# Advisable Learning for Self-driving Vehicles by Internalizing Observation-to-Action Rules

Jinkyu Kim, Suhong Moon, Anna Rohrbach, Trevor Darrell, and John Canny
EECS, University of California, Berkeley
{jinkyu.kim, suhong.moon, anna.rohrbach, trevordarrell, canny}@berkeley.edu

## Abstract

*Humans learn to drive through both practice and theory, e.g. by studying the rules, while most self-driving systems are limited to the former. Being able to incorporate human knowledge of typical causal driving behaviour should benefit autonomous systems. We propose a new approach that learns vehicle control with the help of human advice. Specifically, our system learns to summarize its visual observations in natural language, predict an appropriate action response (e.g. "I see a pedestrian crossing, so I stop"), and predict the controls, accordingly. Moreover, to enhance interpretability of our system, we introduce a fine-grained attention mechanism which relies on semantic segmentation and object-centric RoI pooling. We show that our approach of training the autonomous system with human advice, grounded in a rich semantic representation, matches or outperforms prior work in terms of control prediction and explanation generation. Our approach also results in more interpretable visual explanations by visualizing object-centric attention maps. Code is available at* https://github.com/JinkyuKimUCB/advisable-driving.

## 1. Introduction

Autonomous driving control has made dramatic progress in the last several years. The proposed vehicle controllers use a variety of approaches; recent efforts [5] suggest that deep neural networks can be effectively applied to the controllers in an end-to-end manner. These models, however, are known to be opaque. One way to simplify and expose the underlying reasoning, is via a situation-specific dependence on visible objects in the scene, i.e. by only attending to image areas that are causally linked to the driver's actions [15]. However, the resulting attention maps are not always compelling or human interpretable. Another option is to verbalize the autonomous vehicle's behaviour with natural language [17], Figure 2 (B). The resulting textual explanations are human understandable, but tend to be rather "shallow", as they report the more common objects over the less common ones, which may be more important (*e.g.* construction cones). Both approaches fall short of demonstrating causal behaviour akin to a typical human driver.

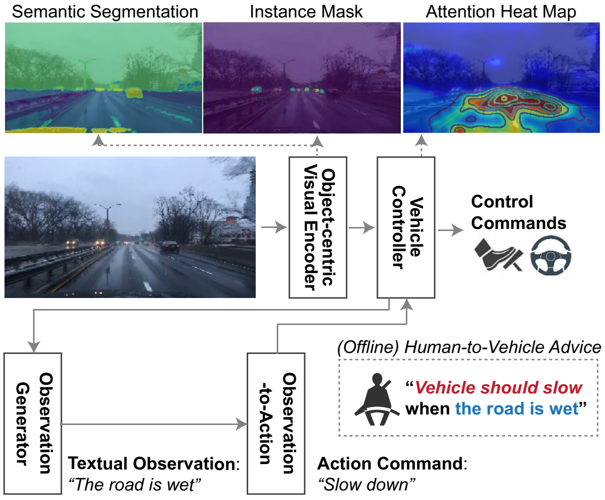
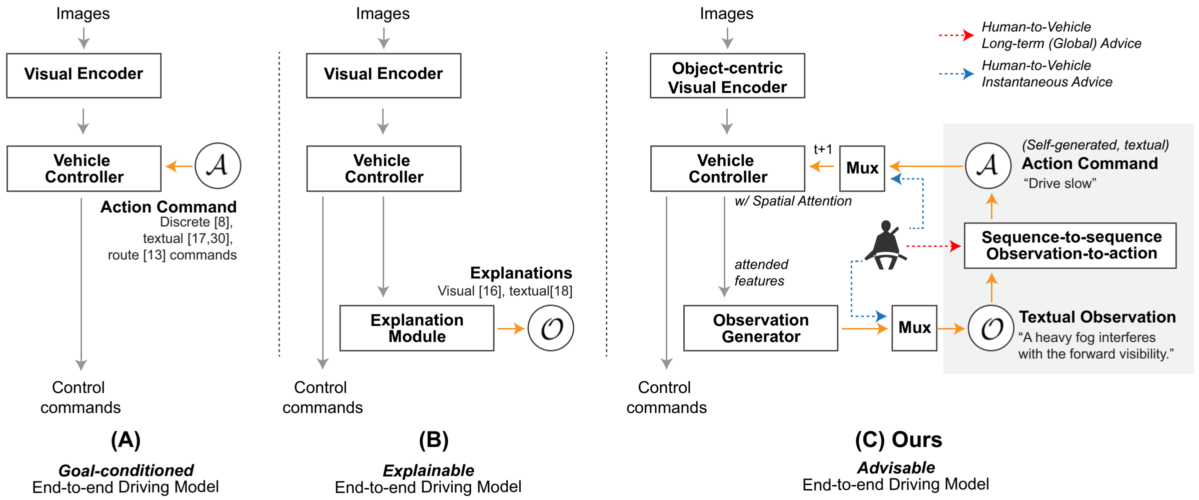

Semantic Segmentation
Instance Mask
Attention Heat Map
Object-centric Visual Encoder
Vehicle Controller
Control Commands
Observation Generator
Observation -to-Action
(Offline) Human-to-Vehicle Advice
"Vehicle should slow when the road is wet"
Textual Observation: "The road is wet"
Action Command: "Slow down"

Figure 1: Our model consists of four main parts: (1) an object-centric visual encoder built upon a semantic segmentation model, (2) an observation generator, which generates textual observation about the scenes ("The road is wet"), (3) an observation-to-action module, which maps a visual scene description to a (high-level) action command ("Slow down"), and (4) a vehicle controller conditioned on the generated action command.

To address this issue, [16] augment an imitation learning dataset with instantaneous human advice (*e.g.* "there is a pedestrian ahead", or "turn left"), see Figure 2 (A). They show that providing such inputs helps more closely imitate a human driver's behavior. While promising, this method requires ground-truth human inputs at test time.

