TVAnet: a spatial and feature-based attention model for self-driving car

Victor Flores-Benites
Universidad Catolica San Pablo
Arequipa, Peru
Email: victor.flores@ucsp.edu.pe

Carlos A. Mugruza-Vassallo
Universidad Nacional Tecnologica de Lima Sur
Lima, Peru
Email: carlos.mugruza@gmail.com

Rensso Mora-Colque
Universidad Catolica San Pablo
Arequipa, Peru
Email: rvhmora@ucsp.edu.pe

***Abstract*—End-to-end methods facilitate the development of self-driving models by employing a single network that learns the human driving style from examples. However, these models face problems of distributional shift problem, causal confusion, and high variance. To address these problems we propose two techniques. First, we propose the priority sampling algorithm, which biases the training sampling towards unknown observations for the model. Priority sampling employs a trade-off strategy that incentivizes the training algorithm to explore the whole dataset. Our results show uniform training on the dataset, as well as improved performance. As a second approach, we propose a model based on the theory of visual attention, called TVAnet, by which selecting relevant visual information to build an optimal environment representation. TVAnet employs two visual information selection mechanisms: spatial and feature-based attention. Spatial attention selects regions with visual encoding similar to contextual encoding, while feature-based attention selects features disentangled with useful information for routine driving. Furthermore, we encourage the model to recognize new sources of visual information by adding a bottom-up input. Results in the CoRL-2017 dataset show that our spatial attention mechanism recognizes regions relevant to the driving task. TVAnet builds disentangled features with low mutual dependence. Furthermore, our model is interpretable, facilitating the intelligent vehicle behavior. Finally, we report performance improvements over traditional end-to-end models.**

I. Introduction

End-to-end is an attractive solution for self-driving because it simplifies the development of driving models. Perception, prediction, and control are learned simultaneously without implicit programming of the human driving style, reducing development cost and time. However, these approaches are sensitive to distributional shift problems due to bias in the naturalistic driving datasets [1]. Moreover, end-to-end models have high variance, hence it is sensitive to initialization and sample order in training. We address these problems through the training algorithm and the driving model. Based on prioritized experience replay [2], we propose priority sampling which prioritizes model training on unknown samples (high loss). However, sampling based only on training loss prioritizes outliers in the database. We solve this problem by incentivizing the exploration in the database with UCT (Upper Confidence Bound 1 applied to trees) [3]. Our results show uniform training across the whole dataset, as well as a reduction in the error of the control signals.

Our main contribution is a model based on the theory of visual attention (TVA) [4], called TVAnet. The goal of our model is to build an optimal representation of the environment (state s_t) by selecting relevant visual information for the driving task. Two mechanisms are used for this purpose. First, we build a visual encoding z_t that abstracts the input visual information biased into relevant regions (spatial attention). Unlike self-attention [5], the selection of regions in TVAnet is based on similarity with a top-down encoding, which represents the driving context. However, this approach is sensitive to omitting information not encoded in the top-down encoding. To incentivize the exploration of new visual information sources, we add a bottom-up input to the visual attention mechanism. Then, the second mechanism (feature-based attention) selects the useful information present in the encoding z_t of the visual input. Previous works [6], [7] use feature attentional maps on the dimensions of visual coding, while our attentional maps are applied to the relevant directions in the feature space. Relevant directions are estimated through disentangling visual encoding z_t into interpretable features, which represent a unique visual knowledge. Finally, the vehicle state s_t is constructed from the selected features. Overall, our model recognizes the relevant and useful entities for driving. Also is an interpretable model, one can visualize the selected regions and features.

II. Related Work

End-to-end for self-driving car. End-to-end models [8]–[12] use a convolutional neuronal network (CNN) to encode the visual environment and a recursive network to encode the dynamics of the vehicle and environment. Furthermore, these models can follow navigation instructions via high-level control command input [8], [11], defined by a planning module. All inputs information is encoded and then decoded by the controller to compute control signals. However, this encoding is not optimal for complex environments, as it can introduce irrelevant information to the controller. Our proposed solution is a mechanism that filters out non-useful information according to the driving context. On the other hand, another important limitation of these methods is the shift distribution [1]. Most proposals address this problem employing data augmentation [8], [11]. It is also proposed to use database subsampling [11], [12], and oversampling [10] algorithms, adding biases over the training. We propose a priority sampling

algorithm for training, which overcomes the limitations of previous proposals by compensating for the introduced biases.

Visual attention for self-driving car. Models based on visual attention use a spatial saliency map [10], [13]–[16] to prioritize regions of the input image that contain information relevant to driving. It is notice these models are interpretable [10], [13], which facilitates error analysis and behavioral analysis. The proposed attention mechanisms are based on soft-attention [13], [15] or self-attention models [10], [16]. Also, attention mechanisms in spatial-temporal features [10] have been proposed. In these proposals, the attentional maps are guided by visual input, omitting the driving context. On the other hand, we propose that information filtering should be guided by top-down processes, i.e., attention mechanisms should enhance information relevant to the driving routine.

III. Theory of visual attention

TVA [4] defines visual attention as the bias competence of visual categories in the encoding in the visual short-term memory (VSTM). In this context, saying that the object x belongs to the visual category i is equivalent to saying that x has the feature i. The encoding rate $v(x,i)$ in the VSTM is defined as:

$$v(x,i) = \eta(x,i)\beta_i \frac{w_x}{\sum_{z \in S} w_z}, \tag{1}$$

where $\eta(x,i)$ is the force of sensory evidence that x has the feature i, β_i is the perceptual bias of the feature i, and w_x is the attention weight of the object x. Equation 1 suggests two attention mechanisms: object and category-based attention. Object-based attention is modulated by $w_x / \sum_{z \in S} w_z$, where the attentional weights are calculated by the weight equation:

$$w_x = \sum_{j \in R} \eta(x,j)\pi_j, \tag{2}$$

where π_j is the pertinence of category j, and R is the set of all visual categories. On the other hand, feature-based attention is modulated by a perceptual bias β_i associated with feature i. Bundesen et al. [17] proposed the hypothesis multiplicative interaction:

$$\beta_i = A p_i u_i, \tag{3}$$

where A is the level of alertness, p_i is the subjective prior probability of being presented with feature i, and u_i is the subjective importance of identifying feature i.

