Training CARLA with reduced data

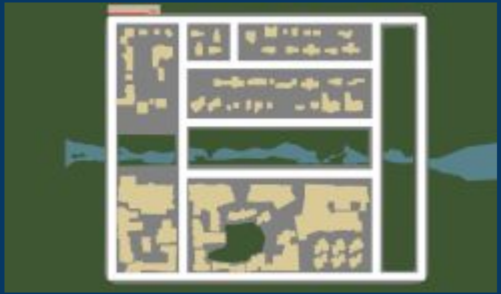
- *Indranil Sarkar, Niladri Shekhar Dutt*

CARLA : Introduction

CARLA is an open-source simulator for autonomous driving research simulating urban driving scenario.
Version : In order to get the dataset , we've used CARLA_0.8.2 stable version. There are 2 towns. Town 1 has a total of 2.9 km of drivable roads, and Town 2 has 1.4 km of drivable roads as shown in images as white strips.

Measurements:-

- ❖ Player agent :- [Steer, Throttle, Brake, Position, Direction]
- ❖ Non-player agent :- [Other vehicle, Pedestrian, Traffic light, Speed sign]
- ❖ Sensor :- [Camera, Lidar]



Town 1

Town 2

Literature Survey

Autonomous navigation can be described as a mapping function from sensory input to control output. To implement this function, four major approaches have been proposed:

1. Modular Pipelines[1]
2. Imitation Learning[1]
3. Reinforcement Learning[1]
4. Direct Perception[2]

[1]CARLA : Dosovitskiy, Alexey, et al. "CARLA: An open urban driving simulator." arXiv preprint arXiv:1711.03938 (2017).

[2]CAL : Sauer, Axel, Nikolay Savinov, and Andreas Geiger. "Conditional affordance learning for driving in urban environments." arXiv preprint arXiv:1806.06498 (2018).

Conditional Affordance learning (CAL)

The table * defines the performance of different methods in 4 different conditions and in 4 different tasks. The methods used for training are ***Modular Pipeline(MP)***, ***Conditional Imitation Learning(CIL)***, ***Reinforcement Learning(RL)*** and ***Conditional Affordance Learning(CAL).***

- ❖ Navigation : No non-player agents are present while running the model.
- ❖ Navigation dynamic : Non-player agents are present in the environment.

	Training conditions				New weather				New town				New town and new weather			
Task	MP	CIL	RL	CAL	MP	CIL	RL	CAL	MP	CIL	RL	CAL	MP	CIL	RL	CAL
Straight	98	95	89	**100**	**100**	98	86	**100**	92	**97**	74	93	50	80	68	**94**
One turn	82	89	34	**97**	95	90	16	**96**	61	59	12	**82**	50	48	20	**72**
Navigation	80	86	14	**92**	**94**	84	2	90	24	40	3	**70**	47	44	6	**68**
Nav. dynamic	77	**83**	7	**83**	**89**	82	2	82	24	38	2	**64**	44	42	4	**64**

As CAL performed better than other three learning, so we used CAL for our training.

*CAL : Sauer, Axel, Nikolay Savinov, and Andreas Geiger. "Conditional affordance learning for driving in urban environments." arXiv preprint arXiv:1806.06498 (2018).