Humans learn to drive not only from practice and demonstration, but also from theory, e.g. by studying the rules. We advocate for a more principled way of integrating human advice during learning. We assume that at training time, human advice is available in the form of observation-action



**(A)**
***Goal-conditioned***
End-to-end Driving Model

**(B)**
***Explainable***
End-to-end Driving Model

**(C) Ours**
***Advisable***
End-to-end Driving Model

Figure 2: (A) Existing goal-conditioned end-to-end driving models that takes (as an input) discrete [7], natural language commands [16, 28], and intended navigational route [12]. (B) Existing explainable end-to-end driving models that transduce DNN states to natural language [17] or visual explanations [15]. (C) Combining two above-mentioned ideas, we can create "Advisable" driving model that takes human-to-vehicle advice in the form of observation-action rules. To incorporate such rules, our model involves a Sequence-to-Sequence Observation-to-Action module, which generates a soft condition-action rule that maps a textual observation to a high-level action command. For details see Section 3.

rules (*e.g*. "if the road is wet, slow down"). Incorporating such rules could help driving models learn more human-like behavior, see Figure 1.

A key requirement of an advisable driving model is its explainability – exposing the controller's internal state is important for a user as an acknowledgement that the system is following advice. As mentioned earlier, visual attention is often used in recent explainable models [15, 17]. These models generate spatial attention maps, which are then displayed over the original images. However, such attention maps are coarse and have limited interpretability. They usually have a low spatial resolution (as the last convolutional layer) and are upsampled with a 2D Gaussian kernel. This blurs out the details and makes it difficult to determine what the model actually attends to. We advocate for using a richer representation, such as semantic segmentation, which provides pixel-wise prediction and delineates object boundaries in images. The output of the last convolutional layer retains information of the corresponding local image regions, which can be advantageous for obtaining more fine-grained attention maps. We thus propose to use semantic segmentation as our input representation. To further improve the quality of the attention maps, we also use an instance segmentation model, which allows us to distribute attention over individual objects.

Overall, we propose a novel self-driving model that is both advisable and explainable, see Figure 2 (C). Our model learns advice from human inputs which convey global rules that the user expects the vehicle to follow (*e.g*. "If a heavy fog interferes with your forward visibility, drive slowly"). We can also provide both visual explanations – by producing fine-grained attention maps, and textual explanations – by generating textual utterances (*e.g*. "the traffic light ahead turned red", thus "the car stopped"). We ground both functionalities in our object-centric visual representation.

We evaluate our approach on the BDD-X dataset [17] and show that our model matches or outperforms prior work in control prediction and textual observation generation. Our attention maps, tied to the semantic segmentation, result in object-centric (and thus more interpretable) visualization of internal states. Our human evaluation in a simulated environment (Carla [8]) further shows that our advisable system can increase user trust.

## 2. Related Work

**End-to-End Learning for Self-driving Vehicles.** Recent works [4, 12] suggest that a driving policy can be successfully learned by neural networks through supervised learning over observation (*e.g*. video) and action (*e.g*. steering) pairs, that are collected from human demonstration. Bojarski *et al*. [5] trained a 5-layer ConvNet to predict steering controls from a dashcam image, while Xu *et al*. [39] utilized a dilated ConvNet combined with an LSTM so as to predict vehicle's discretized future motions. Recently, Hecker *et al*. [12] explored the extended model that takes a surround-view multi-camera system, a route planner, and a CAN bus reader. Codevilla *et al*. [7] explored a conditional end-to-end driving model that takes high-level command input (*i.e*.

left-/right-turn, lane following, and intersection passing) at test time, see Figure 2 (A). To reduce the complexity, there is growing interest in end-to-mid [41] and mid-to-mid [4] driving models that produce a mid-level output representation in the form of a drivable trajectory by consuming either raw sensor or an intermediate scene representation as input. Their behavior, however, is opaque and learning to drive in urban areas remains challenging. These driving models are also known to be "black boxes" and thus lack of transparency may be a major drawback in self-driving applications where a high level of user trust is required to accept such a radical technology.

**Visual and Textual Explanations.** Explainability of deep neural networks has become a growing field in computer vision and machine learning communities [10]. In landmark work, [40] utilized deconvolution layers to visualize the internal representation of a ConvNet. Other approaches [42, 30] have explored synthesizing an image that highly activates a neuron. However, they lack formal measures of how the function estimated by the network is affected by spatially-extended features. Attention-based approaches may be exceptions to this rule. Kim *et al*. [15] utilized an attention model followed by additional salience filtering to show regions that causally affect the output. Wang *et al*. [36] and Wu *et al*. [38] introduced an instance-level attention model that finds objects (*e.g*., cars, pedestrians) that the network needs to pay attention to. However, such attention may be less convenient (especially in the driving domain) for users to "replay". It is also important to be able to justify the decisions that were made and explain why they are reasonable in a human understandable manner, *i.e*. in natural language [13, 14]. Kim *et al*. [17] proposed a textual explanation model to explain the rationales behind the vehicle controller, see Figure 2 (B). Explainable models can help reveal what the model is doing and show the basis for its decisions, which makes it easier to expose weaknesses and further improve. We propose a model that is both explainable and advisable. Human-to-vehicle advice can take a variety of forms, while natural language is an intuitive form of communication for humans. Our approach is inspired by [17], but we incorporate advice through learning to generate observations and corresponding actions in natural language.

**Advice-taking Models.** Recognition of the value of advice-taking has a long history in AI community [23], but few attempts have been made to exploit textual advice. Several approaches have been proposed to translate natural language advice to formal semantic representations, which are then used to bias actions for simulated soccer [19], mobile manipulation [24, 25, 31], and navigation [2]. Recent work suggests that incorporating natural language human feedback can improve text-based QA agents [21, 37] and image captioning performance [22]. Despite its potential, there are various challenges with collecting human feedback on the actions taken by self-driving cars (*e.g*. safety and liability). Other notable approaches (in the reinforcement learning setting) include the work by Tung *et al*. [32] that learns a visual reward detector conditioned on natural language action descriptions, which is then used to train agents. Kim *et al*. [16] introduced an approach to ground instantaneous human-to-vehicle advice w.r.t. perception and action and showed that accepting such advice improves overall control prediction accuracy, while Roh *et al*. [28] focused on conditioning natural language instructions to the driving model, see Figure 2 (A). Inspired by these work, we incorporate observation-action rules at training time, and learn to recognize when to follow advice at test time, rather than expecting such advice to be given by a "passenger" at test time.

