Attentional Bottleneck: Towards an Interpretable Deep Driving Network

Jinkyu Kim and Mayank Bansal
Waymo Research
{jinkyukim, mayban}@waymo.com

Abstract

Deep neural networks are a key component of behavior prediction and motion generation for self-driving cars. One of their main drawbacks is a lack of transparency: they should provide easy to interpret rationales for what triggers certain behaviors. We propose an architecture called Attentional Bottleneck with the goal of improving transparency. Our key idea is to combine visual attention, which identifies what aspects of the input the model is using, with an information bottleneck that enables the model to only use aspects of the input which are important. This not only provides sparse and interpretable attention maps (e.g. focusing only on specific vehicles in the scene), but it adds this transparency at no cost to model accuracy. In fact, we find slight improvements in accuracy when applying Attentional Bottleneck to the ChauffeurNet model in comparison to a traditional visual attention model that degrades accuracy.

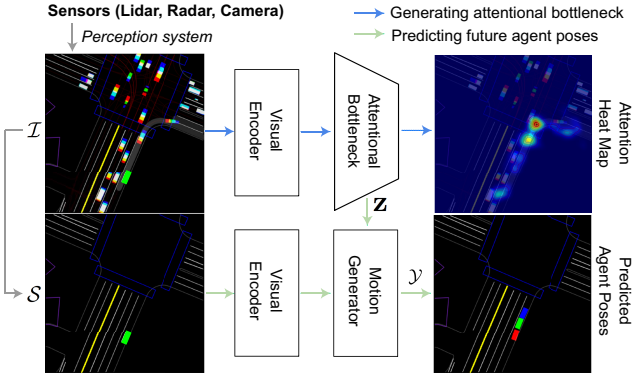

Figure 1. An overview of our interpretable driving model. Our model takes a top-down input representation $\mathcal{I}$ and outputs the future agent poses $\mathcal{Y}$ along with an attention map. An *Attentional Bottleneck* encodes the inputs $\mathcal{I}$ to a latent vector $\mathbf{z}$ while also producing an interpretable attention heat map. The motion generator operates in a partially observable environment using only the dense scene context $\mathcal{S} \subset \mathcal{I}$ along with $\mathbf{z}$ to predict poses $\mathcal{Y}$.

1. Introduction

Deep neural networks are powerful function estimators and have been a key component in self-driving software systems [2, 13]. Such networks are, however, notoriously cryptic – their hidden layer activations may have no obvious relation to the function being estimated by the network. Explainability of deep neural networks has thus seen growing interest in computer vision and machine learning [5, 11], with visual attention-based approaches like Kim *et al*. [7] being specifically applied to autonomous deep driving networks. Visual attention finds spatially varying scalar attention weights $\alpha(x, y) \in [0, 1]$ typically by learning a multilayer perceptron from a set of input features $\mathbf{F} = \{\mathbf{f}(x, y)\}$. Attended features $\mathbf{A} = \{\mathbf{a}(x, y)\}$ obtained as $\mathbf{a}(x, y) = \alpha(x, y)\mathbf{f}(x, y)$ are then used by the model instead of the original features $\mathbf{F}$. The model is trained end-to-end, leading the attention weights to link the network's output to its input – visualizing the weights as a 2D heatmap thus provides insight into the areas of the input image that the network attends to. Furthermore, to be easily interpretable, attention needs to be sparse (i.e. low entropy), while ideally also enhancing the performance of the original model. Unfortunately, given the complexity of the driving task, we find that a straightforward integration of attention maps tends to find all potentially salient image areas, resulting in limited interpretability (e.g. Figure 3).

In this work, we manage to achieve sparse and salient attention maps and good final model performance, by attaching attention to a bottlenecked latent representation of the input. However, given the information loss in the bottleneck, we need to provide the model direct access to a subset of dense inputs (e.g. road lane geometry and connectivity information) that are harder to compress. This frees up the bottleneck branch to focus on selecting the most relevant parts of the dynamic input (e.g. nearby objects), while retaining the model performance.