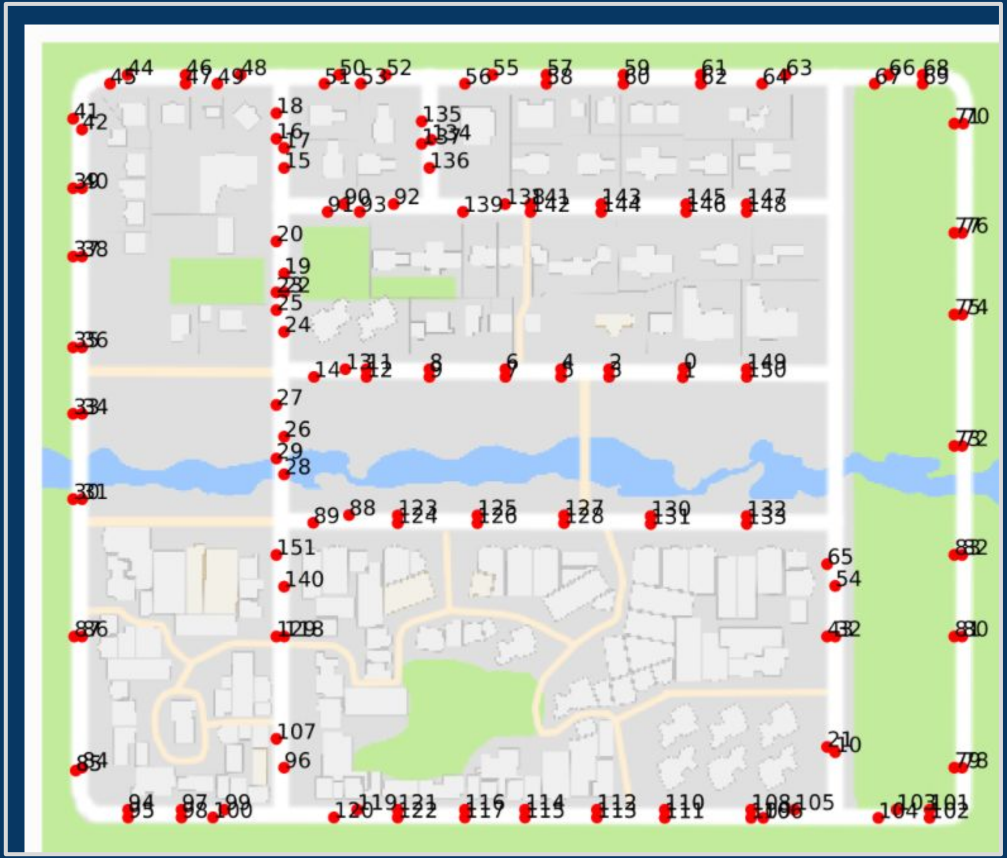
Introduction to terms

- ***Affordance*** :- The term 'Affordance' refers to our implicit understanding of how to interact with an object. There are six affordances, i.e. Hazard stop, Speed sign, Traffic light, Relative angle, Center distance, Vehicle distance.
- ***Features*** :- Features are useful information present in the images taken by camera(s) attached to agent vehicle.
- ***VGG11_bn*** :- Pre-trained batch normalized version of Pytorch model VGG11 has been used as the feature extractor for training and driving purpose.
- ***Taskblock*** :- These correspond to the affordances. They are shallow networks and consist of a task-specific layer, followed by a batch normalization layer and a dropout layer. It is of two types-
    - Conditional - Relative Angle, Center Distance (depends on direction)
    - Unconditional - Hazard Stop, Speed Sign, Traffic Light, Vehicle Distance

