **hazard_stop**: This state is activated when an obstacle in front of the agent is detected, i.e., when $P(hazard\_stop) > P_{hs}$ where we empirically determined the threshold P_{hs} as $P_{hs} = 0.7$. Note the threshold is lower than that for the red_light state, since preventing crashes with road hazards is more critical for successful goal-directed navigation. When a $hazard\_stop$ has been detected, the throttle is set to zero and the brake is set to one.

PID Controller Tuning. The PID controllers used in the $cruising$ and $following$ state follow the standard PID control scheme [1]. The overall control function is given as follows

$$u(t) = K_p e(t) + K_i \int_0^t e(t')dt' + K_d \frac{de(t)}{dt}, \tag{3}$$

where K_p, K_i and K_d are the proportional, integral and derivative coefficients. To be able to tune the coefficients of the two PID controllers, we implemented a visualization tool of speed, distance to the centerline, and other important signals. With this direct visual feedback, it is possible to use

Table 3: **Observation Areas.** The observation areas are rectangular boxes. This table lists the (x, y) coordinates of the vertices v of each observation area (provided in $\boldsymbol{c_{local}}$ in meters.)

Area	Detection of	v_1	v_2	v_3	v_4
$A1$	*red_light* *speed_sign*	(7.4, -0.8)	(7.4, -5.8)	(14.0, -0.8)	(14.0, -5.8)
$A2$	*hazard_stop*	(0.0, 2.0)	(0.0, -2.0)	(8.2,2.0)	(8.2,-2.0)
$A3$	*distance_to_vehicle*	(0.0, 1.6)	(0.0, -1.6)	(50.0, 1.6)	(50.0, -1.6)

standard PID tuning methods. In this work, we leverage the method by Ziegler-Nichols [2]. First, all coefficients are set to zero. Then, the proportional gain K_p is increased until the output of the loop starts to oscillate. Given this "ultimate gain" K_u and the oscillation period T_u, we set the coefficients to

$$K_p = 0.6K_u, \tag{4}$$

$$K_i = T_u/2, \tag{5}$$

$$K_d = T_u/8. \tag{6}$$

Using these values as a starting point, we empirically fine-tune the coefficients for optimal performance, with the goal of enabling fast but smooth reactions to disturbances.

2 Ground Truth Acquisition

The API of CARLA supplies measurements about the agent (speed, acceleration, location, orientation) and about other objects in the scene (cars, pedestrians, traffic lights, and speed limit signs). These measurements include the current status of the traffic light (green, orange, red) and the type of the speed sign (30, 60, 90). Location and orientation are defined in a world coordinate system $\boldsymbol{c_{global}} = (x_g, y_g, z_g)^\intercal$. As these measurements do not directly express the affordances we want to learn, we implemented a procedure to convert them into the desired ground truth.

2.1 Observation Area

We define a local coordinate system $\boldsymbol{c_{local}} = (x_l, y_l, z_l)^\intercal$ at the center of the front axle of the car with the x-axis corresponding to the car's lateral axis and the z-axis corresponding to the up vector. The agent's orientation ψ and the agent's position (x_{ego}, y_{ego}) is supplied in $\boldsymbol{c_{global}}$. Using this information, we convert the position of all other objects to $\boldsymbol{c_{local}}$. Next, we define the *observation areas* as rectangles in the x-y plane of $\boldsymbol{c_{local}}$, see Figure 3 of the main paper for an illustration. If an object falls into an observation area it is considered "detected".

The length, width, position, and orientation of the observation areas are chosen based on the respective affordances. Thus, the observation area for red_light and for $speed\_sign$ is located on the right side of the agent as their respective signals are located on the right sidewalk. The observation area for $hazard\_stop$ is directly in front of the car and very short, in order to only initiate a hazard stop if an accident could not be avoided otherwise. The observation area for $vehicle\_distance$ is in front of the car and has a length of 50 m. If another vehicle is located within this area, the distance of the closest vehicle to the agent is measured. If there is no car in front, the default value for $vehicle\_distance$ (50 m) is used. Table 3 lists the coordinates for each observation area.

2.2 Directional Input

CARLA provides a high-level topological planner based on the A* algorithm. It takes the agents position and the coordinates of the destination and calculates a list of commands. This "plan" advises the agent to turn left, right or to keep straight at intersections.

Table 4: **Comparison of Temporal and Non-temporal Task Blocks**. The last column shows the relative change in performance. The higher the $\overline{IoU}$ and the lower the MAE the better.

		Best performing task blocks		
Task	Metric	non-temporal	temporal	relative Change
Hazard stop	$\overline{IoU}$	84.96 %	87.41 %	+ 2.88 %
Speed sign	$\overline{IoU}$	91.95 %	92.71 %	+ 0.83 %
Red light	$\overline{IoU}$	92.41 %	93.95 %	+ 1.67 %
Relative angle	MAE	0.00797	0.00207	- 74.03 %
Center distance	MAE	0.09642	0.08465	- 12.21 %
Vehicle distance	MAE	0.04497	0.03289	- 26.86 %

3 Additional Experiments

In this section, we provide additional experiments. First, we compare temporal to non-temporal task blocks to assess the influence of the additional temporal information provided by video data. Second, we provide a qualitative evaluation of our agent's driving behavior.

