

# Multi-task Learning with Attention for End-to-end Autonomous Driving

Keishi Ishihara<sup>1,2</sup>, Anssi Kanervisto<sup>2</sup>, Jun Miura<sup>1</sup>, Ville Hautamäki<sup>2</sup>

<sup>1</sup>Toyohashi University of Technology

<sup>2</sup>University of Eastern Finland

ishihara@ais1.cs.tut.ac.jp, jun.miura@tut.jp, {anssk, villeh}@cs.uef.fi

## Abstract

Autonomous driving systems need to handle complex scenarios such as lane following, avoiding collisions, taking turns, and responding to traffic signals. In recent years, approaches based on end-to-end behavioral cloning have demonstrated remarkable performance in point-to-point navigational scenarios, using a realistic simulator and standard benchmarks. Offline imitation learning is readily available, as it does not require expensive hand annotation or interaction with the target environment, but it is difficult to obtain a reliable system. In addition, existing methods have not specifically addressed the learning of reaction for traffic lights, which are a rare occurrence in the training datasets. Inspired by the previous work on multi-task learning and attention modeling, we propose a novel multi-task attention-aware network in the conditional imitation learning (CIL) framework. This does not only improve the success rate of standard benchmarks, but also the ability to react to traffic lights, which we show with standard benchmarks.

## 1. Introduction

In the field of autonomous driving, end-to-end behavioral cloning has emerged recently, and many deep networks trained to mimic expert demonstrations have shown reasonable performances from lane-keeping [3, 30] to point-to-point navigation [6, 7]. Typically, those deep networks are trained by using an offline dataset generated by an expert-driver (off-policy), or they learn a control policy by rolling out the environment itself to find good rewards using reinforcement learning (on-policy).

However, previous studies have revealed several critical limitations. Firstly, an agent that learned a control policy from an offline dataset may not be able to make accurate decisions in previously unobserved environments. In a study [7] conducted by using an open-source driving simulator CARLA [10], it was reported that the driving performance of the imitation learning agent considerably drops under

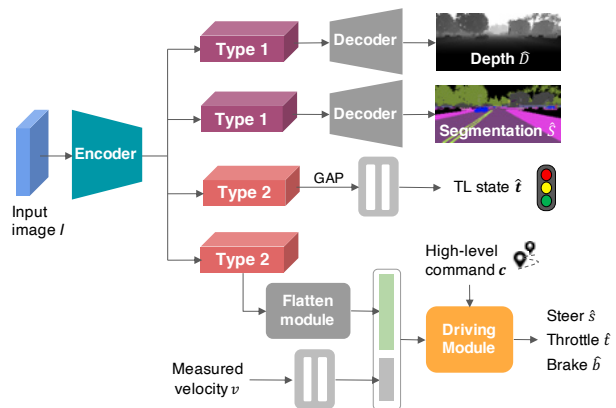


Figure 1: An overview of our proposed multi-task attention-aware network architecture which is composed of seven components: encoder, two decoders, traffic light (TL) state classifier, flatten module, velocity encoder, and driving module. The main target task of this network is to predict control signals from a monocular RGB camera end-to-end. The encoder implements an attention mechanism and generates two types of attention-weighted latent feature maps (see Figure 2).

those conditions such as untrained urban area, weather conditions, and traffic congestion. Secondly, it is important to know how well a network perceives visual inputs for such a safety-critical application of autonomous driving, but only a few studies addressed this issue [8, 21, 24].

Therefore, in this work, we tackle the aforementioned problems based on the following ideas: (1) Learning visual recognition sub-tasks (e.g., semantic segmentation) alongside can encourage the network to learn generalizable scene representations that are effective for decision-making. Similarly to [21], we also introduce multi-tasking to decode scene representations as sub-tasks. Although, in the previous study, the learning process was divided into perception and control, we train all tasks at the same time. Additionally, we adopt traffic light classification as one of the sub-

tasks. In this way, an operator can easily debug the model performance by looking at its predictions. (2) Also, an attention mechanism is introduced to further improve control performance by focusing on salient regions among the extracted features like humans do when driving. Our main contributions can be summarized as follows:

- We propose a novel multi-task attention-aware network for vision-based end-to-end autonomous driving.
- We show that our approach achieves better or comparable results with the current state-of-the-art models on the CARLA benchmarks [7, 10].
- We further perform the traffic light infraction analysis to quantitatively show the effectiveness of our approach in handling traffic lights.
- We study how attention layers change what network focuses on by visualizing saliency maps.

## 2. Related work

**Behavioral cloning for autonomous driving** Bojarski *et al.* [3] were the first to successfully demonstrate lane following task in an end-to-end (image-to-steering) manner using a simple CNN. Later, Xu *et al.* [30] used a large-scale video dataset to predict vehicle egomotion. To extend the learned control policy to goal-directed navigation to solve the ambiguous action problem (e.g., intersections), Codevilla *et al.* [6] proposed Conditional Imitation Learning (CIL) framework. The goal of CIL is to navigate in an urban environment where the autonomous vehicle must take turns at intersections based on high-level command input such as “turn left”, “turn right”, and “go straight”. Because of CIL and open-source access to realistic driving simulator CARLA [10], many follow-up works have been conducted in CIL framework [4, 7, 8, 21, 22, 26] which called a vision-based driving system where perception relies on visual input from cameras only to keep the entire system simple.