IV. An interpretation of TVA

TVA defines the membership of an object x to a category i as "object x has feature i", where a feature defines some property of object x such as color, shape, texture, etc. This information can be encoded by a CNN in a feature vector space. However, the TVA equations employ categorical membership values (scalar). We redefine category membership to make both concepts compatible. Let $\boldsymbol{z} \in \mathbb{R}^d$ be a visual encoding of the object x obtained by a deep neural network. Consider a decomposition $\boldsymbol{z} = \sum_i \boldsymbol{f}_i$, where $\{\boldsymbol{f}_i\}$ is a collection of interpretable features[1] with respect to the output signal. We set $\boldsymbol{f}_i = \hat{\boldsymbol{i}} z_i$, where $\hat{\boldsymbol{i}} = \boldsymbol{f}_i / \|\boldsymbol{f}_i\|$ is an unit or direction vector which defines the category (meaning), and $z_i = \|\boldsymbol{f}_i\|$ quantify the sensory force of the category i in the visual encoding $\boldsymbol{z}$. Thus:

$$\boldsymbol{z} = \sum_{i \in R} \hat{\boldsymbol{i}} z_i, \tag{4}$$

where R is a basis, and z_i quantifies the contribution of the i-th dimension. If the set R is a orthogonal base, each $\hat{\boldsymbol{i}}$ expresses a mutually exclusive knowledge about the object. This condition facilitates the compute the sensory strength of the category i in the object x:

$$\boldsymbol{z} \cdot \hat{\boldsymbol{i}} = z_i. \tag{5}$$

However, a general case may not admit a decomposition as shown in equation 4, since it is not always possible to obtain a basis whose vectors are interpretable. We suggest a decomposition as:

$$\boldsymbol{z} = \sum_{i \in R'} \hat{\boldsymbol{i}} z_i + \boldsymbol{H}, \tag{6}$$

where R' is a subset of R such that $\{\hat{\boldsymbol{i}}\}$ are linearly separable, and $\boldsymbol{H}$ is a non-interpretable vector. The vectors z_i are independent of $\boldsymbol{H}$, otherwise $\boldsymbol{H}$ can be a linear combination of z_i and therefore partially interpretable. However, the independence between categories i limits the hierarchical development of knowledge. Individual entities of semantic knowledge in humans are organized into higher-order conceptual categories that include elements with similar properties [19]. This organization is the support of inductive generalization in humans, which is a skill desired in artificial intelligence for out-of-distribution generalization [20]. We suggest that the building of representation with hierarchical semantic knowledge is desirable. However, these representations do not admit the decomposition given by equation 6. To address this problem, we propose to first construct a base representation x with hierarchical semantic knowledge. Then, this representation is transformed into a representation z that admits the decomposition given by equation 6. The advantage of this strategy is that we can construct several representations z_k from x so that each representation set admits a decomposition with different interpretable features of x.

Weight equation. The equation 2 estimates the attentional weight of an object from the pertinence of each category and the strength of the sensory evidence that the object belongs to those categories. Let $\boldsymbol{\eta}^x$ be a visual encoding of the object x, and $\boldsymbol{\pi}_i$ be a vector whose direction $\hat{\boldsymbol{i}}$ defines a category of interest, and whose magnitude π_i defines the relevance of that category. We estimate the sensory evidence of object x to category i by:

$$\boldsymbol{\eta}_i^x = proj_{\boldsymbol{\pi}_i} \boldsymbol{\eta}^x = \frac{\boldsymbol{\eta}^x \cdot \boldsymbol{\pi}_i}{\pi_i^2} \boldsymbol{\pi}_i = \frac{\boldsymbol{\eta}^x \cdot \boldsymbol{\pi}_i}{\pi_i} \hat{\boldsymbol{i}}. \tag{7}$$

[1]A representation z is interpretable if it contains information about the output signal y [18]. Therefore, in a fully interpretable representation $I(z,y) = H(z) = H(y)$, where $I(\bullet)$ is mutual information and $H(\bullet)$ is entropy.

Note that $(\boldsymbol{\eta}^x \cdot \boldsymbol{\pi}_i)/\pi_i = \|\boldsymbol{\eta}_i^x\|$. If x admits a decomposition given by equation 6, and i is a linearly separable feature, then $\boldsymbol{\eta}_i^x$ is the magnitude of the i-th dimension contribution, i.e. the sensory evidence of the category i in the object x. Then, $\eta(x,i) = \|\boldsymbol{\eta}_i^x\|$ in the equation 1. Therefore, we calculate the attentional weight of the object x by:

$$w_x = \sum_{i \in R} \eta(x,i)\pi_i = \sum_{i \in R} \|\boldsymbol{\eta}_i^x\| \, \|\boldsymbol{\pi}_i\|, \tag{8}$$

where R is the set of relevant features. From equation 7:

$$w_x = \sum_{i \in R} \frac{\boldsymbol{\eta}^x \cdot \boldsymbol{\pi}_i}{\pi_i}\pi_i = \boldsymbol{\eta}^x \cdot \sum_{i \in R} \boldsymbol{\pi}_i = \boldsymbol{\eta}^x \cdot \mathbf{F}, \tag{9}$$

where $\mathbf{F}$ contains the pertinence of all relevant features to the current task.

Perceptual bias. The multiplicative hypothesis of Bundesen et al. [17] suggests that the perceptual bias β_i associated with feature i is a product of the level of alertness A, the subjective prior probability p_i of being presented with feature i, and the subjective importance u_i of identifying i-th feature. Replacing in the rate equation we obtain:

$$v(x,i) = \eta(x,i)\beta_i\alpha_x = \eta(x,i)Ap_iu_i\alpha_x, \tag{10}$$

where $\alpha_x = w_x / \sum_{z \in S} w_z$, S is the set of all objects in the visual scene. We are interested in estimating the categories of objects selected by the weight equation. Then, the probability of categorizing an object x with the category i [21], given that the object x has been chosen by the equation 9:

$$p(i|x) = \frac{\eta(x,i)Ap_iu_ip_x}{\sum\limits_{j \in R} \eta(x,j)Ap_ju_jp_x} = \frac{p_i\eta(x,i)u_i}{\sum\limits_{j \in R} p_j\eta(x,j)u_j}, \tag{11}$$

where $\eta(x,j) = \left\|\boldsymbol{\eta}_j^x\right\|$. Let u_j be the magnitude of the vector $\boldsymbol{u}_j$ whose direction defines a useful category j. Also, we know that $\boldsymbol{\eta}_j^x = proj_{\boldsymbol{u}_j}\boldsymbol{\eta}^x$, then $\eta(x,i)u_i = \boldsymbol{\eta}^x \cdot \boldsymbol{u}_j$. Replacing in the equation 11:

$$p(i|x) = \frac{p_i\left[\boldsymbol{\eta}^x \cdot \boldsymbol{u}_i\right]}{\sum\limits_{j \in R} p_j\left[\boldsymbol{\eta}^x \cdot \boldsymbol{u}_j\right]}, \tag{12}$$

where $\boldsymbol{\eta}^x$ is the visual encoding of the object x, and $\boldsymbol{u}_i$ is a vector whose magnitude is the utility of category i.