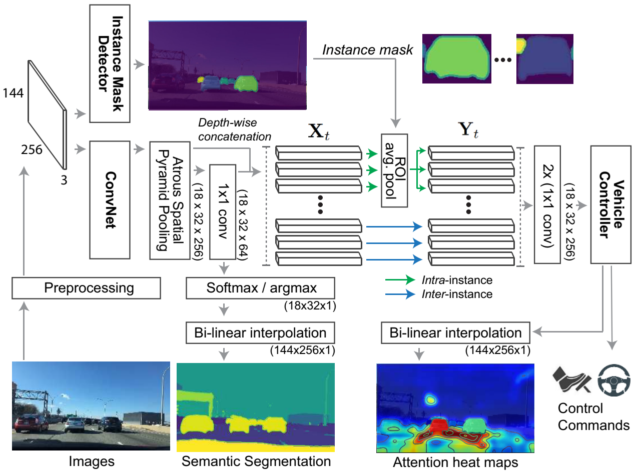


Figure 3: The detailed overview of our *Object-centric Visual Encoder* that is built upon an instance mask detector and a semantic segmentation model, both of which provide pixel-wise category predictions from images along with delineating the boundaries of object.

## 3. Advisable Learning

In this paper, we propose a novel driving model that is both explainable and advisable. Our model can provide the basis of its decision both by visualizing image regions that it attends to and by verbalizing the observations of what it sees (*e.g*. "it is snowing"). Our model is also advisable by incorporating general observation-action rules, which it is expected to follow.

As shown in Figure 2 (**C**), our model includes four main components. Our *Object-centric Visual Encoder* extracts visual (semantic) representations through a ConvNet that is pretrained on the task of semantic segmentation (Section 3.1). The *Vehicle Controller* is trained to predict control commands conditioned on the high-level action commands (*e.g*. "stop at the crosswalk") (Section 3.2). The *Ob-*

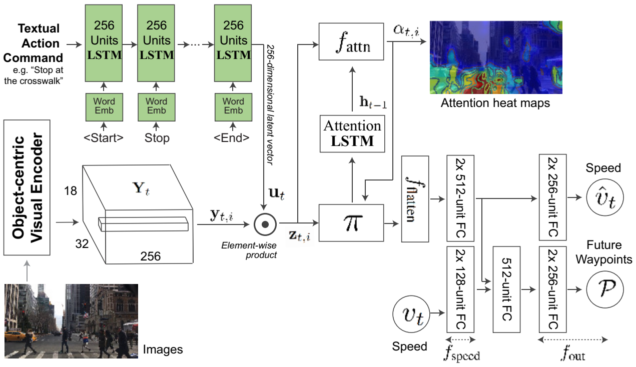


Figure 4: The detailed overview of our goal-conditioned *Vehicle Controller*. We take an action command in natural language as an input and ground it into the controller. Our model adopts spatial attention mechanism $\pi$, which guides where the controller looks. Conditioned on the attended feature and (optionally) the current speed $v_t$, our model outputs future trajectory $\mathcal{P}$ and speed $\hat{v}_t$.

*servation Generator* produces variable-length textual observations about the scenes (*e.g.* "pedestrians are waiting to cross") (Section 3.3). Finally, our *Sequence-to-Sequence Observation-to-Action* module generates soft condition-action rules that map visual scene descriptions (*e.g.*"*it is snowing*") to high-level action commands (*e.g.*"*maintain a slow speed*") (Section 3.4). Note that, our *Vehicle Controller* utilizes a visual (spatial) attention mechanism, which can highlight image regions the model fixates on for the network's output. This attended feature is then fed into the *Observation Generator* for the final prediction.

### 3.1. Object-centric Visual Encoder

We use images that are down-sampled to 10Hz and are resized to have dimensionality 144×256×3 by applying bilinear interpolation. Each image is normalized by subtracting the global mean from the raw pixels and dividing by the global standard deviation [29], see Figure 3.

**Segmentation as an Input Representation.** Instead of training a ConvNet from scratch, we use a semantic segmentation model that is pre-trained on the Mapillary Vistas street-view scene understanding dataset [26]. Our front-end vision module is therefore trained to recognize pixel-wise category predictions from images along with delineating the boundaries of each object. Here, we use the DeepLab v3 model [6], a state-of-the-art network that uses atrous spatial pyramid pooling to robustly segment objects at multiple scales with various filters of different sampling rates and fields-of-view. We obtain a high-level visual representation of an input image at each time step $t$. This representation $\mathbf{X}_t$ (of size 18×32×256) contains a set of 256-dimensional latent vectors over the spatial dimension, *i.e.* $\mathbf{X}_t = \{\mathbf{x}_{t,1}, \mathbf{x}_{t,2}, \ldots, \mathbf{x}_{t,l}\}$, where $l$ (= $w \times h$) is the spatial dimension. Note, that the use of semantic segmentation as the internal representation of visual scenes is generally transferable between real-world and simulated setting.

**Object-centric RoI Pooling.** To further provide object-centric attention heat maps, which highlight more precise object regions, we use an instance detection model, MaskR-CNN model [11], and tie the predicted instance masks to the feature $\mathbf{X}_t$. Given the instance regions (RoIs), a position-sensitive RoI pooling layer is used to aggregate the latent vectors $\mathbf{x}_{t,i}$ for $i = \{1, 2, \ldots, l\}$ for each RoI to obtain the visual feature $\mathbf{y}$. Note, that the pooled latent vector is then distributed equally to replace the original representations. This provides a subset of feature slices that share the same latent representation, and thus allows the model to equally attend to parts of RoI.