**Attention in vision models** In the field of computer vision, attention has been a key idea to improve performance of CNNs in various tasks such as classification [16, 29], object detection [11], image tracking [9], and captioning [31]. Katharopoulos and Fleuret [17] proposed an attention-sampling approach to process megapixel images. Liu *et al.* [23] presented the Multi-Task Attention Network (MTAN) that has a feature-level attention mechanism to select task-specific features for multi-task learning. Usually, incorporating an attention mechanism requires a unique architectural design like the abovementioned ones. In contrast, module-based attention approaches [16, 28, 29] are another trend that can simply be attached to existing

CNNs [5, 14, 15] and improves performances. Convolutional Block Attention Module (CBAM) [29] is a representative of them, where attention is learnable end-to-end without requiring additional labels. A few approaches have also introduced attention in the autonomous driving context. Mori *et al.* [24] used Attention Branched Network [12] to develop a visually interpretable model. Similarly, Kim and Canny [18] visualized saliency part to predict steering by decoding attention heat maps.

An approach similar to ours by Cultrera *et al.* [8] splits features into multi-scale grids and weighs with softmax scores to drop out irrelevant regions. In comparison, our approach uses CBAM specialized for one particular task in multi-task architecture to emphasize features over the channel and spatial dimensions separately and refines specifically to the task. Also, our attention mechanism is based on MTAN [23], which originally handles dense estimation tasks only such as semantic segmentation and depth estimation, while we design it to fit into the CIL framework [6] and our model performs control prediction in an end-to-end manner as well.

## 3. Methodology

Inspired by the previous works of end-to-end driving [6, 21] and visual attention methods [23, 29], we introduce a multi-task attention-aware neural network that learns compact road scene representation to jointly predict scene representations (semantic segmentation and depth estimation), color state of traffic light, and driving control.

First of all, we consider the autonomous driving problem in an urban environment as a goal-directed motion planning task. To this end, we adopt the Conditional Imitation Learning (CIL) framework [6] to train our model in an end-to-end manner.

Secondly, our approach learns semantic segmentation and depth estimation to learn to encode the image into a meaningful feature vector as in [21]. Although the authors of [21] were the first to introduce those scene representation learning in the CIL framework, the learning process was divided into perception and driving. In our approach, we train segmentation and depth estimation together with control. Moreover, we also add traffic light classification task as the fourth task that our network performs at the same time as the others.

Thirdly, our approach incorporates an attention mechanism inspired by the MTAN proposed by Liu *et al.* in [23] to successfully learn task-shared and task-specific features separately. Differently from the original MTAN which drops irrelevant features from global feature maps, we use a CBAM-based attention module to emphasize the global feature maps by additive operation. This contributes to the learning of much more complex behavior of reacting to traffic lights than the ability to keep lane and following vehi-

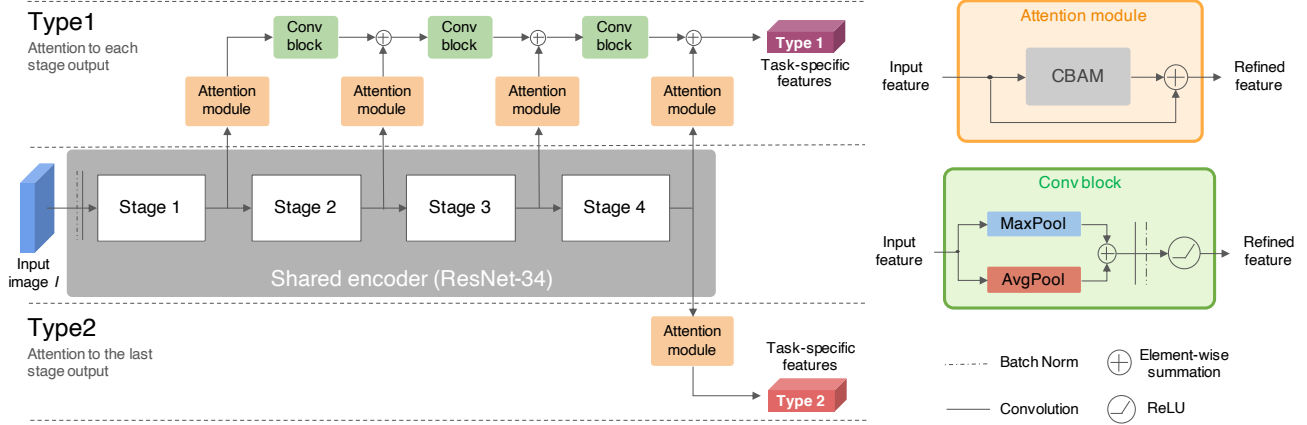


Figure 2: Visualization of the ResNet34-based encoder network with two attention mechanisms. This encoder outputs two types of task-specific attention-weighted feature maps: *Type1* and *Type2*, generated by its attention paths formed by attention modules and convolution blocks, similar to MTAN [23].

cles.

### 3.1. Network architecture

Figure 1 gives an overview of the architectural design of our proposed approach which is composed of seven components: encoder, two decoders for scene representations, traffic light state classifier, flatten module, velocity encoder, and driving module. Our network model is built on the two state-of-the-art approaches in the CARLA benchmark; the Conditional Imitation Learning approach with ResNet-34 backbone and Speed prediction head (CILRS) proposed by Codevilla *et al.* in [7], and the Multi-Task learning approach (MT) proposed by Li *et al.* in [21].

In the CIL framework, the agent is conditioned by the external navigational command (e.g., turn left) and asked for the control command corresponding to it so that the learned agent can be controlled by that command at test time. This eliminates the ambiguous problem at intersections where multiple actions can be taken [6].