V. Methods

Inspired by TVA, we propose a model that filters relevant visual information for the driving task in order to build an optimal state s_t. Our proposal is shown in Figure 1a, where a backbone based on convolutional neural networks is divided into two blocks: low encoder (LE) and high encoder (HE). We use equation 9 for spatial attention, assuming that spatial attention can be approximated to object-based attention by combining the spatial attention weights of objects in the environment. Note that this approximation depends on spatial discretization indeed a fine discretization will better approximate spatial attention towards object-based attention. We employ the spatial attention mechanism between LE and HE since the visual encoding output of LE has an adequate level of discretization (receptive field not large) and a good level of abstraction of visual information. On the other hand, we use equation 12 for feature-based attention, where we replace the subjective prior probability p_i with the probability that feature i is present in the current input. The feature-based attention mechanism is employed after HE since it has the highest level of visual abstraction. Note that our model is recursive (Figure 1a). The processing is non-selective in the first time step, i.e., the selection of regions is random. Then, the feature-based attention mechanism selects the relevant features for the current driving routine (defined by the control command c_t), in order to construct the vehicle state s_t. This selection of relevant features is used in the second time step (selective processing), where spatial regions are selected based on prior knowledge of features relevant to driving (top-down).

Non-selective processing. During the first time step, the relevant categories to the driving task are unknown to the agent, so it is not possible to use equation 9. Hence, the selection of spatial regions is randomized, i.e. the visual encoding is a non-selective processing. On the other hand, we compute the task-relevant features from the visual encoding $\boldsymbol{z}$, using the equation 12. This selection is employed to build the agent state and in the selection of spatial regions in the second stage (selective processing). For this purpose, we estimate $p(i|x)$ (equation 12) as a measure of feature relevance. Our model uses low-level and high-level encoders. The output of the high-level encoder is a visual encoding $\boldsymbol{z}$ without spatial dimensions, i.e. $\boldsymbol{z}$ encodes the whole receptive field of the input. Then, z_t contains the information of all the elements of the visual scene biased with the spatial attention mechanism:

$$\boldsymbol{z} \approx \sum_{l \in L} \alpha_l\boldsymbol{\eta}^l, \tag{13}$$

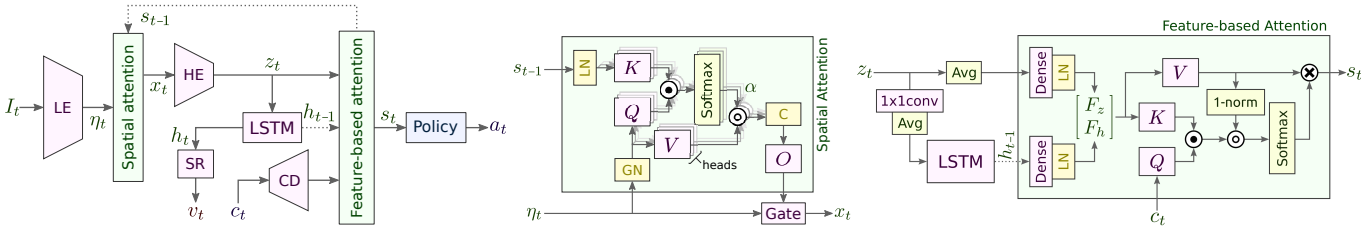
where α_l and $\boldsymbol{\eta}^l$ are the bias and sensory evidence at the spatial region l respectively. We estimate the subjective prior probability p_i of being presented with category i as:

$$p_i \approx \frac{\sum\limits_{l \in L} \alpha_l\eta(l,i)}{\sum\limits_{l \in L}\sum\limits_{j \in R} \alpha_l\eta(l,j)} = \frac{\left[\sum\limits_{l \in L} \alpha_l\boldsymbol{\eta}^l\right]\hat{\boldsymbol{i}}}{\left[\sum\limits_{l \in L} \alpha_l\boldsymbol{\eta}^l\right]\left[\sum\limits_{j \in R} \hat{\boldsymbol{j}}\right]} = \frac{\boldsymbol{z} \cdot \hat{\boldsymbol{i}}}{\boldsymbol{z} \cdot \hat{\boldsymbol{\omega}}} \tag{14}$$

where $\eta(l,j) = \boldsymbol{\eta}^l \cdot \hat{\boldsymbol{j}}$, $\boldsymbol{\eta}^l$ is the visual encoding in the spatial position l, and $\hat{\boldsymbol{\omega}} = \sum_{j \in R} \hat{\boldsymbol{j}}$. By replacing in equation 12:

$$p(i|z) = \frac{\frac{\boldsymbol{z} \cdot \hat{\boldsymbol{i}}}{\boldsymbol{z} \cdot \hat{\boldsymbol{\omega}}}\left[\boldsymbol{z} \cdot \boldsymbol{u}_i\right]}{\sum\limits_{j \in R} \frac{\boldsymbol{z} \cdot \hat{\boldsymbol{j}}}{\boldsymbol{z} \cdot \hat{\boldsymbol{\omega}}}\left[\boldsymbol{z} \cdot \boldsymbol{u}_j\right]} = \frac{\left[\boldsymbol{z} \cdot \hat{\boldsymbol{i}}\right]^2\left[\boldsymbol{u} \cdot \hat{\boldsymbol{i}}\right]}{\sum\limits_{j \in R}\left[\boldsymbol{z} \cdot \hat{\boldsymbol{j}}\right]^2\left[\boldsymbol{u} \cdot \hat{\boldsymbol{j}}\right]}, \tag{15}$$