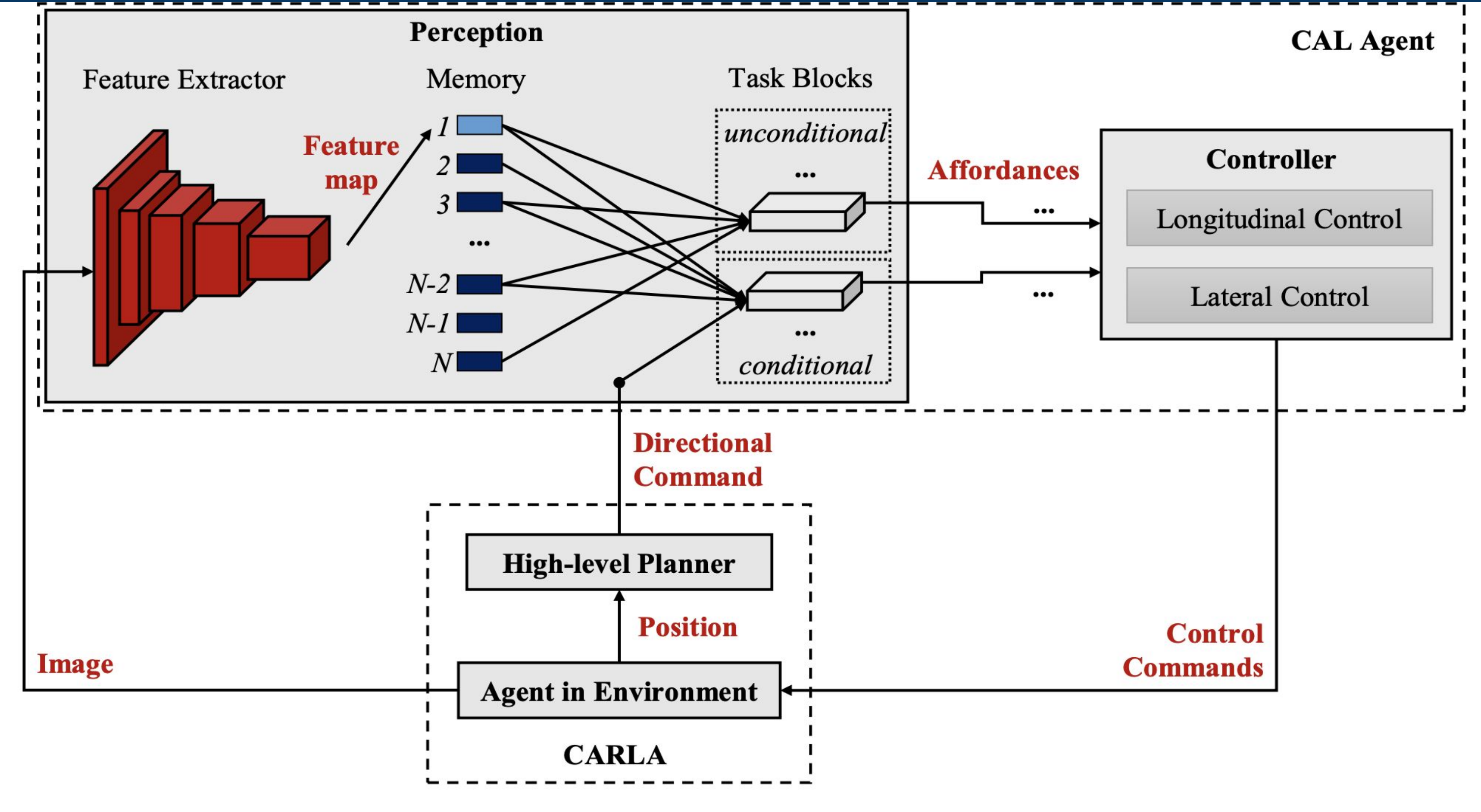
Poses

Poses:- A pair of values, indicates the starting and ending point in a Town.

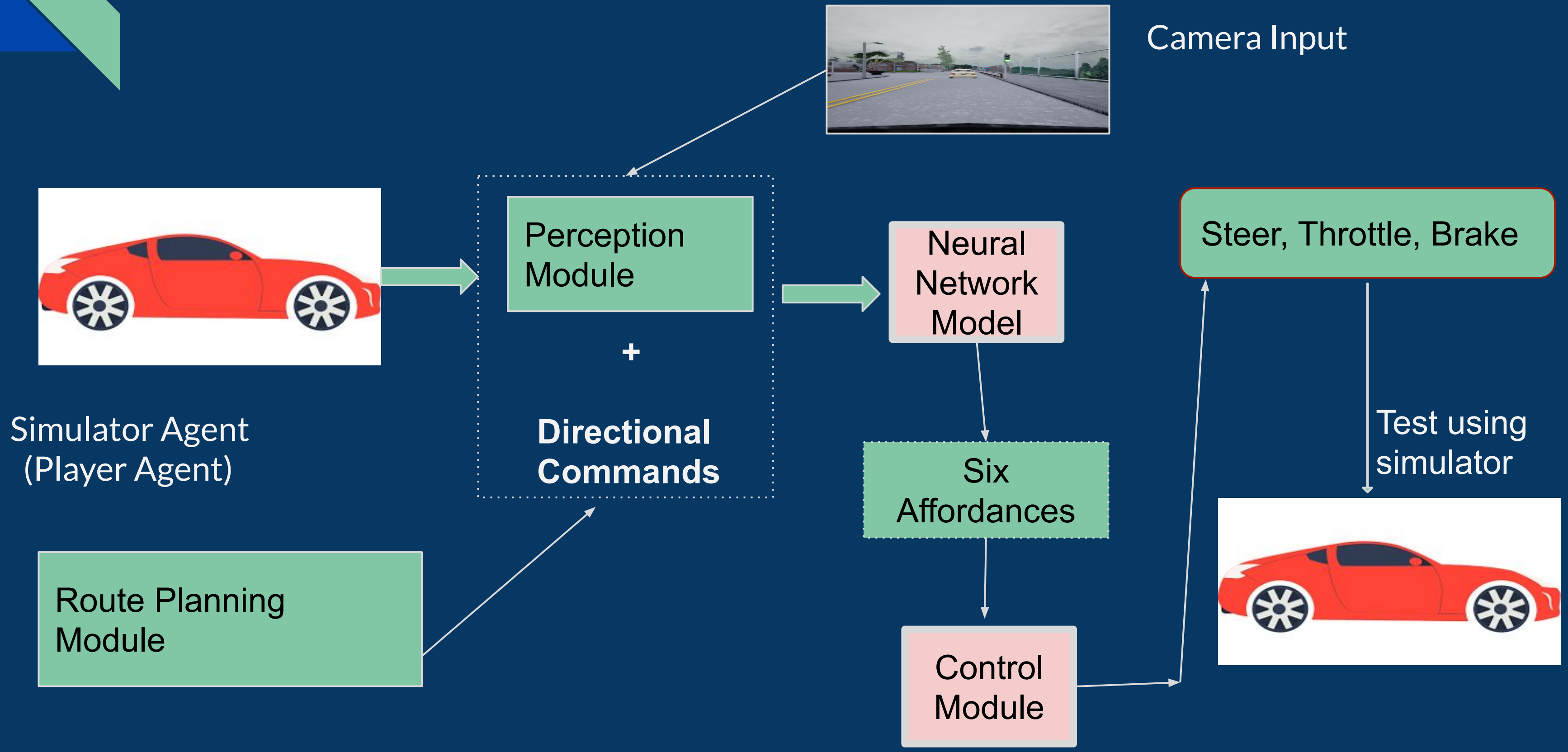
Here is the image of Town01, with 140 point presents in it.