3.1 Comparison of Temporal and Non-temporal Task Blocks

Table 4 shows the best performing task blocks for each task. By using a temporal task block, all the classification and regression results improve. This demonstrates that each task profits from the additional temporal information.

The biggest relative improvement can be seen for the *relative_angle* task block. The error of the temporal task block is almost four times lower than the non-temporal task block. This suggests that this task profits more from the temporal context than other tasks. The smallest improvement is achieved for the *speed_sign* task. To keep computation time low during training and inference, we therefore use the non-temporal task block for this task in our final model.

In addition, we empirically observed that there is no dominating temporal layer in terms of performance. LSTMs, GRUs and temporal convolution layers perform very similar. We employ temporal convolution layers in our final implementation.

3.2 Driving Behaviour

The experiments in the main text examined whether the goal was reached and if there were any rule violations. This section focuses on qualitative driving experience, i.e., how the driving would be perceived by a passenger. The evaluation is done for the task "navigation without dynamic objects" to evaluate the general driving behavior without distorting the results by the challenges of stopping for cars or pedestrians. We use the following metrics for evaluation:

- **Centerline distance:** Staying in the middle of the road is the main task of every lane keeping system. We evaluate the ability of the algorithm to minimize the deviation from the centerline. The reported result is the median over all episodes. The median is more descriptive for the qualitative driving experience than the mean value since failed episodes during which an agent drifts off the road produce large outliers.
- **Jerk** is the rate of change of acceleration. The jerk can be felt during driving in the form of a jolt or sudden shock. It is commonly used to quantify ride comfort [3]. A smoother ride results in lower fuel consumption increased passenger comfort and more safety. A way to quantify jerk is to compare the root mean square (RMS) jerk [4]. The analysis further distinguishes between longitudinal and lateral jerk with lateral jerk separately evaluated for straight roads and in turns.

Table 5 reports our results. In contrast to the previous evaluations, the results are not reported depending on the weathers conditions or the environments. The results are very similar under all conditions for all agents. The CAL agent is able to achieve the best performance on all four metrics.

Table 5: **Driving Behaviour.** Qualitative evaluation of the general driving performance, the lower a metric the better.

Metrics	Unit	CIL	RL	CAL
Distance to centerline	$[m]$	0.390	0.755	**0.334**
Longitudinal jerk	$[m/s^3]$	0.449	1.368	**0.333**
Lateral jerk driving straight	$[m/s^3]$	0.084	0.336	**0.052**
Lateral jerk driving turns	$[m/s^3]$	0.242	0.548	**0.065**

Distance to Centerline

All agents perform on a similar level and are able to keep the distance to the centerline low. The exception is the RL approach. When driving on a straight road, the RL agent regularly starts swaying from side to side over the entire lane, resulting in the high value.

Longitudinal Jerk

The CAL agent performs best, followed by CIL and RL. The control parameters of the CAL agent are freely adjustable which allows to accelerate and decelerate smoothly as well as driving at a constant pace. The RL agent is only able to set the throttle value to either 0 or 1. This results in a sudden jerk every time the agent utilizes the throttle.

Lateral Jerk while Driving Straight

On straight roads, both the CAL and the CIL agent perform similarly. When driving straight, the RL agent often outputs a steering value of 0. This leads to the agent drifting off the road. When correcting against the drift, the RL agent steers abruptly, resulting in large jerk values.

Lateral Jerk while Turning

The CAL agent performs exceptionally well. There are several reasons for this. First, the agent is slowing down when approaching a turn. Second, our agent turns smoothly without abrupt changes in steering. Third, the jerk peaks are generally lower than for the other approaches. Despite this good performance, the transition from turns to straight roads leaves room for improvement. Changes in the directional switch result in sudden jumps in the prediction of the relative angle in some cases, resulting in slight short-timed jerk. The CIL agent is not as good as the CAL agent, but it is generally able to drive through turns smoothly. RL, in contrast, conducts strong and abrupt steering movements, resulting in a higher jerk value.

References

[1] K. H. Ang, G. Chong, and Y. Li. PID control system analysis, design, and technology. *IEEE transactions on control systems technology*, 13(4):559–576, 2005.

[2] J. G. Ziegler and N. B. Nichols. Optimum settings for automatic controllers. *trans. ASME*, 64 (11), 1942.

[3] Q. Huang and H. Wang. Fundamental study of jerk: evaluation of shift quality and ride comfort. Technical report, SAE Technical Paper, 2004.

[4] D. Hrovat and M. Hubbard. Optimum vehicle suspensions minimizing rms rattlespace, sprung-mass acceleration and jerk. *Journal of Dynamic Systems, Measurement, and Control*, 103(3): 228–236, 1981.