where $\boldsymbol{z} \cdot \boldsymbol{u}_i = \boldsymbol{z} \cdot \left[\left(\boldsymbol{u} \cdot \hat{\boldsymbol{i}}\right)\hat{\boldsymbol{i}}\right] = \left[\boldsymbol{z} \cdot \hat{\boldsymbol{i}}\right]\left[\boldsymbol{u} \cdot \hat{\boldsymbol{i}}\right]$. Equation 15 shows that the probability of selecting category i grows quadratically with the magnitude of $\boldsymbol{z}$ in the i direction, while it grows linearly with the utility of feature i. As in equation 12, we can define the direction $\hat{\boldsymbol{i}}$ with $\boldsymbol{u}_i$.However, we propose that defining the direction with $\hat{\boldsymbol{i}} = \boldsymbol{z}_i / \|\boldsymbol{z}_i\|$ is better because



(a) Our proposal: TVAnet. (b) Spatial attention. (c) Feature-based attention.

Fig. 1. Our proposed network architecture. In the figure, LE: low-level encoder, HE: high-level encoder, SR: speed regularization network, CD: command decoder, LN: layer normalization, GN: group normalization, C: concatenate, $\odot$: dot product, $\circledcirc$: spatial element-wise product, $\otimes$: matrix multiplication.

the computation of $\boldsymbol{z}_i$ gives information about the relevant features contained in $\boldsymbol{z}$. Therefore, the process of computing $\boldsymbol{z}_i$ can be understood as disentangling the $\boldsymbol{z}$ representation. We define $\boldsymbol{F}_i = \boldsymbol{z}_i = \left[\boldsymbol{z} \cdot \hat{\boldsymbol{i}}\right] \hat{\boldsymbol{i}}$, then:

$$p(i|z) = \frac{F_i^2 \left[\boldsymbol{u} \cdot \dfrac{\boldsymbol{F}_i}{F_i}\right]}{\sum\limits_{j \in R} F_j^2 \left[\boldsymbol{u} \cdot \dfrac{\boldsymbol{F}_j}{F_j}\right]} = \frac{F_i \left[\boldsymbol{u} \cdot \boldsymbol{F}_i\right]}{\sum\limits_{j \in R} F_j \left[\boldsymbol{u} \cdot \boldsymbol{F}_j\right]}, \tag{16}$$

where $\boldsymbol{F}_i$ is a feature obtained from the encoding of the visual scene. Therefore, the probability that feature i is encoded in state s is equal to the normalization-by-sum of the similarity between $\boldsymbol{F}_i$ and u weighted by the magnitude of sensory evidence of $\boldsymbol{F}_i$. Finally, we compute the state as the expected value of feature $\boldsymbol{F}_i$ given $i \sim p(i|z)$:

$$s = \mathbb{E}_{i \sim p(i|z)} \left[\boldsymbol{F}_i\right] = \sum_{i \in R} \beta_i \boldsymbol{F}_i, \tag{17}$$

where $\beta_i = p(i|z)$. Note that β_i operates as a category bias.

Selective processing. The spatial selection mechanism described in equation 9 employs the feature selected in the first time step, where we set $\mathbf{F} = s$ because s contains the features used by the agent multiplied by an importance bias to the driving task (contextual encoding). We compute the attention map α at spatial position l as $\alpha_l = w_l / \sum_{i \in L} w_i$ The spatial attention map emphasizes regions with relevant features estimated in the previous time step, i.e., it is oriented from prior knowledge of the scene, intentions (selection), and driving goals. We call this mechanism top-down processing. Equation 9 assumes that $\boldsymbol{z}$ admits a decomposition given by equation 6, and the relevant features contained in the state s are linearly separable. However, the set of relevant features may have been related to each other, moreover, it would be difficult to find a space where every direction $\hat{\boldsymbol{i}}$ is independent of each other. For such a reason, our model build multiple attention maps α so that the resulting spaces admit the decomposition given by equation 6. Then, top-down encoding x^{TD} is equal to the combination of applying all the attention maps. However, a model guided only by this mechanism may have an important drawback: it would not be able to recognize a new relevant element in the scene that is different from the previously known elements. We solve this problem by introducing a bottom-up input, guided only by the sensory strength of the input. Hence, the complete attention mechanism is:

$$x_t = (1 - \varphi) x_t^{\text{BU}} + \varphi x_t^{\text{TD}}, \tag{18}$$

where φ is a gate that controls the effect of top-down and bottom-up processing, x_t^{BU} are features obtained by the bottom-up mechanism, x_t^{TD} are features obtained by the top-down mechanism, and x_t is the complete representation. Note that bottom-up is a mechanism that operates with raw sensory input, while top-down depends on superior perception processes, i.e. the agent has control over this process.

VI. Implementation

Our model receive two inputs: an RGB image $I_t \in \mathbb{R}^{a \times b \times 3}$ and a command control $c_t \in \mathbb{R}^4$, where a, b are the width and height. The command control inputs are encoded with one-hot encoding, and it represents four driving routines: "straight", "turn-left", "turn-right", or "follow lane". The command decoder network is a 16-neuron perceptron with ReLU activation function. The low and high encoder are the two halves of the ResNet-34 network.The first half has a visual encoding output $\eta_t \in \mathbb{R}^{H \times W \times d_1}$. While the second half has a visual encoding output $z_t \in \mathbb{R}^{d_2}$. The output of the spatial attention module is $x_t \in \mathbb{R}^{H \times W \times d_1}$. While the output of the features module is the state $s_t \in \mathbb{R}^{n \times d_s}$. On the other hand, the hidden state of LSTM is $h_t \in \mathbb{R}^{d_h}$. Approach of Xu et al. [22] was used to initialize the LSTM. Finally, our architecture has two outputs: the action $a_t \in \mathbb{R}^3$, and the velocity v_t (regularization method).