Overview of the system



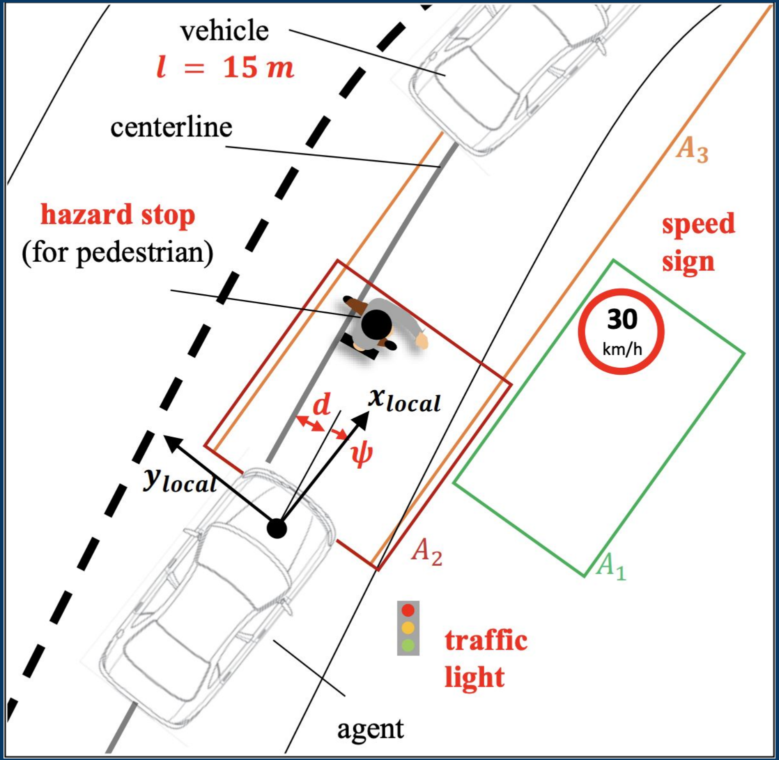
Driving Model Description



Affordance

Details of the usage of the six affordances:

TYPE	CONDITIONAL	AFFORDANCE	VALUES TAKEN
Discrete	No	Hazard Stop	{True, False}
	No	Traffic Light	{True, False}
	No	Speed Sign (km/h)	{None, 30, 60, 90}
Continuous	No	Distance to Vehicle (m)	[0,50]
	Yes	Relative Angle (rad)	[-π ,+π]
	Yes	Distance to Centerline (m)	[-2, +2]



Training strategy

- We collected data (20,793 images) from each of the 3 camera angles from Town-1, using the CARLA simulator
- We concatenated the images of the three cameras.
- We also collected the key frames for every bath of 5 images using ffmpeg , thus reducing the total count to 4159 images.