End-to-end driving models that directly process a camera image as input have several scene elements confounded into nearby pixels thus making a separation into dense and sparse input subsets infeasible. Therefore, we focus on improving the interpretability of a driving model that uses a mid-level input representation. This means that instead of directly using low-level sensor data, the model uses higher-level semantic information like objects detected by a per-

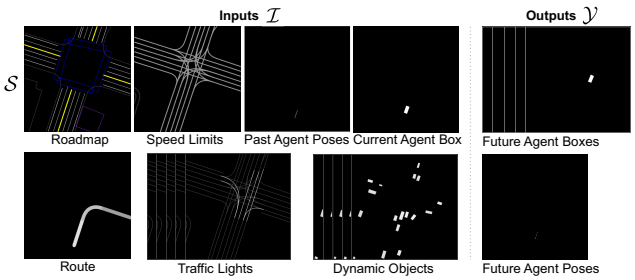

Figure 2. Top-down rendered inputs $\mathcal{I}$ (left) and outputs $\mathcal{Y}$ (right) for the *ChauffeurNet* model. The subset of dense scene context inputs $\mathcal{S}$ are shown in the top row.

ception system. As a proxy for such a network, we work with the recently published *ChauffeurNet* [2] model, although the ideas presented are more generally applicable.

To generate sparser and more interpretable attention maps, we propose an architecture called *Attentional Bottleneck* (Figure 1) that combines visual attention with the information bottleneck approach [9] of training deep models through supervised learning [1, 4, 6]. We define $\mathbf{z}$ as a bottleneck latent representation of an attention weighted feature encoding $\mathbf{A}_\mathcal{I} = \boldsymbol{\alpha}_\mathcal{I} \cdot \mathbf{F}_\mathcal{I}$ of the input features $\mathbf{F}_\mathcal{I}$. We leverage the mid-level input representation to separate the subset of dense inputs into a set $\mathcal{S} \subset \mathcal{I}$. Conditioned on $\mathbf{z}$ and $\mathcal{S}$, the motion generator finally predicts the target $\mathcal{Y}$. Our goal is to learn both the attention weighting function $\boldsymbol{\alpha}_\mathcal{I}$ and an encoding $\mathbf{z}$ that is maximally informative about the target $\mathcal{Y}$. To prevent $\mathbf{z}$ from being the identity encoding of the inputs and to focus the network on specific areas of causality, we impose an information bottleneck constraint on the complexity of $\mathbf{z}$ by a pooling operation. We preserve spatial information in the attention map by incorporating a positional encoding step, and encode non-local information by using Atrous convolutions.

We evaluate our approach on the large-scale dataset from [2] and show quantitative and qualitative results illustrating that our generated attention maps result in much sparser (and thus more interpretable) visualization of the internal states than a baseline visual attention model. We also show that our approach improves the motion generation accuracy in contrast to a traditional visual attention model that results in decreased accuracy.

2. Related Work

Explainability of deep neural networks has seen growing interest in computer vision and machine learning [5]. In landmark work, Zeiler *et al*. [12] utilized deconvolution layers to visualize the internal representation of a ConvNet. Bojarski *et al*. [3] developed a richer notion of contribution of a pixel to the output, while other approaches [14, 8] have explored synthesizing an image causing high neuron activations. However, a difficulty with de-convolution based approaches is the lack of a formal notion of contribution of

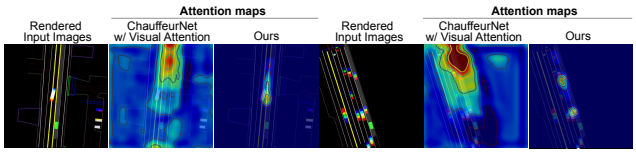

Figure 3. Comparison of attention maps from our model against those from a baseline visual attention model. Note that our heatmaps are much sparser and thus more interpretable.

spatially-extended features (rather than pixels).

Attention-based approaches [11] have been increasingly employed for improving a model's ability to explain by providing spatial attention maps that highlight areas of the image that the network attends to. Kim *et al*. [7] utilize an attention model followed by additional salience filtering to show regions that causally affect the output. To reduce the complexity of explanations, Wang *et al*. [10] introduce an instance-level attention model that finds objects (i.e. cars and pedestrians) that the network needs to pay attention to. Such attention may be more intuitive and interpretable for users to understand the model's behavior.

However, the model needs to take the whole input context as an additional input, which may compromise the causality of the attention – explanations may not represent causal relationships between the system's input and its behavior. To preserve the causality, we use a top-down representation of the environment as an input, which consists of information around the agent rendered in separable channels.