A. Spatial attention

The spatial attention module selects spatial regions with features relevant to the task. Our implementation is based on equation 9. We replace normalization-by-sum by Softmax, since this function builds sparse attention maps. Moreover, we use the notation of transformers [5]:

$$\alpha = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_1}}\right) \tag{19}$$

where $Q \in \mathbb{R}^{h \times L \times d_1}$ is the query, $K \in \mathbb{R}^{h \times n \times d_1}$ is the key, $\alpha \in \mathbb{R}^{h \times L \times n}$ is the spatial attention map, $L = H \times W$, h is the number of heads, and n is the number of tasks. At each head, property 6 is locally satisfied. We compute the query Q by 1×1-convolution network with input η_t. While we compute the key K by a perceptron that has the previous state as input s_{t-1}.

Additionally, we compute the value $V \in \mathbb{R}^{h \times L \times d_1}$ by an 1×1-convolution network with input η_t. Then, we calculate the top-down encoding with $x^{\text{TD}} = VW_O^{\text{SA}}$, where W_O^{SA} is a weight of the linear transformation. For simplicity, we make $x^{\text{BU}} = \eta_t$, since η_t encodes the sensory strength of the input features. The bottom-up and top-down encodings are combined through a GRU-type gating [23]. We employ layer normalization in the input s_{t-1}, and group normalization in the input η_t.

B. Feature-based attention

Our implementation is based on equation 16. We replace normalization-by-sum with Softmax, since this function constructs sparse attention maps.

$$\beta = \text{Softmax}\left(\frac{\|V\|_1}{d_s}\frac{QK^T}{\sqrt{d_s}}\right), \tag{20}$$

where $Q \in \mathbb{R}^{n \times d_s \times 1}$ is the query, $K \in \mathbb{R}^{n \times d_s \times m}$ is the key, $V \in \mathbb{R}^{n \times d_2 \times m}$ is the value, $\beta \in \mathbb{R}^{n \times m \times 1}$ is the feature attention map, d_s is the state depth, m is the number of features, and n is the number of tasks. We compute the key K by a perceptron with input $c_t \in \mathbb{R}^{16}$ which is the coding of the control command. On the other hand, we compute the query Q by a perceptron with input the concatenation $F = [F_z, F_h]$, where $F_z \in \mathbb{R}^{n \times d_s \times m/2}$ is calculate from the visual encoding $z_t \in \mathbb{R}^{d_2}$, while $F_h \in \mathbb{R}^{n \times d_s \times m/2}$ is compute from the previous hidden state $h_{t-1} \in \mathbb{R}^{d_h}$ of the LSTM. Note that z_t contains information about the content of the visual scene, and h_{t-1} contains information about the variations of z_t and the vehicle dynamics. Likewise, we compute the value V by a perceptron with input the concatenation $F = [F_z, F_h]$. Finally, we calculate (equation 17): $s = V\beta$, where $s \in \mathbb{R}^{n \times d_s}$ is the internal state of the agent, and n is the number of sub-tasks.

C. Policy

We have described two mechanisms of visual information selection that depend on a running task. However, self-driving is a complex task that requires the simultaneous execution of a set of subtasks [24]. We work with two main sub-tasks: steering angle control and velocity control. The first controls the direction changes with the steering angle output. While the second control the car speed changes with two outputs: throttle and brake. Our model assumes that each subtask requires a different type of information. Then, each sub-task has its sub-state, so that $s_t = [s^{st}, s^v]$, where s^{st} and s^v are the sub-states of steering angle and speed control respectively. For this reason, we propose that each task makes a selection of its relevant features using equation 16. Furthermore, each task selects spatial regions with relevant features using equation 9. We employ a dense block for each sub-task since it showed good results in locomotion tasks [25].

Loss function is a linear combination of action error predictions (steer a^{st}, throttle a^{th}, and brake a^{br}). We employ 1-norm as an error measure because its good correlation to driving performance [26]. Loss function for actions is defined as:

$$\mathcal{L}_a = \lambda_s \left\|a^{st} - \hat{a}^{st}\right\|_1 + \lambda_t \left\|a^{th} - \hat{a}^{th}\right\|_1 + \lambda_b \left\|a^{br} - \hat{a}^{br}\right\|_1,$$

where $\hat{a}^{st}$, $\hat{a}^{th}$ and $\hat{a}^{br}$ are the predict values of steer, throttle and brake respectively; while a^{st}, a^{th}, a^{br} are the real values. The coefficients λ_{st}, λ_{th}, λ_{br} weight the error of each action.

We employ a speed prediction regularization network for two purposes. The first is to orient the hidden state of the LSTM to it learn information about the driving dynamics [1]. The second function is to prevent gradient vanishing problem [27]. The loss function with regularization is $\mathcal{L} = \lambda_a \mathcal{L}_a + \lambda_v \mathcal{L}_v$, where $\mathcal{L}_v = \|v - \hat{v}\|_1$, v and $\hat{v}$ are the real and predict values of velocity respectively. The coefficients λ_a, λ_v weight the loss of action and regularization respectively.

VII. Training algorithm

We employ a non-uniformly sample method, based on prioritized experience replay [2]. This algorithm was proposed for deep Q-learning (DQN) to uniformize the training effectiveness in the whole replay buffer since in DQN common experiences are better learned (low TD-error), while rare experiences could be unknown to the agent (high TD-error). To solve this problem, the prioritized experience replay algorithm assigns a priority to each experience, so that unknown samples are more probable to be chosen for training.

We define the priority as $p_i = \mathcal{L}$ at i-th sample, where $\mathcal{L}$ is the training loss:The sampling probability is:

$$P_i = \frac{p_i^\gamma}{\sum_k p_k^\gamma} \tag{21}$$

where the parameter γ controls the relevance of the priority sampling. For $\gamma = 0$, the sampling is uniform. However, the algorithm might introduce a bias on outliers of the dataset. Thus, few unknown samples that are difficult to resolve may be selected many times due to its selection probability is high. As a result, the network may forget to solve simple problems that are common. To solve this problem of exploration-exploitation trade-off, we propose to use UCT, so that the probability of sampling the i-th sample at the j-th epoch:

$$\hat{P}_{i,j} = P_{i,j} + c\sqrt{\frac{\ln N_j}{n_{i,j}}}, \tag{22}$$

where $P_{i,j}$ is the sampling probability calculated with equation 21 (the priority is $p_i = \mathcal{L}_i$, where $\mathcal{L}_i$ is the loss value for the training), N_j is the total number of sample selections, $n_{i,j}$ is the number of times that the i-th sample was selected until the j-th epoch, and c is the exploration parameter, which theoretically is equal to $\sqrt{2}$ [3]. We normalize $\hat{P}_{i,j}$ such that $\sum_i \hat{P}_{i,j} = 1$. UCT introduces a compensation that stimulates exploration: the sampling probability decreases with the number of times the sample is selected.