Training strategy

- Affordances are of 2 types: Discrete and Continuous. Weighted cross entropy loss (H) is used for Discrete affordances, while Mean Absolute Error (MAE) is used for Continuous Affordances.
- A multi task learning component is used to optimise these losses with Adam as optimizer.

$$\mathcal{L} = \sum_{j=1}^{3} H_j + \sum_{k=1}^{3} MAE_k.$$

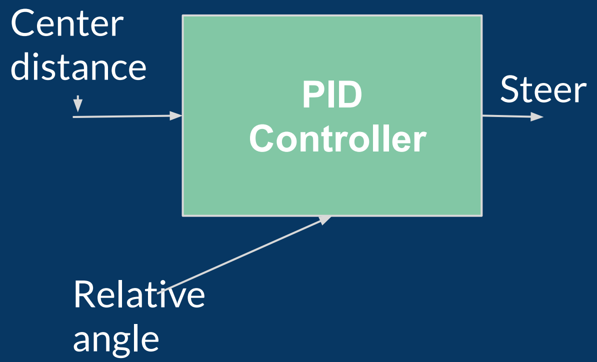
- We trained the model using the collected data for 30 epochs, with a learning rate = 0.001.
- We ran the simulator using the trained model and compared our results with the CAL model by evaluating on the same poses.
- We use PyTorch as our DL framework.

CAL model

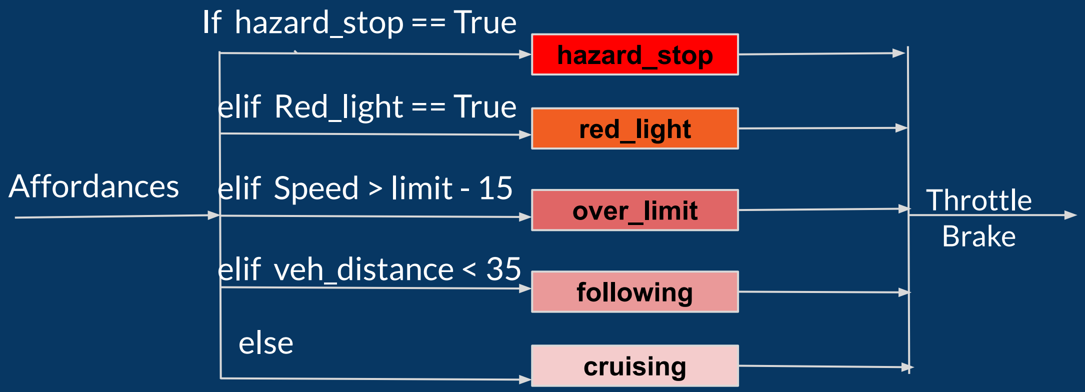
- **Longitudinal control** : The longitudinal controller is subdivided into several states (in ascending importance) : cruising, following, over limit, red light, and hazard stop. [All states are mutually exclusive]

- **Lateral control** : Uses the PID (Proportional Integral Differential) Controller which uses two affordances: the distance to centerline d(t) and the relative angle ψ(t).

Lateral Control



Longitudinal Control

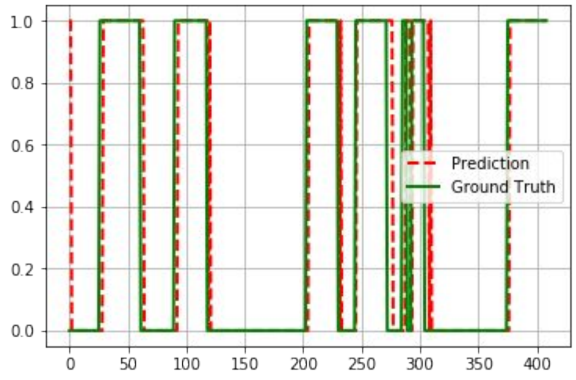


Hyper Parameters used

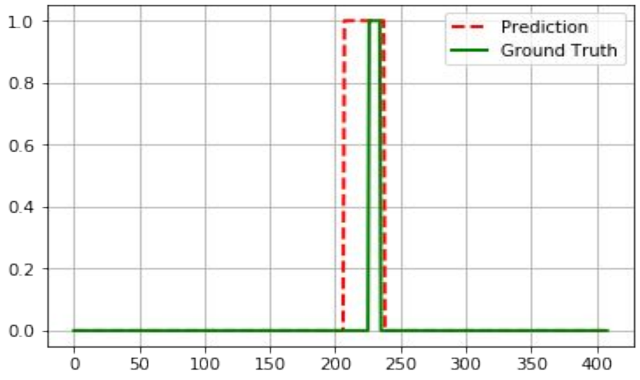
We've tried different hyper parameters for different task blocks, and these listed parameters worked best for our model