3. Visual Attention

ChauffeurNet. Bansal *et al*. [2] introduced a mid-to-mid driving network called *ChauffeurNet* that recurrently predicts future poses of the agent by processing a top-down representation of the environment as an input. The inputs $\mathcal{I}$ to this network consist of information about the roadmap, traffic lights, dynamic objects, etc. rendered in separate channels into a common top-down view coordinate system around the agent. The model predicts future agent poses $\mathcal{Y}$ in the same top-down view (see Figure 2).

The rendered inputs $\mathcal{I}$ are fed to a convolutional *FeatureNet*, which outputs features $\mathbf{F}$ that capture the environmental context and the intent. This feature $\mathbf{F}$ (of size $w \times h \times d$) contains a set of d-dimensional latent vectors over the spatial dimension, i.e. $\mathbf{F} = \{\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_l\}$, where $\mathbf{f}_i \in \mathcal{R}^d$ and l (= $w \times h$) is the spatial dimension of the extracted features. The feature encoding $\mathbf{F}$ is fed to a recurrent neural network, *AgentRNN*, which predicts the output agent poses $\mathcal{Y}$.

ChauffeurNet with Visual Attention. The goal of visual attention is to find an attended feature $\mathbf{A} = \{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_l\}$, where $\mathbf{a}_i \in \mathcal{R}^d$ from the original fea-

ture $\mathbf{F}$. As discussed by several works [11, 7], the attended features can be computed as $\mathbf{a}_i = \pi(\alpha_i, \mathbf{f}_i) = \alpha_i \mathbf{f}_i$ for $i = \{1, 2, \ldots, l\}$, where α_i are scalar attention weights in $[0, 1]$ satisfying $\sum_i \alpha_i = 1$. These weights are estimated from the input features $\mathbf{F}$ typically by a multi-layer perceptron, i.e. $\alpha_i = f_{\text{MLP}}(\mathbf{f}_i)$ where the parameters of f_{MLP} are learned as part of training the entire model end-to-end. Since the attention weights vary spatially and depend on the input (via the features $\mathbf{F}$), they can be visualized as an *attention heatmap* aligned with the input image, with brighter regions reflecting areas salient for the task.

To allow us to explain the driving decisions made by ChauffeurNet, we apply this vanilla visual attention approach by replacing the original features $\mathbf{F}$ with the attended features $\mathbf{A}$ as shown in Figure 4. As shown in Figure 3, this approach generates vague and verbose attention maps which do not add to the interpretability of the model. Therefore, we use this approach as a baseline for our "Attentional Bottleneck" approach.

4. Attentional Bottleneck

We propose a novel architecture called *Attentional Bottleneck* with a focus on generating sparse and fine-grained visual explanations. We encode the environment $\mathcal{I}$ through an information bottleneck that serves to restrict information in the input to only the most relevant parts of the input, and thus allows the driving model to focus on specific features in the environment. We tie this feature selection to the spatial distribution of features by employing a spatial attention mechanism before the bottleneck. While the driving task involves focusing on specific objects and entities in the scene for the immediate driving decisions, humans also employ a holistic understanding of some elements of the environment e.g. of the overall map. We find that compressing this kind of dense information through the bottleneck either leads to dense attention maps or degrades the model performance. Therefore, we leverage the mid-level separable input representation and provide the model full access to a subset of inputs $\mathcal{S} \subset \mathcal{I}$ containing the dense context about the environment, through a separate branch. This frees up the bottleneck branch to focus on specific parts of the input (e.g. specific objects) making the attention map sparser and more interpretable.