However, priority sampling introduced a bias changing the solution to training convergence. To address this problem, Schaul et al. [2] uses importance-sampling (IS) weights:

$$w_i = \left(\frac{1}{N} \cdot \frac{1}{P(i)}\right)^\rho \tag{23}$$

where the parameter ρ controls the degree of compensation. Weights are scaling to the maximum value for stability reasons.

TABLE I
PARAMETERS SETTING USED IN OUR EXPERIMENTS.

Parameter	Value	Description
a, b	$96, 192$	RGB image shape
H, W	$12, 24$	Mid-level visual feature shape
d_1	128	Depth of low encoder
d_2	512	Depth of high encoder
d_s	64	Depth of state
d_h	1024	Depth of hidden state of LSTM
h	2	Number of heads in spatial attention
n	2	Number of tasks
m	32	Number of features
γ	1	Relevance of the priority sampling
ρ	1	Degree of IS compensation
c	$\sqrt{2} \approx 1.4$	Exploration parameter (UCT)
λ_{st}	0.45	Loss coefficient of steering angle error
λ_{th}	0.45	Loss coefficient of throttle error
λ_{br}	0.10	Loss coefficient of brake error
λ_s	0.05	Loss coefficient of speed regularization
λ_a	0.95	Loss coefficient of action error

TABLE II
PERFORMANCE COMPARISON.

Model	Loss		Control signals		
	Train	Eval.	Steer	Throttle	Brake
CIL [8]	0.0334	0.1294	0.0537	0.2079	0.1569
CILRS [1]	0.0317	0.1231	0.0542	0.1982	0.1519
SA [13]	0.0569	0.0918	0.0557	0.1499	0.1118
CIL [8] + PS	0.0343	0.0798	0.0474	0.1226	0.0691
CILRS [1] + PS	0.0293	0.0790	0.0438	0.1266	0.0610
SA [13] + PS	0.0396	0.0758	0.0440	0.1187	**0.0587**
TVAnet (T-$\|V\|$)	0.0263	**0.0722**	0.0364	0.1157	0.0647
TVAnet (T)	0.0288	0.0777	0.0334	0.1045	0.0594
TVAnet (D+$\|F\|$)	**0.0224**	0.0725	0.0332	**0.0932**	0.0593
TVAnet (D+$\|V\|$)	0.0227	0.0731	**0.0290**	0.0957	0.0618

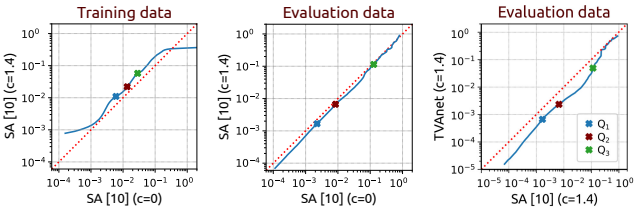
VIII. RESULTS

We evaluate our proposal on the CoRL2017 dataset [8]. The network and training parameters are defined in Table I. We employ Xavier initialization, and the optimizer Adam with parameters $\beta_1 = 0.7$ and $\beta_2 = 0.85$ based on [8]. Empirically we set learning rate at 0.0001, which decays to one-tenth after 80 epochs. The parameters λ in Table I are based on [1].

A. Performance

The performance results are summarized in Table II, where our proposal is compared with three self-driving models (CIL [8], CILRS [1], and Kim et al. [13]). Note that the loss function of our approach includes speed regularization. Our proposal outperforms the proposals studied, since it obtains a lower error in the control signals. An ablation study is presented:

Priority sampling. Table II shows the results of comparing the three networks with and without priority sampling (PS). CIL and CILRS are models that employ independent policies for each control command. On the other hand, the proposal of Kim et al. is based on soft-attention (SA), which estimates a spatial attention map. The three networks improve when trained with priority sampling: the error of the control signals is reduced. Morever, we show that the SA model obtains a lower evaluation loss than CIL and CILRS.



(a) Effect of UTC on training loss (b) Effect of UTC on validation loss. (c) Comparison of SA [13] and TVAnet.

Fig. 2. Percentile curves, plotted from the percentile values of the abscissa and the ordinate. In the figure, Q_1, Q_2, and Q_3 are the 1st, 2nd, and 3rd quartile respectively. From left to right decreases the model's knowledge about the data.

UCT compensation. Figure 2a shows a comparison of the percentile curve of the training loss of SA when not using and using UCT. The training loss of the known samples is lower when using UCT, however is higher with the weakly known samples. This result suggests that the model trained without UCT overfits on a subset of data and forgets how to deal with unfamiliar samples, while priority sampling with UCT uniformly learns the experiences of the database. Figure 2b shows the result for the validation loss, where priority sampling with UCT presents a better generalization. Finally, in Figure 2c we compare the percentile curve of the validation loss when using SA and when using our model. The results show that TVAnet outperforms SA in most cases. Note that Figure 2c does not present the S-shape of Figure 2a, so the training is uniform in both models.

Feature attention equation. Equation 20 differs from the self-attention equation[2] by the $\|V\|_1 / d_s$ term. We evaluated not using the scaling term (T-$\|V\|$ in the Table II). The results show that the evaluation loss is lower when using the self-attention equation. However, this reduction is due to a smaller error in the regularization. Our proposal obtains a lower error in the control signals.

Features disentangled. We evaluate the benefit of using disentangled features (D in the Table II) in the equation 20. Our proposal with tangled features (T in the Table II) omits the perceptron network of Figure 1c. Then, we compute Q_z, K_z, V_z from z_t, and Q_h, K_h, V_h from h_{t-1}. So, query is $Q = [Q_z, Q_h]$, key is $K = [K_z, K_h]$, and value is $V = [V_z, V_h]$. The results suggest that disentangling features before the feature attention mechanism improves performance. We noticed improvements in steering angle and throttle, however the brake error increases by 0.0001.

Magnitude of sensory evidence in the feature attention equation. We evaluated the definition of the magnitude of sensory evidence of feature (F_i in equation 16) . In Figure 1c we define the magnitude of sensory evidence of feature as the 1-norm of V. However, it can also be defined as the 1-norm of the magnitude of the disentangled feature $F = [F_z, F_h]$. The results in the Table II show a slight improvement in evaluation loss when using the 1-norm of the untangled feature.