Our model’s inputs are a monocular RGB camera image  $I$  from a front-facing camera, a measured velocity  $v$  of the ego-vehicle, and a high-level command  $c$  which is a one-hot encoded vector of (“follow lane”, “turn left”, “turn right”, “go straight”). Our model performs predictions of control signals (steering  $\hat{s}$ , throttle  $\hat{t}$ , and brake  $\hat{b}$ ) as main task along with three different complementary sub-tasks: semantic segmentation  $\hat{S}$ , depth estimation  $\hat{D}$ , and classification of the traffic light state  $\hat{t}$  which is one-hot vector of (“red”, “yellow”, “green”, “none”) at every frame. By doing such sub-tasks, we suppose that the model can capture road scenes without missing necessary information for generating control signals and encode those into its latent feature maps. Additionally, we employ a simple channel-spatial at-

tention mechanism described in the next paragraph inspired by the study [23] where task-specific attention modules are used to select and emphasize features from one shared encoder for different image recognition tasks.

**Residual network with attention** Figure 2 shows the encoder module that consists of ResNet-34 [14] for the backbone network, attention modules, and convolutional blocks. Using these attention modules and convolutional blocks, we build two types of task-specific attention paths along with the shared backbone encoder: *Type1* and *Type2*. *Type1* is formed by a set of attention modules and convolution blocks, which then takes convolution feature maps from every end of the ResNet stages. The attention modules play as feature refiners that emphasize the global feature maps specifically to a certain task. In comparison, MTAN [23] used attention to select which features to use, while we emphasize them via an additive operation.

The *Type2* is made of an attention module that refines the feature maps obtained from the last layer of the ResNet. In total, we prepare two *Type1* attention paths for segmentation and depth estimation, and two *Type2* paths for traffic light classification and control command prediction.

We have empirically found that combining the two types of attention mechanisms results in better performance than using only *Type1* or *Type2* individually for all tasks. Intuitively this is reasonable because tasks that require dense estimation, such as segmentation and depth estimation, may often be better to have access to higher resolution feature maps to avoid missing details, as is a common technique in architectures like U-Net [25]. Conversely, tasks such as control prediction and traffic light classification may require more abstract features in the hidden representation. For the

attention module, we adopt CBAM [29].

**Decoder networks** As shown in Figure 1, we build two decoder networks for the image decoding sub-tasks: semantic segmentation and depth estimation, each receiving *Type1* task-specific features from the encoder. The goal of those sub-tasks is to learn compact and generalizable latent representation for control prediction as done in [21, 30]. This multi-tasking approach enforces the encoder to generate representations such that they encode “what”, “where”, and “how far” information in the feature maps. The resolution of both decoded images is the same as the input image resolution of  $384 \times 160$ . The decoder architectures are based on [21], but empirically we have found that performance is improved when the segmentation decoder larger than the depth one.

**Traffic light classifier** Reacting to the traffic lights has not yet been explicitly modeled in vision-based end-to-end driving approaches [6, 8, 21]. In some works [4, 7], the authors addressed the learning of the behavior of reacting to the traffic lights from only the demonstration of stopping for a red light and starting going forward when it turns green. However, in situations where an agent is facing a traffic light, it is important for humans to know which state of the traffic light the agent is aware of. Also, the ability to classify the light state can help the network learn the behavior of reacting to it according to the color. Therefore, we introduce traffic light state classification as one of the sub-tasks. Our approach classifies a traffic light state at every frame into one of four classes including (“red”, “yellow”, “green”, “none”).

**Driving module** The driving module takes a vector feature map denoted as  $j$  produced by concatenating flattened feature map, encoded velocity input  $v$ , high-level command  $c$  and outputs control commands (steering  $\hat{s}$ , throttle  $\hat{t}$  and braking signal  $\hat{b}$ ), which are referred to as regression tasks. Following prior works in the CIL framework [4, 6, 7, 8, 21, 26], our driving module also implements branched prediction head: four branches corresponding to each navigational command. Command input selects which of these branches is used to predict control commands. The agent that has the command-dedicated prediction head performs better than that with one prediction head, which is reported by Codevilla *et al.* in [6].

### 3.2. The model objective

To the network, we define the objective function  $\mathcal{L}_{total}$  as the weighted sum of control regression loss  $\mathcal{L}_{control}$ , traffic light state classification loss  $\mathcal{L}_{tl}$ , semantic segmentation loss  $\mathcal{L}_{seg}$ , and depth estimation loss  $\mathcal{L}_{dep}$  with the

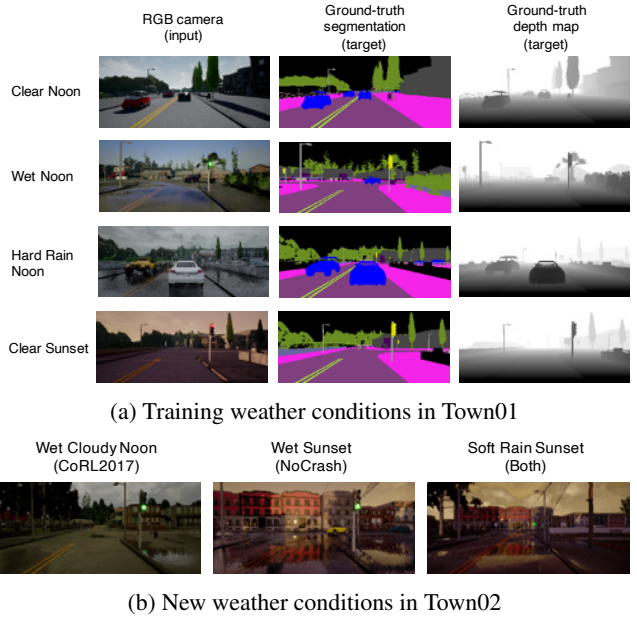


Figure 3: Example frames of the collected dataset (training conditions in Town01) and new weather conditions in Town02.