### 3.2. Goal-conditioned Vehicle Controller

**Grounding Natural Language Action Command.** Our vehicle controller is trained to predict control commands conditioned on the high-level action command (*e.g.* "maintains a slow speed"). We use a textual encoder that takes a variable-length textual command and grounds it into the vehicle controller. Following [16], we use an LSTM to encode an input word sequence and yield a 256-dimensional latent vector $\mathbf{u_t}$. We combine this vector with the visual feature $\mathbf{y}_{t,i}$ by an element-wise multiplication and obtain a feature vector $\mathbf{z}_{t,i} = \mathbf{y}_{t,i} \odot \mathbf{u_t}$ for $i = \{1, 2, \ldots, l\}$, which is then fed into visual attention module to generate attention maps. We provide detailed model architecture in Figure 4.

**Visual Attention.** Visual attention provides introspective (visual) explanations by filtering out non-salient image regions, while the attended regions have a potential causal effect on the output. The goal of visual attention mechanism is to find a context $\mathbf{C}_t = \{\mathbf{c}_{t,1}, \mathbf{c}_{t,2}, \ldots, \mathbf{c}_{t,l}\}$ by minimizing a loss function, where $\mathbf{c}_{t,i} = \pi(\alpha_{t,i}, \mathbf{z}_{t,i}) = \alpha_{t,i}\mathbf{z}_{t,i}$ for $i = \{1, 2, \ldots, l\}$. Note that a scalar attention weight value $\alpha_{t,i}$ is in $[0, 1]$ such that $\sum_i \alpha_{t,i} = 1$. We use a multi-layer perceptron to compute these attention weights, *i.e.* $\alpha_{t,i} = f_{\text{attn}}(\mathbf{z}_{t,i}, \mathbf{h}_{t-1})$, conditioned on the previous hidden state $\mathbf{h}_{t-1}$ (of the *Attention LSTM*), and the current advice-grounded feature vector $\mathbf{z}_{t,i}$. Softmax regression function is used to obtain the final normalized attention weight.

**Output.** Inspired by the prior work [4, 41], our vehicle controller predicts a future trajectory $\mathcal{P} = [p_{t,\Delta}, p_{t,2\Delta}, \ldots, p_{t,N\Delta}]$ along with speed $\hat{v}_t$. Each point $p_{t,j\Delta}$ for $j = \{1, 2, \ldots, N\}$ is characterized by its future longitudinal and latitudinal location after the time $j\Delta$. This trajectory can be converted into low-level driving control commands (*i.e.* steering, braking, and acceleration) by an optimizer within the constraints of the vehicle's dynamics. Different types of vehicles may utilize different control outputs to achieve the same driving trajectory, which argues against training a network to directly output low-level steering and acceleration control.

To predict the future trajectory, we use additional hidden layers $f_{\text{out}}$ conditioned on the latent representation $\mathbf{C}_t$ (from our *Advice-grounded Visual Attention*) and the current speed $v_t$, *i.e.* $\mathcal{P} = f_{\text{out}}([f_{\text{flatten}}(\mathbf{C}_t), f_{\text{speed}}(v_t)])$, where $f_{\text{speed}}$ denotes additional hidden layers to encode the speed in a high-dimensional latent space. $f_{\text{flatten}}$ is a flattening function. We use $\Delta$ as 0.5 seconds and $N$ as 6 (thus, we predict the future trajectory in the next 3 seconds).

**Loss Function.** We minimize the proportional control error (*i.e.* the difference between human-demonstrated and predicted) to train our future trajectory predictor.

$$\mathcal{L}_{ctl} = \frac{1}{NT}\sum_{t=1}^{T}\sum_{j=1}^{N}\lambda_j\|p_{t,j\Delta} - \hat{p}_{t,j\Delta}\|_2^2 + \lambda_0\|v_t - \hat{v}_t\|_2^2 \quad (1)$$

where $\lambda_j$ and $\lambda_0$ control the strength of each term, chosen to be inversely proportional to the global variance.

### 3.3. Textual Observation Generator

The main goal of our textual observation generator is to summarize visual observations, which need to be considered while driving, *e.g.* "there is a school bus with lights flashing" (this usually means the vehicle should pull over and remain stopped). Here, we use the term "observation" to convey the notion of the model's ability to actively perceive and register visual cues as being important for the vehicle controller. These observations can take a variety of forms with different levels of urgency and will be provided to the vehicle controller at every time step.

To generate such observations, our model involves a video-to-text module that takes a sequence of video frames and generates variable-length textual observations. In order to implement such a model, we start from the work of [17] that is originally designed to generate textual descriptions/explanations such as a pair "vehicle slows down" (description) and "because it is approaching an intersection and the light is red" (explanation). Unlike [17], where descriptions/explanations are predicted jointly as a single sequence (separated by a token), we focus on generating the later part (*i.e.* explanations) and treat them as observations. These observations are then used to predict the corresponding textual action commands, directing the vehicle to behave in a certain way (*e.g.* go, pass, turn), in Section 3.4.

We collect the latent vector $\bar{\mathbf{c}}_t$ over the past $T$ timesteps by summing over the attended feature vectors $\{\mathbf{c}_{t,i}\}$, *i.e.* $\bar{\mathbf{c}}_t = \sum_{i=1}^{l}\mathbf{c}_{t,i}$. We then apply a temporal attention mechanism with weights $\beta_{k,t}$ to those vectors at each time step $k$ (of sentence generation), *i.e.* $\mathbf{g}_k = \sum_{t=t_0-T+1}^{t_0}\beta_{k,t}\bar{\mathbf{c}}_t$ where $t_0$ is the current timestep and $\sum_t \beta_{k,t} = 1$ with $\beta_{k,t}$ is in $[0,1]$. The weight $\beta_{k,t}$ is computed by an attention model, which is similar to the spatial attention. This is common practice in sequence-to-sequence models and allows flexibility in output tokens relative to input samples [3].

Our decoder outputs per-word softmax probabilities. We minimize the following negative log-likelihood $\mathcal{L}_{obs} = -\sum_k \log p(\mathbf{o}_k|\mathbf{o}_{k-1}, \mathbf{g}_k)$.