[2]The self-attention equation is Softmax $\left(QK^T/\sqrt{d}\right)$ [5].

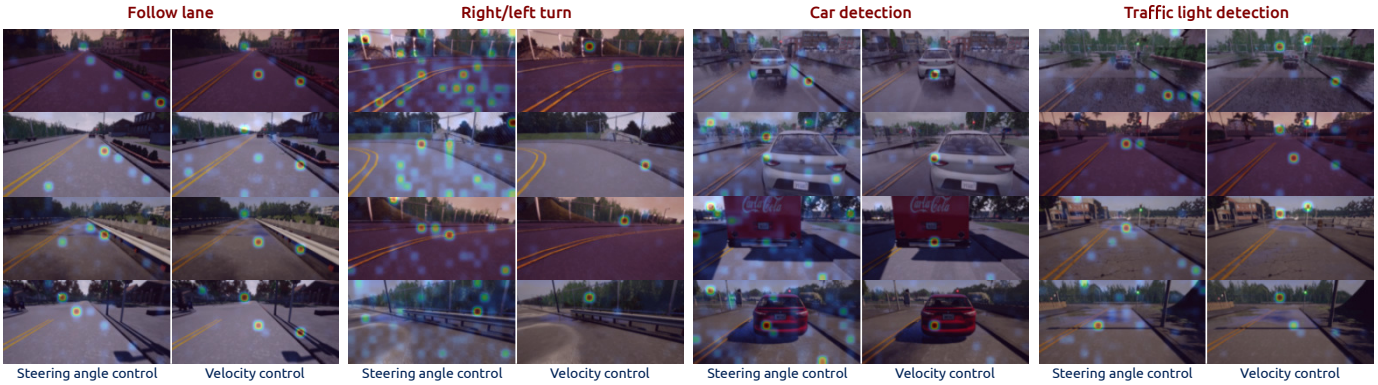
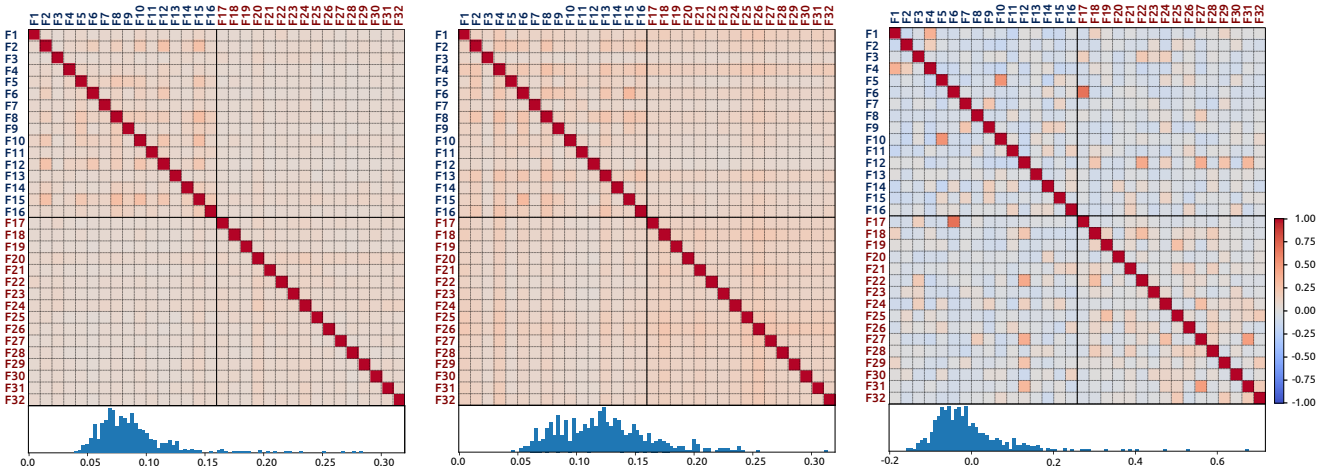


Fig. 3. Spatial attention maps obtained in three driving routines (follow lane, right and left turn) and in the detection of cars and traffic lights.



(a) Uncertainty coefficient between activations of the features attention map (**T1**).

(b) Uncertainty coefficient between activations of the features attention map (**T2**).

(c) Cosine distance between the disentangled features.

Fig. 4. Analysis of the attention maps and magnitudes of the features. We evaluate the disentangled features with the cosine distance (similarity measure). In addition, we evaluate the activation of feature attention maps with uncertainty coefficient.

B. Spatial attention

We show some examples of spatial attention maps in the Figure 3. The images are obtained by combining the attention maps of the two heads of each sub-task: steering angle control (T1) and velocity control (T2). Note that the attention maps are mostly sparse. The attention maps in the follow lane and right/left turn routines focus on the road line and sidewalk edge. However, our model has preferences in specific regions of the input image, so the attention mechanism could have difficulties in recognizing road line or sidewalk edge in the regions where it is not usually located. We note that velocity control has a preference in the spatial region corresponding to the road horizon, even when it is not visible (right/left turn). On the other hand, the steering angle control of our model does not focus on a specific region when performing the right/left turn routines. Nonetheless, on some occasions, it focuses on the road line.

We evaluate our proposal in critical driving situations. We notice that the attention maps stay static when the car is stopped. When the car is stopping, the attention maps focus on the critical element (front car front or traffic light). However, we notice that the agent focuses on the same spatial region (image center) when detecting cars. The behavior is similar when detecting traffic lights, where it focuses on the upper right region of the image. Nevertheless, the agent does not always focus on the traffic lights (the last row of Figure 3). We suggest that the dataset could be responsible for these errors since it contains sequences that omit the traffic light. However, this hypothesis could not be proven in this study.

C. Feature attention

In the Figure 4, we analyze the disentangled features and the feature attention maps. The features are sorted from the mode of feature magnitude into two groups: the first group corresponds to the features obtained from the visual encoding z_t (F1-F16), and the second group corresponds to the features obtained from the hidden state h_{t-1} of the LSTM (F17-F32). Figure 4a and 4b show the uncertainty coefficients between activations of the features attention map

for each sub-task: steering angle control (T1) and velocity control (T2). The uncertainty coefficient is computed with $U(x|y) = I(x,y)/H(x)$, where $I(x,y)$ is the mutual information between x and y, and $H(x)$ is the entropy of x. We also show the histogram of the uncertainty coefficients. The results suggest a low dependence on the activation of the feature attention map. The dependence is higher in the attention map of T2, and in the attention map values obtained from the visual encoding z_t in T1.

