Going Deeper: Autonomous Steering with Neural Memory Networks

Tharindu Fernando Simon Denman Sridha Sridharan Clinton Fookes
{t.warnakulasuriya, s.denman, s.sridharan, c.fookes}@qut.edu.au

Image and Video Research Laboratory,
Queensland University of Technology,
Australia.

Abstract

Although autonomous driving is an area which has been extensively explored in computer vision, current deep learning based methods such as direct image to action mapping approaches are not able to generate accurate results, making their application questionable. This is largely due to the lack of capacity of the current state-of-the-art architectures to capture long term dependencies which can model different human preferences and their behaviour under different contexts. Our work explores a new paradigm in deep autonomous driving where the model incorporates both visual input as well as the steering wheel trajectory and attains a long term planning capacity via neural memory networks. Furthermore, this work investigates optimal feature fusion techniques to combine these multimodal information sources, without discarding the vital information that they offer. The effectiveness of the proposed architecture is illustrated using two publicly available datasets where in both cases the proposed model demonstrates human like behaviour under challenging situations including illumination variations, discontinuous shoulder lines, lane merges, and divided highways, outperforming the current state-of-the-art.

1. Introduction

Autonomous driving has been a popular topic among researchers, receiving attention from both academic groups and commercial projects such as Google self-driving cars, Uber and Tesla.

Even though video cameras offer cheap solutions for the data capture needs of these systems, current state-of-the-art methods use high cost devices such as laser sensors and radars in addition to vision sensors rather than completely relying on video input. We believe current deep learning based computer vision approaches [3,19,24] will ultimately have the power necessary for safe open road driving using vision based sensors alone.

Our work draws inspiration from recent seminal work in [24] in which the authors model the autonomous driving problem as predicting future motion given the present observation and previous history of the agent. This work, with the aid of historical states of the agent, achieves commendable results in contrast to popular behaviour reflex approaches [2, 11, 20–22] that directly map an input image to a driving action.

Even though this shallow memory architecture is capable of modelling short term dependencies and achieves significant improvement, they fail to capture long term dependencies due to an inherent problem with the sequential LSTM architecture, that the hidden state activations are dominated by the most recent inputs [6, 9]. As more and more observations come into the memory, most recent inputs dominate the memory, completely discarding the vital factors from long term history. Therefore, the current state-of-the-art method of [24], which uses sequential LSTMs to estimate the behaviour, fails to model important factors such as road and weather conditions, or traffic conditions and density which are vital for long term planning, due to the underlying limitations of the LSTM representation.

The proposed approach is shown in Fig. 1 (a) and in Fig. 1 (b) we compare it to the method proposed in [24]. We adopt recent advances in tree memory networks [9] to develop the ability to capture both long-term and short-term temporal dependencies which is crucial to effectively model driving behaviour [3]. Both models receive a sequence of video frames as the spatial input. In [24] the authors model past ground truth sensor information for speed and angular velocity as a trajectory, where as in our model we utilise the trajectory steering wheel angle. Then the video frames are passed through a Convolution Neural Network (CNN) to encode the visual input. The significant differences between the two approaches arise when considering the process of modelling historical states. In the method proposed in [24], the authors directly merge the two input sequences and pass the merged feature through a shallow Long Short

Term Memory (LSTM) [12] layer; where as in the proposed method we utilise separate LSTM models for two input streams and map their evolution via two separate external memories, before merging current LSTM outputs together with the memory outputs in order to generate the final output.

This work offers several novel contributions. First, we take inspiration from recent works in [24] and [9] to build a new approach to capture dependencies between dense spatial inputs together with a sparse steering wheel angle trajectory. Second, we propose and evaluate three fusion methods to fuse spatial and trajectory memory states with the present input. Finally, we report experimental results of the proposed model on two popular publicly available driving datasets and compare to various baselines, where in both cases the proposed architecture is shown to outperform the current state-of-the-art.

2. Related work

2.1. Autonomous driving

From the first attempts by Pomerleau et. al [21] to use a neural network for autonomous driving, which used a shallow network to directly map pixel values to simple driving commands, there has been a growing interest among researchers in achieving fully autonomous vehicle navigation in complex real world scenarios.