### 3.4. Sequence-to-Sequence Observation-to-Action

We want our model to incorporate natural language human-to-vehicle advice. Such advice is typically high-level, rather than low-level (where the vehicle controller operates). Recent work [16] proposed a model that allows short-term (or local) textual advice from passengers (*e.g.* "there are construction cones" or "slow down"). More generally, advice might take the form of condition-action rules. In this work, we focus on such long-term (or global) advice from humans (*e.g.* driving instructors).

We use a general encoder-decoder framework to incorporate the observation-action rules. Our LSTM encoder takes a *generated* variable-length textual observation ("there is a sharp turn ahead") and yields a representative latent vector, while the decoder (another LSTM) outputs an action command sequence ("slow down"). The model is trained by minimizing the negative log-likelihood (similar to the observation generator). Our model is supervised by human inputs in the form of observation-action rules that the user expects the vehicle to follow. The predicted action commands are given as input to the vehicle controller. Note that such rules are learned during offline training separately from our vehicle controller and textual observation generator – we can learn such rules with different datasets. Our approach can also be applicable in an online setting by reinforcing the learning of our observation-action rules. A policy-gradient method can be used to train an agent to generate such rules in an online setting while estimating the reward signal by measuring automatic scores.

**Human-to-Vehicle Instantaneous Advice.** We currently assume that advice is given offline, rather than during online human-vehicle interaction. Note, however, that our model can also take instantaneous human-to-vehicle advice. As shown in Figure 2 (C), we use two multiplexers to accept observational and navigational advice. The observational advice is mapped to an action command by our model.

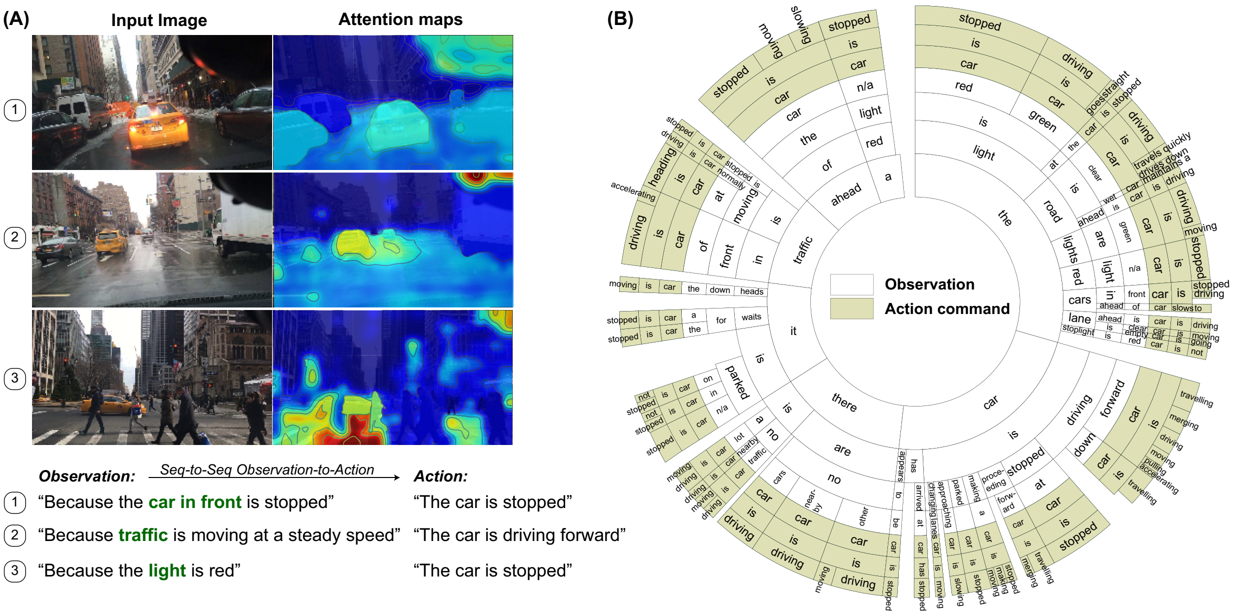
**Loss Function.** Our *Observation-to-Action* module outputs per-word softmax probabilities and we minimize the following negative log-likelihood $\mathcal{L}_{obs2act}$:

$$\mathcal{L}_{obs2act} = -\sum_m \log p(\mathbf{a}_m|\mathbf{a}_{m-1}, \{\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_K\}) \quad (2)$$

We minimize the following loss function $\mathcal{L}$ to train our entire driving model end-to-end, $\mathcal{L} = \mathcal{L}_{obs} + \mathcal{L}_{ctl} + \mathcal{L}_{obs2act}$.

## 4. Experiments

**Dataset.** We use the Berkeley DeepDrive-eXplanation (BDD-X) dataset [17] to train and evaluate our proposed



*Observation:* Seq-to-Seq Observation-to-Action → *Action:*

1 "Because the **car in front** is stopped" "The car is stopped"

2 "Because **traffic** is moving at a steady speed" "The car is driving forward"

3 "Because the **light** is red" "The car is stopped"

Figure 5: (A) Example observations and action commands generated by our model. We provide input raw images and attention maps of the vehicle controller. (B) The distribution of our top-100 generated observation/action pairs by their first four or three words, respectively. The ordering of the words starts from the center, and the length of the arc indicates the proportion of the number of words. Note that we remove areas where the number of words is too small to show.

Table 1: We report the vehicle control prediction performance for our approach and existing baselines. We compare the performance in terms of the median of average displacement errors (ADEs) as well as the 1st (Q1) and 3rd (Q3) quartiles (lower is better), *i.e.* Median [Q1, Q3].

| Model | ADE (in meters) ↓ | |
|---|---|---|
| | *without* speed inputs | *with* speed inputs |
| **A**. CNN+FC [5] | 2.36 [1.18, 4.61] | - |
| **B**. A + LSTM [39] | 3.29 [1.49, 6.93] | - |
| **C**. B + Attention [15] | 2.22 [1.17, 4.61] | - |
| **D**. A + Discrete commands (w/ branched output) [7] | 2.28 [0.89, 4.56] | 1.35 [0.66, 2.76] |
| **E**. C + (natural language) commands [16] | 2.11 [0.84, 4.86] | 1.35 [0.42, 2.94] |
| **F**. D + Long-term (global) Advice | 2.14 [0.93, 4.57] | 0.81 [0.45, 1.61] |
| **G**. F + Object-centric Visual Encoder (ours) | **1.93 [1.03, 4.26]** | **0.65 [0.46, 1.43]** |

model. BDD-X contains front-view dashcam videos (≈ 40 seconds) collected during urban driving in the United States, covering all the typical driving events. Alongside the video data, the dataset provides corresponding time-stamped IMU sensor measurements, which we use as a ground-truth control signal. We provide the dataset details in supplemental material.