Finally, we evaluate the similarity between the disentangled features. Note that in the section IV we suggested that the results of decomposing z_t should be orthogonal features. We evaluate this condition by means of the cosine distance in Figure 4c. We also show the histogram of the cosine distance values. The results show that the cosine distance is non-zero, but low (in range $[-0.2, 0.2]$) in most cases. There are cases where the cosine distance is high, which introduces biases in the feature attention mechanism.

Limitations. The present work has evaluated our model and training method on the CoRL2017 dataset [8]. As discussed in Section VIII-B, the vehicle-traffic light interaction has not been properly studied. In future work, we will evaluate our approach on CARLA100 dataset [1], where the elements of the driving environment play an active role.

IX. Conclusion

We propose a training algorithm and a self-driving model based on TVA. We showed that (i) Priority sampling reduced the error of the control signals in three networks: CIL [8], CILRS [1], and Kim et al. [13]. (ii) Training with priority sampling is uniform throughout the database and improves the generalization of the model. (iii) The spatial attention maps of our model help to understand the vehicle behavior. However, our proposal has a spatial bias which could affect the performance when changing the environment. (iv) Our model built disentangled features with low similarity (cosine distance) and the feature attention map has low activation dependence. However, few disentangled features have high similarities, which introduces a bias in feature selection. On the other hand, our self-driving model showed performance improvements when compared to the three networks studied.

References

[1] F. Codevilla, E. Santana, A. M. López, and A. Gaidon, "Exploring the limitations of behavior cloning for autonomous driving," *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9328–9337, 2019.

[2] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized experience replay," *CoRR*, vol. abs/1511.05952, 2016.

[3] L. Kocsis and C. Szepesvári, "Bandit based monte-carlo planning," in *European conference on machine learning*. Springer, 2006, pp. 282–293.

[4] C. Bundesen, S. Vangkilde, and A. Petersen, "Recent developments in a computational theory of visual attention (TVA)," *Vision research*, vol. 116, pp. 210–218, 2015.

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[6] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[7] I. Bello, B. Zoph, A. Vaswani, J. Shlens, and Q. V. Le, "Attention augmented convolutional networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 3286–3295.

[8] F. Codevilla, M. Miiller, A. López, V. Koltun, and A. Dosovitskiy, "End-to-end driving via conditional imitation learning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1–9.

[9] D. Wang, J. Wen, Y. Wang, X. Huang, and F. Pei, "End-to-end self-driving using deep neural networks with multi-auxiliary tasks," *Automotive Innovation*, pp. 1–10, 2019.

[10] Y.-C. Liu, Y.-A. Hsieh, M.-H. Chen, C.-H. H. Yang, J. Tegner, and Y.-C. J. Tsai, "Interpretable self-attention temporal reasoning for driving behavior understanding," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 2338–2342.

[11] Z. Huang, C. Lv, Y. Xing, and J. Wu, "Multi-modal sensor fusion-based deep neural network for end-to-end autonomous driving with scene understanding," *IEEE Sensors Journal*, 2020.

[12] K. Ishihara, A. Kanervisto, J. Miura, and V. Hautamäki, "Multi-task learning with attention for end-to-end autonomous driving," *arXiv preprint arXiv:2104.10753*, 2021.

[13] J. Kim and J. Canny, "Interpretable learning for self-driving cars by visualizing causal attention," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2942–2950.

[14] J. Kim, T. Misu, Y.-T. Chen, A. Tawari, and J. Canny, "Grounding human-to-vehicle advice for self-driving vehicles," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 591–10 599.

[15] 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)*. IEEE, 2019, pp. 1577–1582.

[16] L. Li, B. Yang, M. Liang, W. Zeng, M. Ren, S. Segal, and R. Urtasun, "End-to-end contextual perception and prediction with interaction transformer," *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5784–5791, 2020.

[17] C. Bundesen, S. Vangkilde, and T. Habekost, "Components of visual bias: a multiplicative hypothesis," *Annals of the New York Academy of Sciences*, vol. 1339, no. 1, pp. 116–124, 2015.

[18] K. Do and T. Tran, "Theory and evaluation metrics for learning disentangled representations," in *International Conference on Learning Representations*, 2020.

[19] A. M. Saxe, J. L. McClelland, and S. Ganguli, "A mathematical theory of semantic development in deep neural networks," *Proceedings of the National Academy of Sciences*, vol. 116, no. 23, pp. 11 537–11 546, 2019.

[20] B. Schölkopf, F. Locatello, S. Bauer, N. R. Ke, N. Kalchbrenner, A. Goyal, and Y. Bengio, "Toward causal representation learning," *Proceedings of the IEEE*, vol. 109, no. 5, pp. 612–634, 2021.

[21] G. D. Logan, "An instance theory of attention and memory." *Psychological review*, vol. 109, no. 2, p. 376, 2002.

[22] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *Proceedings of the 32nd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, F. Bach and D. Blei, Eds., vol. 37. Lille, France: PMLR, 07–09 Jul 2015, pp. 2048–2057.

[23] E. Parisotto, F. Song, J. Rae, R. Pascanu, C. Gulcehre, S. Jayakumar, M. Jaderberg, R. L. Kaufman, A. Clark, S. Noury *et al.*, "Stabilizing transformers for reinforcement learning," in *International Conference on Machine Learning*. PMLR, 2020, pp. 7487–7498.

[24] J. Aasman and J. A. Michon, "Multitasking in driving," *Soar: A cognitive architecture in perspective*, pp. 169–198, 1992.

[25] S. Sinha, H. Bharadhwaj, A. Srinivas, and A. Garg, "D2RL: Deep dense architectures in reinforcement learning," *arXiv preprint arXiv:2010.09163*, 2020.

[26] F. Codevilla, A. M. Lopez, V. Koltun, and A. Dosovitskiy, "On offline evaluation of vision-based driving models," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 236–251.

[27] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.