task weightings denoted by  $\lambda$  with each task subscript:

$$\mathcal{L}_{total} = \lambda_{control} \mathcal{L}_{control} + \lambda_{tl} \mathcal{L}_{tl} + \lambda_{seg} \mathcal{L}_{seg} + \lambda_{dep} \mathcal{L}_{dep} \quad (1)$$

Control loss  $\mathcal{L}_{control}$  is the weighted linear combination of steering, throttle, and brake losses  $\mathcal{L}_c$  with weighting terms  $\gamma_c$ , where  $c = 1$  is for steering,  $c = 2$  for throttle, and  $c = 3$  for brake:

$$\mathcal{L}_{control} = \sum_{c=1}^3 \gamma_c \mathcal{L}_c \quad (2)$$

We use a mean squared error for each control regression loss  $\mathcal{L}_c$ . For traffic light state classification and semantic segmentation, we use class-weighted categorical cross-entropy losses:  $\mathcal{L}_{seg}$  calculated between the network output  $\hat{S}$  and ground-truth label  $S$ , and  $\mathcal{L}_{tl}$  done between  $\hat{t}$  and  $t$ . Lastly, for depth estimation, we use a pixel-wise mean squared error between the network output  $\hat{D}$  and the ground-truth label  $D$ . All coefficients  $\lambda$  and  $\gamma$  are obtained empirically by examining validation errors to determine when they were valid.

### 4. Environment and dataset

We perform all our experiments in CARLA simulator [10] version 0.8.4. This version of the simulator provides two different suburban towns: Town01 for training, and Town02 for testing. It has a diverse, non-deterministic

environment where dynamic obstacles may appear (e.g. auto-cruising vehicles and pedestrians) and weather conditions can also change by user preferences. In particular, pedestrians have random behaviors; crossing the street without any previous notice, which is unpredictable and thus an agent needs to react quickly as described in [7]. Moreover, they may crash themselves into a car stopping at a red light or an obstacle, resulting in an incomplete evaluation.

#### 4.1. Dataset collection

We collected our dataset simply because provided one by Codevilla *et al.* in [7] do not contain traffic light information. To do so, we adapt a hand-coded expert autopilot provided by Codevilla *et al.* in [6], which leverages simulated information to follow the waypoints nearly perfectly and stop for forward-driving vehicles and pedestrians to avoid collisions. It also reacts to the traffic lights regarding their color state. To obtain the ground-truth label of the light state, the original code provided was modified to save it additionally. When multiple lights are visible, the state of the light facing the ego-vehicle is saved as the ground-truth label. We collected 466,000 frames from Town01 and randomly split into 372,000 of the training set and 94,000 of the validation set. Each frame consists of front-faced monocular RGB camera image  $I$ , auto-generated semantic segmentation  $S$ , depth map  $D$ , and measurements including steering  $s$ , throttle  $t$ , brake  $b$ , speed  $v$ , high level command  $c$ , and ground-truth traffic light state  $t$ . The dataset is collected under four different weather conditions (“Clear Noon”, “Wet Noon”, “Hard Rain Noon”, “Clear Sunset”) as it is specified condition by the CARLA benchmarks, described more in Section 5.3.

#### 4.2. Data augmentation and balancing

We found that data augmentation and balancing are crucial for better generalization performance as reported in previous works [6, 21]. In order to enlarge training dataset distribution, we use the same set of common image augmentations as done in [6], including gaussian noise, blurring, pixel dropout, and contrast normalization during training. Furthermore, those image transformations are followed by PCA color augmentation which is proposed in [20] across every mini-batch.

Additionally, it is well known that driving datasets have inherently unbalanced distribution in general, especially in control [2]. Our dataset is no exception and thus contains mostly driving forwards, so without balancing agent only drives forward. To ease the bias of the skewed distribution of our dataset, we performed undersampling based on steering values during training. Specifically, a sample has a smaller chance of being selected if it is overrepresented in the dataset. Consequently, straight driving scenes which are

Training conditions (4 weathers, Town01)				
Task	MT	CILRS	Ours w/o Att.	Ours
Straight	<b>100</b> ± 0	98 ± 3	99 ± 1	<b>100</b> ± 0
One Turn	<b>100</b> ± 1	92 ± 7	97 ± 4	<b>100</b> ± 1
Navigation	<b>100</b> ± 0	97 ± 4	98 ± 3	<b>100</b> ± 0
Nav. Dyn.	<b>99</b> ± 3	95 ± 4	96 ± 4	<b>99</b> ± 2
New weather (2 weathers, Town01)				
Task	MT	CILRS	Ours w/o Att.	Ours
Straight	<b>100</b> ± 0	95 ± 5	98 ± 4	<b>100</b> ± 0
One Turn	<b>100</b> ± 1	85 ± 8	94 ± 7	99 ± 3
Navigation	<b>100</b> ± 0	71 ± 6	94 ± 9	97 ± 6
Nav. Dyn.	<b>99</b> ± 3	70 ± 4	96 ± 7	97 ± 4
New town (4 weathers, Town02)				
Task	MT	CILRS	Ours w/o Att.	Ours
Straight	98 ± 4	95 ± 4	98 ± 4	<b>99 ± 3</b>
One Turn	93 ± 6	85 ± 7	92 ± 8	<b>98 ± 2</b>
Navigation	81 ± 12	71 ± 9	84 ± 14	<b>93 ± 10</b>
Nav. Dyn.	78 ± 16	70 ± 14	82 ± 13	<b>91 ± 11</b>
New town & weather (2 weathers, Town02)				
Task	MT	CILRS	Ours w/o Att.	Ours
Straight	99 ± 2	92 ± 6	<b>100</b> ± 0	99 ± 1
One Turn	<b>99</b> ± 2	83 ± 12	91 ± 8	<b>99</b> ± 1
Navigation	88 ± 7	68 ± 24	90 ± 9	<b>96</b> ± 6
Nav. Dyn.	86 ± 11	67 ± 21	88 ± 9	<b>91</b> ± 9