Moreover, the dataset provides textual (i) descriptions of the vehicle's actions (*what* the driver is doing), and (ii) explanations for the driver's actions (*why* the driver took that action from the point of view of a driving instructor), such as the pair: "the car slows down" and "because it is approaching an intersection". This dataset is collected from human annotators in Amazon Mechanical Turk. We supervise our Textual Observation Generator with the textual explanations, while our Sequence-to-Sequence Observation-to-Action module is supervised with action descriptions (*i.e.* as navigational commands).

**Training and Evaluation Details.** Except for our object-centric visual encoder, we train other parts end-to-end using random initialization (*i.e.* no pre-trained weights). Unless otherwise stated, we use a single LSTM layer for all the components of our framework. For training, we use Adam optimization algorithm [18] and Xavier initialization [9]. For evaluation, we use the average displacement error (ADE) to quantitatively evaluate control prediction performance by comparing to ground-truth human-demonstrated control commands. To evaluate the textual utterances generated by our model, we use popular automatic metrics: BLEU [27], METEOR [20], CIDEr-D [34], and SPICE [1].

**Driving Performance Evaluation.** We report the vehicle control prediction performance for our model and a number of baselines to evaluate the ability to control a vehicle conditioned on the determined actions. We compare to end-to-end driving models, CNN+FC [5], CNN+FC+LSTM [39], and CNN+FC+LSTM+Attention [15] and goal-conditioned driving models that ground different types of goal: discrete commands [7], top-down view intended route [12], and natural language commands [16]. For a fair comparison, we use the same base CNN [7] in all cases except the model

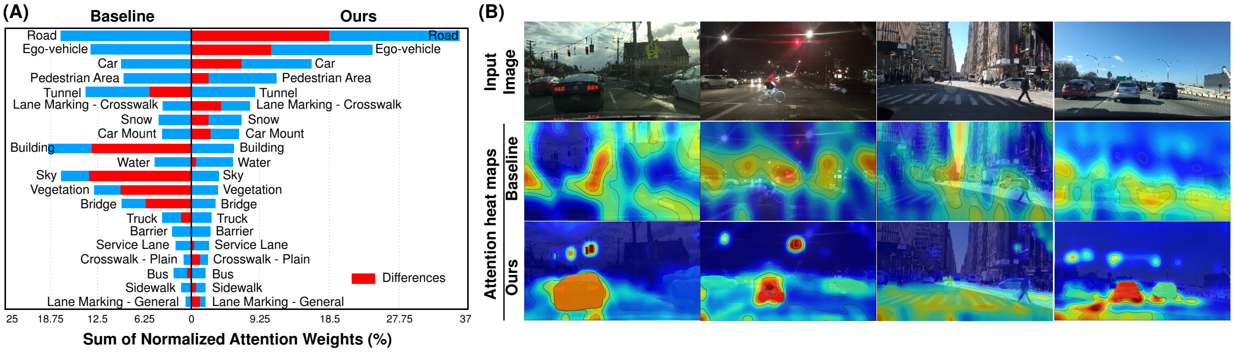


Figure 6: (A) The sum of normalized attention weights (blue) over the individual semantic regions for the baseline [15] and our model; differences shown in red. Our model attends more to road, car, pedestrian area, lane markings, and less to buildings, sky, vegetation. (We chose the top-20 most frequently attended regions of our model.) (B) We provide input images and compare attention maps from the baseline and our model. Attention maps are overlaid by their contour lines and shown over the input images. Higher value (red) of attention weight shows what the driving model attends to.

**G**, which uses our object-centric front-end visual encoder. All models have the same output layer and are trained by minimizing the same loss function.

We report performance of the aforementioned models in Table 1 (lower is better). Consistent with the prior work, goal-conditioned models [7, 16] (**D** and **E**) generally provide better control prediction performance against the non-goal-conditioned models (top three rows). Our model is built upon the model **D** – a goal-conditioned driving model that takes four different discrete navigational commands (i.e. lane following, turning, merging, parking). Based on this, our model **F** takes natural language commands for stimulus-driven action, e.g. the vehicle may make a stop, slow and/or deviate because of traffic participants, obstacles, any other environmental reasons. We observe that our model is further improved by adding long-term (or global) advising module (compare **F** vs. **D**). Our controller shares the attended feature with the Observation Generator, and thus encourages the model to attend to important visual cues (*e.g.* stop sign, traffic lights, pedestrians). Using our Object-centric Visual Encoder (instead of training a ConvNet from scratch) further improves control prediction performance (compare **G** vs. **F**).

**Analysis of Observation-to-Action Module.** In Figure 5 (A), we provide qualitative examples of the textual observations (*e.g.* "because the car in front is stopped") and corresponding high-level action commands ("the car is stopped") generated by our model. We also show the generated attention maps, which highlight image regions that have influenced the network's outputs (*i.e.* both textual observations and control commands). Our model attends to relevant visual cues and generates corresponding textual sequences. The vehicle controller also looks at other driving-related objects, *e.g.* lane markings. Importantly, our model is able to learn observation-action rules, which are provided by humans at training time, and correctly reflect typical links between visual causes and actions of human driving behavior.