Task block	*Layer type*	*Nodes*	*Dropout rate*	*Sequence length*	*Dilation value*
Traffic light	*GRU*	120	0.27	14	2
Hazard stop	*Temp. convolution*	120	0.68	6	1
Speed sign	*MLP*	100	0.55	1	1
Relative angle	*Temp. convolution*	100	0.44	10	1
Center distance	*Temp. convolution*	100	0.44	10	1
Vehicle distance	*GRU*	120	0.38	11	1

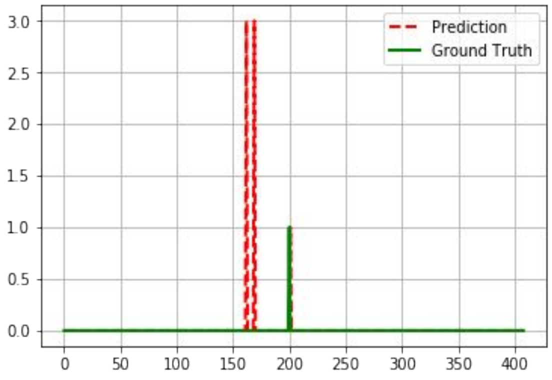
Results & Plots for Discrete Affordances



Red Light
Acc: 90.95



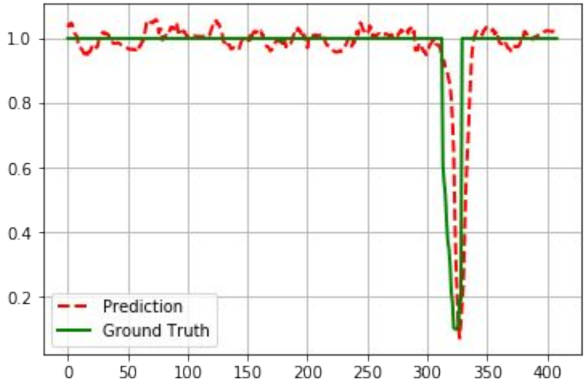
Hazard Stop
Acc: 94.62

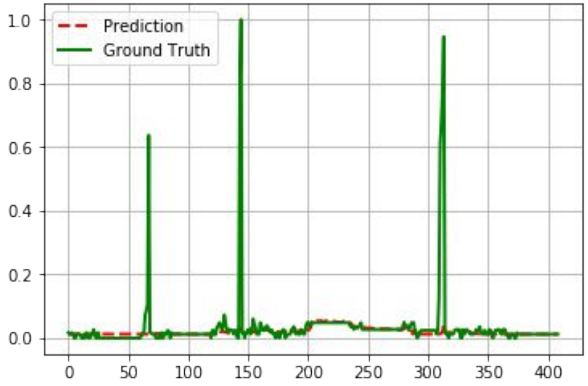


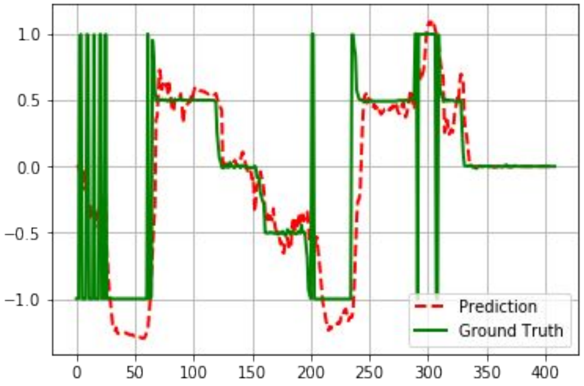
Speed Sign
Acc: 98.29

*Acc. = Accuracy

Results & Plots for Continuous Affordances







Vehicle Distance
MAE: 0.09

Centerline Distance
MAE: 0.21

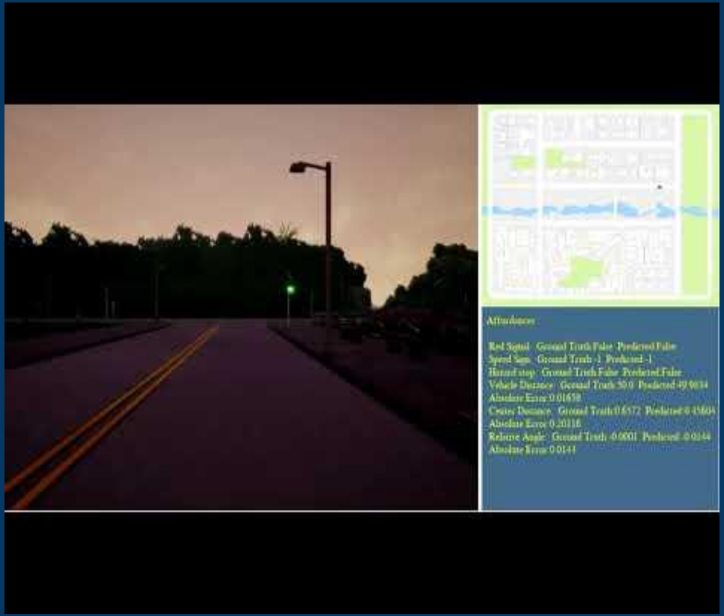
Relative Angle
MAE: 5.02

*MAE = Mean Absolute Error

Car running in Town-1 (pose 138-17)

Using Concatenated Images Model

Using Conditional Affordance Model



Training Town (Town - 1)

Task	Summary	Concatenated Image Model(4159 images)					CAL Model(2,40,000 images)				
		7-3	9-3	138-17	120-117	138-134	7-3	9-3	138-17	120-117	138-134
Navigation Dynamic	Result (Success/ Failure)	1	1	0	1	0	1	1	0	1	0
	Veh_col	0	0	0	0	0	0	0	0	0	0
	Ped_col	0	0	0	0	0	0	0	0	0	0
	Final time	8.75	13.15	98.72	10.2	57.32	8.75	13.15	98.72	10.2	57.31
Navigation	Result (Success/ Failure)	1	1	0	1	0	1	1	0	1	0
	Veh_col	0	0	0	0	0	0	0	0	0	0
	Ped_col	0	0	0	0	0	0	0	0	0	0
	Final time	8.75	13.15	98.72	10.25	57.31	8.75	13.15	98.72	10.2	57.32

Test Town (Town - 2)

Task	Summary	Concatenated Image Model (4159 images)					CAL Model(2,40,000 images)				