Table 1: Comparison of the success rate of the proposed approach to the state-of-the-art models on the original CARLA benchmark (CoRL2017) [10]. Mean and standard deviation over three repetitions are reported (MT [21] has two repetitions). For all methods, seed is not fixed at training time. Our approach emits competitive or better results than the previous state-of-the-art models of end-to-end driving.

the dominant mode in the dataset are often skipped when training. Without this balancing, our agents rarely completed full episodes in the benchmarks.

## 5. Experiments

### 5.1. Training setup

We implemented the proposed approach in TensorFlow 2.3.0 framework [1]. In contrast to the [21], we do not divide the training process into stages. We use a mini-batch size of 32 and the Adam optimizer [19] in which the initial learning rate is set to 0.005. When training, we validate the presented model at the end of every epoch, and then if the validation error of control does not record the best for over five epochs, we divide the learning rate by 5. The training lasts for approximately two days on a single Nvidia GeForce GTX 1080Ti. The resolution of all images

Task	Training conditions				New weather			
	MT	CILRS	Ours w/o Att.	Ours	MT	CILRS	Ours w/o Att.	Ours
Straight	<b>0.0</b> ±0.0	1.0±1.7	0.7±1.5	0.3±1.1	1.0±1.7	2.9±2.1	3.6±4.0	<b>0.0</b> ±0.0
One Turn	0.3±0.7	1.1±0.9	0.8±1.2	<b>0.0</b> ±0.0	<b>0.0</b> ±0.0	0.6±1.3	1.1±1.1	<b>0.0</b> ±0.0
Navigation	1.0±1.1	1.8±1.3	0.8±1.0	<b>0.1</b> ±0.3	0.6±0.6	0.6±0.6	0.6±0.9	<b>0.4</b> ±0.6
Nav. Dyn.	0.6±0.6	1.1±1.3	0.6±1.1	<b>0.5</b> ±0.9	1.5±0.5	2.6±1.1	2.0±1.3	<b>0.2</b> ±0.4

Task	New town				New town & weather			
	MT	CILRS	Ours w/o Att.	Ours	MT	CILRS	Ours w/o Att.	Ours
Straight	15.9±10.1	8.3±4.5	10.3±3.5	<b>5.4</b> ±4.5	17.7±5.9	17.3±20.5	11.8±3.4	<b>7.8</b> ±6.5
One Turn	11.1±5.3	11.3±4.8	8.2±3.9	<b>4.1</b> ±3.7	15.7±6.7	22.3±19.8	14.8±8.1	<b>4.2</b> ±3.3
Navigation	9.7±6.4	12.0±6.7	12.9±2.8	<b>8.7</b> ±4.9	13.6±5.6	16.8±11.9	18.7±17.1	<b>11.4</b> ±6.9
Nav. Dyn.	8.7±3.5	9.8±4.6	11.4±7.1	<b>6.1</b> ±3.6	<b>9.5</b> ±4.0	10.1±3.8	10.0±1.75	14.2±10.5

Table 2: Traffic light infraction analysis on CoRL2017 benchmark [10]. Percentage of times an agent crossed on red traffic lights is reported with mean and standard deviation over three repetitions except for MT [21] that has two repetitions (lower is better). Our proposed approach records a smaller number of infractions under most tasks. Note that the MT method contains traffic light classifier in our implementation, not present in the original MT [21].