Table 2: We report the quality of the generated textual observations (top) and action commands (bottom). We rely on standard automatic metrics: BLEU-4 [27], METEOR [20], CIDEr-D [34], and SPICE [1]. $^\dagger$: reported by [17]

| Model | Textual Observation Generation | | | |
|---|---|---|---|---|
| | BLEU-4 | METEOR | CIDEr-D | SPICE |
| S2VT [35]+SA+TA$^\dagger$ | 5.84 | 10.9 | 52.7 | 14.3 |
| S2VT+SA+TA+WAA [17]$^\dagger$ | 7.28 | 12.2 | 69.5 | 17.5 |
| Transformer-based Decoder [33] | 9.90 | 13.6 | 70.1 | 17.5 |
| Ours | **11.7** | **16.0** | **98.2** | **20.7** |
| Model | Textual Action Commands Generation | | | |
| S2VT [35]+SA+TA$^\dagger$ | 27.1 | 26.4 | 157.0 | 55.1 |
| S2VT+SA+TA+WAA [17]$^\dagger$ | 32.3 | 29.2 | 215.8 | 59.6 |
| Ours | **42.6** | **34.6** | **338.5** | **62.6** |

To see the distribution of the learned observation-to-action rules, we cluster observation/action pairs based on the first few words (*e.g.* the-light-is-red-car-is-stopped from the pair: "because the light is red" and "the car is stopped") as shown in Figure 5 (B). Our model generates a variety of observation-to-action pairs, which are compatible with the human driver's general knowledge. For example, the observation starts with "the road is wet" produces an action command starting with "the car maintains slow speed".

**Towards Semantically Rich Driving Model.** Analyzing the generated attention maps confirms that our model focuses more on important object-related visual cues (*e.g.* vehicles, lane markings, etc). In contrast, a baseline model [15] often attends to background (*e.g.* sky, trees, buildings, etc) but under-attends to important visual cues.

In Figure 6 (A), we provide the top 20 semantic segmentation labels where our model attends to. Blue bars represent the sum of normalized attention weights for each label. The top 3 attended regions for our model are *road, ego-vehicle, pedestrian area*, while the baseline focuses on *building, road, sky*. To see the difference between those models, we also visualize the differences as a red bar. Ours clearly focuses more on driving-related features, *e.g*. road, car, pedestrian area, lane markings, snow, and less on buildings, sky, vegetation, etc. In Figure 6 (B), we further compare the attention maps between ours and a baseline model [15]. We provide input video frames (1st row), attention maps generated by the baseline model (2nd row), and our attention maps (3rd row). Attention maps show that our model attends to important object-related visual cues (*e.g*. vehicles, lane markings, etc) with delineated object boundaries.

**Generated Observation/Action Quality.** Next we evaluate the quality of our generated observations and action commands, see Table 2 (higher is better). Our Textual Observation Generator predicts natural language observations based on the visual inputs. Some of our baselines are video captioning approaches, which do not take the vehicle control into account (S2VT [35]+SA (spatial attention)+TA (temporal attention) and Transformer-based approach [33]). At the same time, our full system is trained end-to-end, including the loss on the predicted controls, thus our textual observations are encouraged to be relevant to driving behavior. Therefore, we also compare to the best version of [17], the WAA model (weakly-aligned attention). This model generates action descriptions and explanations conditioned on predicted vehicle control, and we interpret the latter as observations. This is unlike our approach, where, conversely, vehicle control is predicted based on observations/action commands. Nevertheless, these are meaningful reference numbers for our approach. As we see, our model obtains the highest scores in all metrics both for generated observations and action commands.

**Simulation and Human Evaluation.** Explainable and advisable driving models can increase user trust by providing effective communication, which helps users convey their preferences/guidance to the vehicle and vice versa. To verify this, we run a human evaluation. We first migrate our driving model from the offline setting to a simulated environment, Carla [8], *i.e*. our model is trained on the BDD-X dataset and tested in the Carla simulator. We choose three different driving scenarios: (i) stopping at red lights, (ii) stopping at red lights in heavy rain, and (iii) stopping at a stop marking. In these experiments our driving model fails to stop for (ii) and (iii) scenarios. We then test the model with the following advice: "the light is red" and "there is a stop sign" for respective scenarios. We observe that the failure rate drops (see Figure 7 (A)). Further, we recruit 20 human judges and study the following three cases: (i) user only observes the car's behavior, (ii) user observes the model's behavior along with the attention and textual explanations, and (iii) user observes the model's behavior, attention, and textual explanations, *before and after* providing advice. As shown in Figure 7 (B), our explainable and advisable system shows better responses for user-trust. Specifically, providing visual and textual explanations slightly improves the user trust (blue vs. red). Further, showing users an example where the driving model accepts human-to-vehicle advice significantly improves the user-trust (red vs. yellow). In addition, we obtain feedback from the users by asking "To what level has the system improved with the human-to-vehicle advice?". Our evaluators acknowledge that advice improves the driving system, see Figure 7 (C). We provide the details of our evaluation in the Carla simulator in the supplemental material.

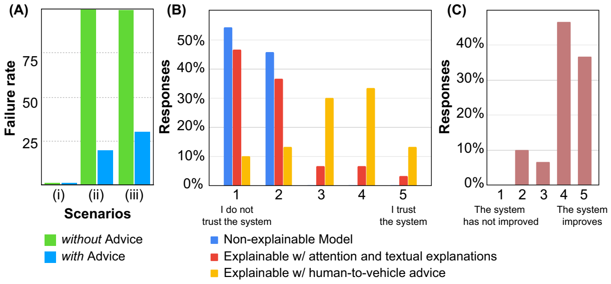


Figure 7: (A) We report the failure rate with and without advice inputs in the following three scenarios on a Carla simulator. (B-C) We also report the responses from our human study for the questions: (B) "How much do you trust this system?", and (C) "To what level has the system improved with the human-to-vehicle advice?". Answers were measured on a 1-5 Likert scale.

## 5. Conclusion

Towards learning more human-like driving behavior, we propose to use human advice in the form of observation-action rules. Specifically, we present a new approach where such advice is used as supervision during training and the controls are predicted based on the textual action commands. We rely on a semantic visual representation to better ground the textual observations and generate object-centric attention maps. Our experiments on the BDD-X dataset show that our model matches or outperforms prior work in control prediction and textual observation generation. Our human evaluation on the Carla simulator further shows that our advisable system can increase user trust.