		65-69	10-19	23-6	70-73	1-56	65-69	10-19	23-6	70-73	1-56
Navigation Dynamic	Result (Success/ Failure)	1	1	1	1	1	1	1	1	0	0
	Veh_col	0	0	0	0	0	0	0	0	1	0
	Ped_col	0	0	0	0	0	0	0	0	0	0
	Final time	16.4	12.8	13.8	239.85	12.1	18.5	27.0	41.1	388.55	51.40
Navigation	Result (Success/ Failure)	1	1	1	1	1	1	1	1	0	0
	Veh_col	0	0	0	0	0	0	0	0	1	0
	Ped_col	0	0	0	0	0	0	0	0	0	0
	Final time	16.45	12.8	13.8	239.75	12.1	16.65	26.95	44.45	388.54	51.40

Summary of our work

- We used CARLA simulator to collect the images (each of 400 x 300) and ground truth affordances for running the Conditional Affordance Learning model.
- We ran 4 episodes using 3 cameras. We concatenated the 3 camera images which fetched us 20,793 images (each of 1200 x 300 dimension). Having taken keyframes,(1 out of every 5) we created the dataset comprising of 4159 images.
- Performing a grid search, we trained the Conditional Affordance Learning (CAL) Model using the above dataset using the best performing hyperparameters for our data.
- We also ran the agent vehicle on some poses using the above trained model and recorded metrics like Collision.
- Alongside, we ran the agent vehicle using the CAL model(trained on 2,40,000 images) and recorded the same metrics.

Thank you!

We would like to thank Dr. Sourangshu Bhattacharya and Ms. Soumi Das for their constant guidance and support. Without them this project would have been impossible and we wouldn’t have been here!

We learnt how to code better, analyze results, debug and most importantly how to conduct research.

You can find our work here : https://github.com/niladridutt/Conditional-Affordance-Learning