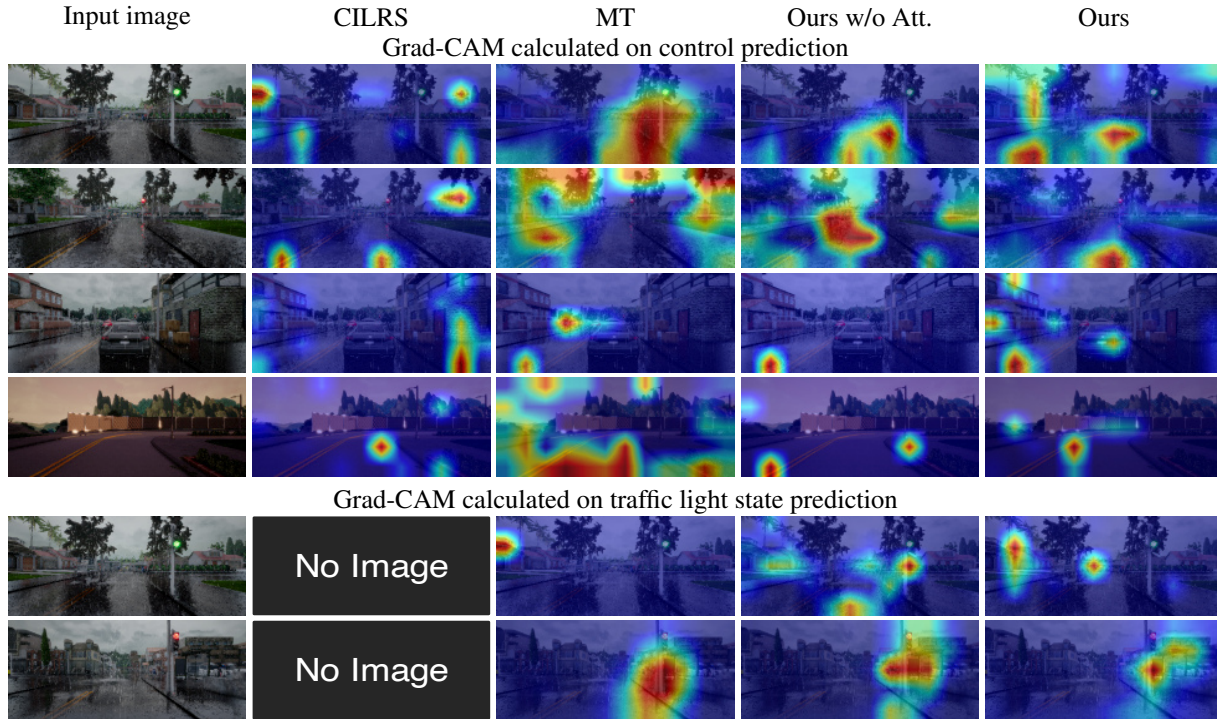


Figure 4: Randomly picked Grad-CAM [27] visualization results. CILRS does not have traffic light prediction. Shown results on control prediction (top four rows) are summed over individual Grad-CAM results done on each control output.

is  $384 \times 160$ , including depth and semantic segmentation images. Without setting a random seed, we repeated all methods three times from training to evaluation (except for MT, which was repeated only twice), while the best seed out of five runs was selected in [7]. This may result in larger standard deviations in the results, but the reproducibility

may also be higher. Note that we have come up with the above hyperparameter settings through experiments.

## 5.2. Baselines

We compare our proposed method to the two methods we build on: the conditional imitation learning approach

Training conditions (4 weathers, Town01)				
Task	MT	CILRS	Ours w/o Att.	Ours
Empty	<b>100</b> ± 0	94 ± 6	97 ± 4	99 ± 1
Regular	90 ± 3	87 ± 5	<b>91</b> ± 5	88 ± 5
Dense	46 ± 11	41 ± 6	45 ± 14	<b>53</b> ± 11
New weather (2 weathers, Town01)				
Task	MT	CILRS	Ours w/o Att.	Ours
Empty	<b>98</b> ± 3	85 ± 9	94 ± 7	97 ± 4
Regular	84 ± 3	76 ± 8	87 ± 10	<b>93</b> ± 5
Dense	<b>45</b> ± 9	35 ± 6	42 ± 7	44 ± 12
New town (4 weathers, Town02)				
Task	MT	CILRS	Ours w/o Att.	Ours
Empty	66 ± 14	57 ± 11	76 ± 11	<b>90</b> ± 12
Regular	50 ± 10	44 ± 11	62 ± 13	<b>69</b> ± 11
Dense	20 ± 4	17 ± 8	23 ± 5	<b>37</b> ± 9
New town & weather (2 weathers, Town02)				
Task	MT	CILRS	Ours w/o Att.	Ours
Empty	57 ± 10	37 ± 11	71 ± 11	<b>81</b> ± 11
Regular	44 ± 7	30 ± 11	61 ± 8	<b>67</b> ± 9
Dense	<b>24</b> ± 10	14 ± 5	22 ± 10	23 ± 5

Table 3: Results on the NoCrash benchmark [7] with mean and standard deviation over three runs except for MT [7] that has two runs. For all methods, seed is not fixed when training.

(CILRS) proposed by Codevilla *et al.* [7] and the original multi-task learning approach (MT) proposed by Li *et al.* [21] (see Section 3 for detailed information). For a fair comparison, we implement the two approaches with the same setup as ours using our dataset and the same network module for all methods (just reflecting the original idea of the papers), so results are different from the original ones. Also, note that MT contains a traffic light classifier that does not appear in the original implementation.

### 5.3. Experimental setup

We evaluate our proposed method on the original CARLA benchmark named CoRL2017 [10] and on the NoCrash benchmark [7]. Both evaluate an agent in a goal-directed navigation scenario whether or not the agent can accomplish point-to-point navigation in a suburban setting. Each benchmark defines a fixed number of starting and goal points in both Town01 and Town02, accompanied by a route planner that computes a global path between starting and goal points using the A\* algorithm [13]. This planner sends the high-level command  $c$  at every frame considering the location of an agent to indicate which direction it should proceed from its current position. All evaluation trials are done in four generalization contexts: training condition, new weather, new town, and new town & new weather.

“Training condition” indicates that each trial is done under four training weather conditions in Town01, which are encountered when training, whereas “new weather” specifies two new weather conditions that do not appear when training (see examples for the weather conditions shown in Figure 3). Also, as suggested by its name, “new town” indicates Town02 which is unseen when training, and thus, “new town & new weather” specifies the most difficult full generalization test.

The CoRL2017 benchmark consists of four different driving conditions, which are “driving straight”, “driving with one turn at an intersection”, “full navigation with multiple turns at intersections”, and “The same full navigation but with dynamic obstacles”. A trial is considered successful if an agent reaches within 2 meters from the defined goal point. Passing through a red traffic light is not counted as a failure. Therefore, this benchmark is mainly designed for evaluating the ability of lane-keeping and taking turns to navigate to the specified goal, but those high-level behaviors like handling dense traffic scenarios are not required.