**Acknowledgements.** We thank Y. Gao, D. Wang, O. Watkins, and C. Devin at UC Berkeley for their helpful discussion. This work was supported by DARPA XAI program and Berkeley DeepDrive. J. Kim was in part supported by Samsung Scholarship.

# References

[1] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. Spice: Semantic propositional image caption evaluation. In *ECCV*, pages 382–398. Springer, 2016. 6, 7

[2] Yoav Artzi and Luke Zettlemoyer. Weakly supervised learning of semantic parsers for mapping instructions to actions. *TACL*, 2013. 3

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *ICLR*, 2014. 5

[4] Mayank Bansal, Alex Krizhevsky, and Abhijit Ogale. Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst. *RSS*, 2019. 2, 3, 4

[5] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al. End to end learning for self-driving cars. *CoRR abs/1604.07316*, 2016. 1, 2, 6

[6] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *TPAMI*, 2018. 4

[7] Felipe Codevilla, Matthias Miiller, Antonio López, Vladlen Koltun, and Alexey Dosovitskiy. End-to-end driving via conditional imitation learning. In *ICRA*, pages 1–9. IEEE, 2018. 2, 6, 7

[8] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. *CoRL*, 2017. 2, 8

[9] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, 2010. 6

[10] David Gunning. Explainable artificial intelligence (xai). *Defense Advanced Research Projects Agency (DARPA)*, 2017. 3

[11] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pages 2961–2969, 2017. 4

[12] Simon Hecker, Dengxin Dai, and Luc Van Gool. End-to-end learning of driving models with surround-view cameras and route planners. In *ECCV*, 2018. 2, 6

[13] Lisa Anne Hendricks, Zeynep Akata, Marcus Rohrbach, Jeff Donahue, Bernt Schiele, and Trevor Darrell. Generating visual explanations. In *ECCV*, 2016. 3

[14] Lisa Anne Hendricks, Ronghang Hu, Trevor Darrell, and Zeynep Akata. Grounding visual explanations. In *ECCV*, 2018. 3

[15] Jinkyu Kim and John Canny. Interpretable learning for self-driving cars by visualizing causal attention. *ICCV*, 2017. 1, 2, 3, 6, 7, 8

[16] Jinkyu Kim, Terihusa Misu, Yi-Ting Chen, Ashish Tawari, and John Canny. Grounding human-to-vehicle advice for self-driving vehicles. *CVPR*, 2019. 1, 2, 3, 4, 5, 6, 7

[17] Jinkyu Kim, Anna Rohrbach, Trevor Darrell, John Canny, and Zeynep Akata. Textual explanations for self-driving vehicles. In *ECCV*, 2018. 1, 2, 3, 5, 7, 8

[18] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ICLR*, 2015. 6

[19] Gregory Kuhlmann, Peter Stone, Raymond Mooney, and Jude Shavlik. Guiding a reinforcement learner with natural language advice: Initial results in robocup soccer. In *AAAI Workshop*, 2004. 3

[20] Alon Lavie and Abhaya Agarwal. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *EMNLP*, 2005. 6, 7

[21] Jiwei Li, Alexander H Miller, Sumit Chopra, Marc'Aurelio Ranzato, and Jason Weston. Dialogue learning with human-in-the-loop. *arXiv preprint arXiv:1611.09823*, 2016. 3

[22] Huan Ling and Sanja Fidler. Teaching machines to describe images via natural language feedback. *arXiv preprint arXiv:1706.00130*, 2017. 3

[23] John McCarthy. *Programs with common sense*. RLE and MIT computation center, 1960. 3

[24] Dipendra K Misra, Jaeyong Sung, Kevin Lee, and Ashutosh Saxena. Tell me dave: Context-sensitive grounding of natural language to manipulation instructions. *IJRR*, 2016. 3

[25] Dipendra Kumar Misra, Kejia Tao, Percy Liang, and Ashutosh Saxena. Environment-driven lexicon induction for high-level instructions. In *ACL*, 2015. 3

[26] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulo, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *ICCV*, 2017. 4

[27] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 2002. 6, 7

[28] Junha Roh, Chris Paxton, Andrzej Pronobis, Ali Farhadi, and Dieter Fox. Conditional driving from natural language instructions. *CoRL*, 2019. 2, 3

[29] Samuel Rota Bulò, Lorenzo Porzi, and Peter Kontschieder. In-place activated batchnorm for memory-optimized training of dnns. In *CVPR*, 2018. 4

[30] 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 *ICCV*, pages 618–626, 2017. 3

[31] Stefanie Tellex, Thomas Kollar, Steven Dickerson, Matthew R Walter, Ashis Gopal Banerjee, Seth J Teller, and Nicholas Roy. Understanding natural language commands for robotic navigation and mobile manipulation. In *AAAI*, 2011. 3

[32] Hsiao-Yu Fish Tung, Adam W Harley, Liang-Kang Huang, and Katerina Fragkiadaki. Reward learning from narrated demonstrations. *CVPR*, 2018. 3

[33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 7, 8

[34] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *ICCV*, 2015. 6, 7

[35] Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko.

Sequence to sequence-video to text. In *ICCV*, pages 4534–4542, 2015. 7, 8

[36] Dequan Wang, Coline Devin, Qi-Zhi Cai, Fisher Yu, and Trevor Darrell. Deep object centric policies for autonomous driving. *ICRA*, 2019. 3

[37] Jason E Weston. Dialog-based language learning. In *NeurIPS*, 2016. 3

[38] Jialin Wu and Raymond J Mooney. Faithful multimodal explanation for visual question answering. *arXiv preprint arXiv:1809.02805*, 2018. 3

[39] Huazhe Xu, Yang Gao, Fisher Yu, and Trevor Darrell. End-to-end learning of driving models from large-scale video datasets. In *CVPR*, 2017. 2, 6

[40] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision*, pages 818–833. Springer, 2014. 3

[41] Wenyuan Zeng, Wenjie Luo, Simon Suo, Abbas Sadat, Bin Yang, Sergio Casas, and Raquel Urtasun. End-to-end interpretable neural motion planner. In *CVPR*, pages 8660–8669, 2019. 3, 4

[42] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *CVPR*, pages 2921–2929, 2016. 3