Grounding Attentional Bottleneck into AgentRNN. Like the baseline model, the inputs $\mathcal{I}$ are first encoded into features $\mathbf{F}_\mathcal{I}$ by the *FeatureNet* network. To capture non-local information, we propose an Atrous Spatial Attention layer that computes the attention weights $\boldsymbol{\alpha}_\mathcal{I}$ and outputs the attended features $\mathbf{A}_\mathcal{I}$. The attended features are depth-concatenated with a positional encoding $\mathbf{V}$ followed by a multi-layer perceptron g_{MLP}, and an average pooling layer

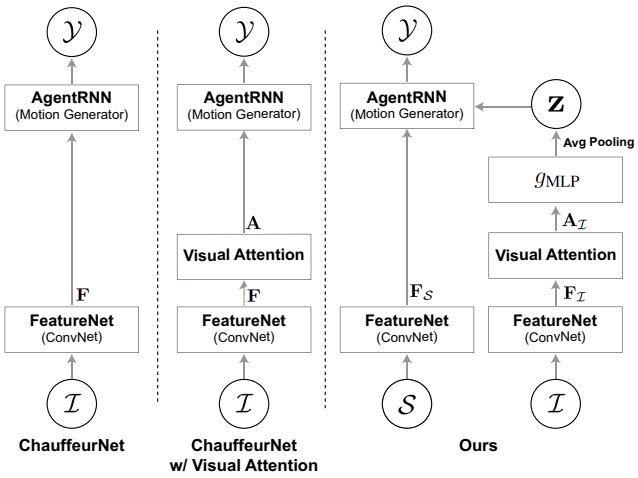

Figure 4. Attentional Bottleneck design compared with a baseline visual attention model applied to ChauffeurNet.

to generate the final bottleneck representation $\mathbf{z}$.

$$\mathbf{z} = \sum_{i=1}^{l} g_{\text{MLP}}([\mathbf{a}_i; \mathbf{v}_i]) \tag{1}$$

The dense scene context inputs $\mathcal{S}$ are similarly encoded into features $\mathbf{F}_\mathcal{S}$ using another *FeatureNet* network with identical architecture. We modify *AgentRNN* to incorporate the bottleneck vector by concatenating it with each of the features $\mathbf{f}_i \in \mathbf{F}_\mathcal{S}$.

5. Experiments

We trained our models end-to-end on the large-scale dataset from [2] with ChauffeurNet's default losses. To quantitatively evaluate motion generation performance, we use two widely-used Euclidean distance-based metrics: (i) the average displacement error (ADE) $\frac{1}{K}\sum_{k=0}^{K} ||\hat{\mathbf{p}}_k - \mathbf{p}_k^{gt}||_2$, and (ii) the final displacement error (FDE) $||\hat{\mathbf{p}}_K - \mathbf{p}_K^{gt}||_2$, where $K = 10$ is the total number of predicted waypoints, and the superscript gt denotes the ground-truth values. To measure the sparseness of the generated attention maps, we measure the entropy of the generated attention heat map α, i.e. $S(\alpha) = -\sum_{i=1}^{l} \alpha_i \log \alpha_i$.

Analysis. Figure 6 compares our motion generation and attention sparsity metrics across different model variants. We observe that the incorporation of visual attention for improving the interpretability of the baseline model degrades its performance as measured by the larger ADE and FDE numbers (model B). This is not the case with our attentional bottleneck model where we observe improved ADE and FDE numbers (model C) – possibly due to improved focus by the model on specific causal factors. Examples in Figure 3 compare our attention maps to those from the visual attention model, and confirm that the latter generates verbose attention heat maps – finding all potentially salient

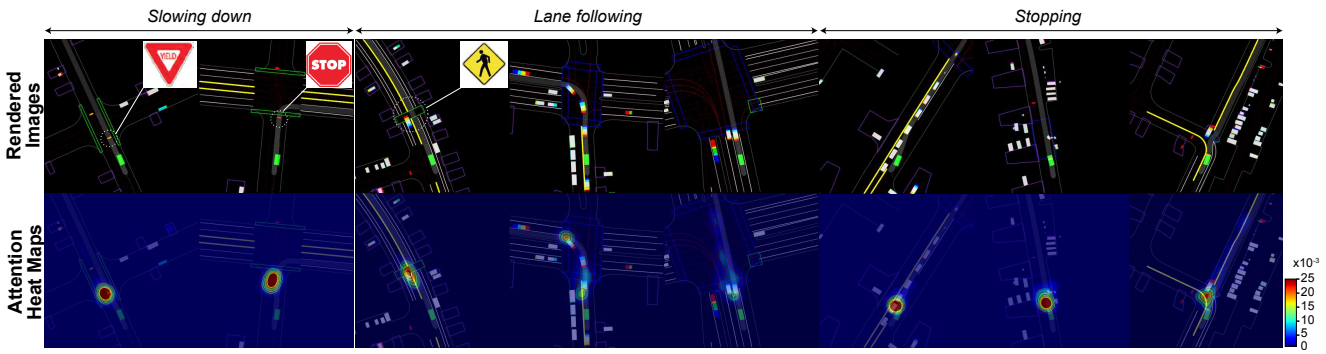

Figure 5. We provide typical examples of attention heat maps in diverse driving scenarios. Our model attends to driving-related visual cues like highlighting stop/yield signs, crosswalks or cars ahead that cause braking, road contours on curved roads, or multiple pinch points from parked cars on narrow roads.