We use NoCrash benchmark [7] for more complex scenarios. It measures the ability of agents to handle more complex events in both towns under six weather conditions as in CoRL2017. This benchmark defines 25 goal-conditioned routes with a difficulty level equivalent to the “full navigation” condition of the CoRL2017 benchmark, with three traffic congestions: 1) empty town: no dynamic agents, 2) regular traffic: moderate number of cars and pedestrians, 3) dense traffic: a large number of pedestrians and heavy traffic.

Unlike in CoRL2017, the NoCrash benchmark considers an episode as a failure if a collision bigger than a predefined threshold occurs. This makes the evaluation more similar to the ones in the real-world setting where the counted number of human interventions per kilometer is a common metric and those human interventions will put the vehicle back to the safe state [7]. Additionally, this benchmark also measures the percentage of traffic light violations, however, an infraction of crossing red traffic light does not terminate a trial.

### 5.4. Evaluation results

The results for the CoRL2017 benchmark are shown in Table 1, where we can see our implementation outperforms the baselines. Also, both of our models outperform or are comparable with the current state-of-the-art CILRS. Especially, although the performance of the CILRS model decreases as the driving task gets harder in the “new town” and “new town & weather” condition, the other models that have multi-task knowledge can keep its performance to some extent, and furthermore, our proposed approach stays at over 91 in all tasks. Table 2 reports the percentage of times an agent crossed on red traffic lights in each driving

Task	Training conditions				New weather			
	MT	CILRS	Ours w/o Att.	Ours	MT	CILRS	Ours w/o Att.	Ours
Empty	<b>0.7</b> ±0.7	1.8±1.4	1.0±0.9	0.8±0.6	<b>0.7</b> ±0.8	3.0±1.4	1.2±1.1	1.8±1.2
Regular	<b>0.8</b> ±0.7	1.7±1.5	<b>0.8</b> ±0.9	1.0±0.7	<b>0.6</b> ±0.6	3.3±1.1	1.2±1.1	1.9±1.4
Dense	9.3±13.4	7.3±5.2	<b>3.0</b> ±2.1	10.7±19.1	<b>4.4</b> ±1.2	13.8±10.1	4.8±2.8	13.5±17.7

Task	New town				New town & weather			
	MT	CILRS	Ours w/o Att.	Ours	MT	CILRS	Ours w/o Att.	Ours
Empty	8.9±5.6	10.8±7.8	10.6±2.2	<b>7.1</b> ±4.1	15.8±5.2	16.5±13.8	21.1±14.7	<b>15.5</b> ±14.4
Regular	6.5±3.9	7.2±4.2	7.1±1.9	<b>5.1</b> ±2.5	<b>8.9</b> ±3.7	10.8±4.5	9.4±2.4	13.2±17.1
Dense	<b>11.2</b> ±6.2	12.1±8.8	12.8±8.7	16.3±19.1	8.1±3.6	8.9±3.4	21.8±19.0	<b>3.1</b> ±2.5

Table 4: Traffic light infraction analysis on NoCrash benchmark [7]. As in Table 2, percentage of times an agent crossed on red traffic lights is reported with mean and standard deviation over three repetitions except for MT that has two repetitions (lower is better). As in Table 2, MT contains traffic light classifier in our implementation, not present in the original MT [21].

condition on the CoRL2017 benchmark. From this table, we can see that our proposed approach is much less likely to experience the infraction of crossing on red traffic lights in every condition. Thus, adding our task-specific attention mechanism can enhance the capability of reacting to traffic lights.

The NoCrash benchmark results are reported in Table 3, along with the traffic light violation rates reported in Table 4. Similar to the results on the first CoRL2017 benchmark, our proposed approach overall records the best numbers in the “new town”. We noticed, when we were visualizing the benchmarking runs, there were some scenes where the agent got stuck every time in those situations where it was exposed to strong direct sunlight from the front covering huge regions of the input image. We believe the noisy numbers of traffic violations are a result of visual difference between training and testing set (especially “Wet Sunset” and “Soft Rain Sunset” conditions in testing) and the crowded traffic in the NoCrash benchmark.

Comparing MT with our proposed approach without attention, these two approaches showed overall the same performance on the NoCrash benchmark, but occasionally MT appeared to overfit the training town (also in CoRL2017). This gives us an intuition that learning the control task and scene representation simultaneously allows the model to learn a more generalizable representation. Interestingly, the MT approach outperformed the CILRS both in the success rate of episodes and in handling traffic lights, which was the opposite of the results reported in [7].

### 5.5. Grad-CAM visualizations

For the qualitative analysis, we use the Grad-CAM visualization [27] shown in Figure 4 for all methods using randomly picked images from training conditions. Grad-CAM allows us to study what parts of the input play important role in deciding the output. Firstly, we visualized salient parts

that contributed to control outputs. We can see that Grad-CAM attention maps of all methods may cover meaningful regions like the center line or the border between road and sidewalk. When taking a close look, CILRS might spot irrelevant points and MT appeared to be distracted into large regions, while our proposed approach looks at narrow yet important points like the center of the road or leading vehicle. Secondly, we also visualized attention maps based on traffic light prediction. MT and our proposed approach without attention widely capture the traffic light including the pole when it is red, whereas our proposed approach attentively looks at it but the red color point is not included.

## 6. Conclusion

In this work, we have presented a novel multi-task attention-aware network model for end-to-end autonomous driving in conditional imitation learning framework [6]. This model uses two types of attention paths to generate task-specific feature maps fed into each task-specific module. To verify its effectiveness, we conducted experiments on two CARLA benchmarks [7, 10] and quantitatively showed that the attention improves the learned control policy including the ability to handle traffic lights, outperforming the baseline methods. By visualizing the attended regions using Grad-CAM, we find that our model attends to correct points or objects of interest (e.g. center lines, other cars) when making control decisions.