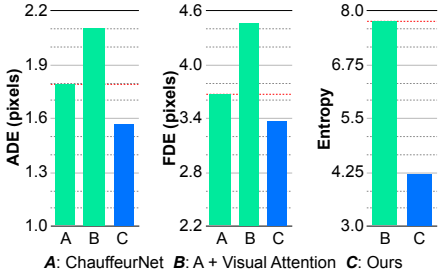

Figure 6. Comparison of motion generation performance and attention map sparsity between baseline ChauffeurNet, visual attention and our Attentional Bottleneck design.

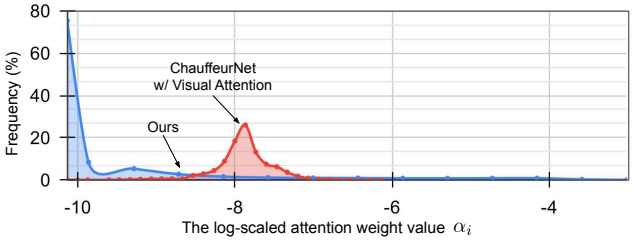

Figure 7. Distribution comparison of the (log-scaled) scalar attention weight values α_i.

objects. In contrast, our model provides much sparser attention heat maps which are easier to associate with specific objects or rendered features and are thus easier to interpret. Figure 5 shows several examples of our attention output across common driving scenarios that involve slowing down, stopping, avoiding obstacles etc. This is evident by comparing their distributions as shown in Figure 7 where the attention weights from our model are mostly concentrated around zero probability values.

6. Conclusions

We described an approach for improving interpretablity of a mid-to-mid deep driving model by augmenting a visual attention model with an attentional bottleneck layer. Our results highlight sparse attention maps which are easy to interpret and do not degrade model performance. We see opportunity in taking this further to generate instance level attention maps and to also use these maps as a guide to improving the performance of the baseline driving model.

Acknowledgements

We thank Dragomir Anguelov, Anca Dragan, and Alexander Gorban at Waymo Research, John Canny, Trevor Darrell, Anna Rohrbach, and Yang Gao at UC Berkeley for their helpful comments.

References

[1] A. A Alemi, I. Fischer, J. V Dillon, and K. Murphy. Deep variational information bottleneck. *ICLR*, 2017.

[2] M. Bansal, A. Krizhevsky, and A. Ogale. Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst. *RSS*, 2019.

[3] Mariusz Bojarski, Anna Choromanska, Krzysztof Choromanski, Bernhard Firner, Larry Jackel, Urs Muller, and Karol Zieba. Visualbackprop: visualizing cnns for autonomous driving. *arXiv preprint*, 2016.

[4] M. Chalk, O. Marre, and G. Tkacik. Relevant sparse codes with variational information bottleneck. In *NeurIPS*, 2016.

[5] David G. Explainable artificial intelligence (xai). *DARPA*, 2017.

[6] A. Goyal *et al*. Infobot: Transfer and exploration via the information bottleneck. *ICLR*, 2019.

[7] J. Kim and J. Canny. Interpretable learning for self-driving cars by visualizing causal attention. *ICCV*, 2017.

[8] 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.

[9] N. Tishby, F. C Pereira, and W. Bialek. The information bottleneck method. *The Allerton Conf. on Communication, Control, and Computing*, 1999.

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

[11] K. Xu *et al*. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2015.

[12] M. D Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *ECCV*, pages 818–833. Springer, 2014.

[13] W. Zeng, W. Luo, S. Suo, A. Sadat, B. Yang, S. Casas, and R. Urtasun. End-to-end interpretable neural motion planner. In *CVPR*, 2019.

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