One future direction would be to include temporal information as well, e.g. by using videos. This way, the end-to-end driving model can more precisely capture the visual inputs by distinguishing dynamic and static objects. Also, it would be interesting to see if the proposed method can transfer to a real-world data such as BDD100K dataset [32], after which it would be possible to show how a real car would behave when controlled by our approach.



## References

- [1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org. [5](#)
- [2] Alexander Amini, Wilko Schwarting, Guy Rosman, Brandon Araki, Sertac Karaman, and Daniela Rus. Variational autoencoder for end-to-end control of autonomous driving with novelty detection and training de-biasing. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 568–575. IEEE, 2018. [5](#)
- [3] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D. Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, Xin Zhang, Jake Zhao, and Karol Zieba. End to end learning for self-driving cars, 2016. [1](#), [2](#)
- [4] Dian Chen, Brady Zhou, Vladlen Koltun, and Philipp Krähenbühl. Learning by cheating. In *Conference on Robot Learning (CoRL)*, 2019. [2](#), [4](#)
- [5] F. Chollet. Xception: Deep learning with depthwise separable convolutions. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1800–1807, 2017. [2](#)
- [6] Felipe Codevilla, Matthias Müller, Antonio López, Vladlen Koltun, and Alexey Dosovitskiy. End-to-end driving via conditional imitation learning. In *International Conference on Robotics and Automation (ICRA)*, 2018. [1](#), [2](#), [3](#), [4](#), [5](#), [8](#)
- [7] Felipe Codevilla, Eder Santana, Antonio M. Lopez, and Adrien Gaidon. Exploring the limitations of behavior cloning for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. [1](#), [2](#), [3](#), [4](#), [5](#), [6](#), [7](#), [8](#)
- [8] Luca Cultrera, Lorenzo Seidenari, Federico Becattini, Pietro Pala, and Alberto Del Bimbo. Explaining autonomous driving by learning end-to-end visual attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 340–341, 2020. [1](#), [2](#), [4](#)
- [9] Misha Denil, Loris Bazzani, Hugo Larochelle, and Nando de Freitas. Learning where to attend with deep architectures for image tracking. *Neural computation*, 24(8):2151–2184, 2012. [2](#)
- [10] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16, 2017. [1](#), [2](#), [4](#), [5](#), [6](#), [7](#), [8](#)
- [11] Qi Fan, Wei Zhuo, Chi-Keung Tang, and Yu-Wing Tai. Few-shot object detection with attention-rpn and multi-relation detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4013–4022, 2020. [2](#)
- [12] Hiroshi Fukui, Tsubasa Hirakawa, Takayoshi Yamashita, and Hironobu Fujiyoshi. Attention branch network: Learning of attention mechanism for visual explanation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. [2](#)
- [13] Peter Hart, Nils Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968. [7](#)
- [14] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. [2](#), [3](#)
- [15] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. [2](#)
- [16] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. [2](#)
- [17] Angelos Katharopoulos and François Fleuret. Processing megapixel images with deep attention-sampling models. In *International Conference on Machine Learning*, pages 3282–3291. PMLR, 2019. [2](#)
- [18] Jinkyu Kim and John Canny. Interpretable learning for self-driving cars by visualizing causal attention. In *Proceedings of the IEEE international conference on computer vision*, pages 2942–2950, 2017. [2](#)
- [19] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. [5](#)
- [20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25, pages 1097–1105. Curran Associates, Inc., 2012. [5](#)
- [21] Zhihao Li, Toshiyuki Motoyoshi, Kazuma Sasaki, Tetsuya Ogata, and Shigeki Sugano. Rethinking self-driving: Multi-task knowledge for better generalization and accident explanation ability. *arXiv preprint arXiv:1809.11100*, 2018. [1](#), [2](#), [3](#), [4](#), [5](#), [6](#), [7](#), [8](#)
- [22] Xiaodan Liang, Tairui Wang, Luona Yang, and Eric Xing. CIRL: Controllable imitative reinforcement learning for vision-based self-driving. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018. [2](#)
- [23] Shikun Liu, Edward Johns, and Andrew J Davison. End-to-end multi-task learning with attention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1871–1880, 2019. [2](#), [3](#)
- [24] K. Mori, H. Fukui, T. Murase, T. Hirakawa, T. Yamashita, and H. Fujiyoshi. Visual explanation by attention branch

- network for end-to-end learning-based self-driving. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 1577–1582, 2019. [1](#), [2](#)
- [25] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 9351 of *LNCS*, pages 234–241. Springer, 2015. (available on arXiv:1505.04597 [cs.CV]). [3](#)
- [26] Axel Sauer, Nikolay Savinov, and Andreas Geiger. Conditional affordance learning for driving in urban environments. In *Conference on Robot Learning (CoRL)*, 2018. [2](#), [4](#)
- [27] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. [6](#), [8](#)
- [28] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang. Residual attention network for image classification. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6450–6458, 2017. [2](#)
- [29] S. Woo, Jongchan Park, Joon-Young Lee, and In-So Kweon. Cbam: Convolutional block attention module. In *ECCV*, 2018. [2](#), [4](#)
- [30] Huazhe Xu, Yang Gao, Fisher Yu, and Trevor Darrell. End-to-end learning of driving models from large-scale video datasets. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2174–2182, 2017. [1](#), [2](#), [4](#)
- [31] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057. PMLR, 2015. [2](#)
- [32] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. [8](#)