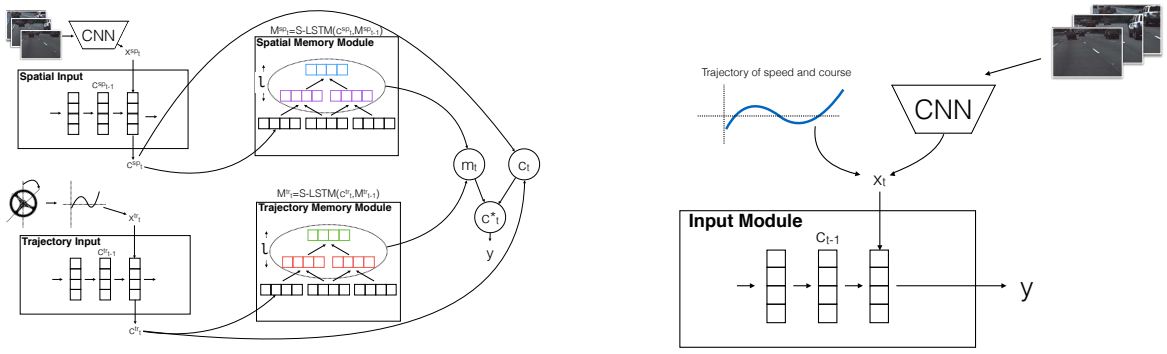
Approaches related to autonomous driving can be broadly categorised into mediated approaches and behaviour reflex approaches. In mediated approaches [1, 8, 10, 16], the authors map pixels to pre-defined affordance measures, such as lanes, pedestrians, traffic lights and surrounding cars. For example in [8, 16] the authors generate bounding boxes on detected cars and in [1] splines for detected lane markings. In order to utilise such detections for navigation in [1, 8, 16] the authors generate a layout of the intersection and traffic details by passing those various detections through a probabilistic model. Even though this approach generates interpretable results, evaluating such a complete set of measures in real world conditions may be computationally expensive and unmanageable. Furthermore, in real world conditions with missing lane markings, discontinuous shoulder lines, varying illuminations, and cluttered backgrounds, these approaches tend to produce false detections, which makes their applicability limited [3].

The second class of techniques, behaviour reflex approaches [2, 11, 20–22], are motivated by human behaviour and construct a direct mapping from an image to a steering angle. Although this idea is straightforward, it has not been capable of generating accurate results for several reasons [3]. Firstly, as humans we have our own preferences and this may lead to different styles of driving. Hence under similar contexts human drivers may make completely different decisions. One driver may give way to the merging traffic where as another may not. Secondly, blind pixel to action mappings cannot perform long term planning. Even though it can generate reactive behaviours to avoid collisions, such low level modelling fails to capture the underlying semantics of driving.

Recently, with the advances in recurrent neural networks, an extension to the behaviour reflex approach, called privileged training, is presented [24]. The authors investigate the use of deep visual features extracted via a convolutional neural network (CNN) [4] model and then utilise a LSTM for sequence modelling. Even though it has been able to achieve encouraging results when compared with behaviour reflex approaches, as shown in [5], such a naive CNN to LSTM mapping fails to capture vital dependencies among the input sequences. Furthermore such direct merging of a sparse trajectory input with dense visual features (see Fig. 1 (b)) may lead to a blind mapping of features without fully determining the degree of attention that each feature stream requires.

2.2. Memory architectures

Along with deep learning models such as Recurrent Neural Networks (RNN) and LSTMs a vast number of approaches [13, 15, 17, 23] have been proposed that utilise memory architectures in order to better predict the variable of interest. The memory stores important facts from historical inputs in order to capture temporal dependencies. For instance, in the area of natural language processing Weston et. al [23] have used a memory module to capture the semantics within different languages. In similar works, in [13] and in [14], the "memory module" has been utilised in order to capture the coherence among images and their captions. The experimental results presented in [9] show that such naive memory architectures [15, 23] capture only short term dependencies, instead of capturing long term dependencies. But capturing both long term and short term dependencies within the memory is extremely important for long term planning. For instance by having a long term memory we can capture how a particular driver behaves in different contexts, adapts to different weather conditions, or traffic congestion; allowing the memory to describe his or her driving style, and utilise that information for future planning. As a consequence of this need to capture both long and short term dependencies, a memory architecture called Tree Memory Networks [9] is proposed, which structures the memory as a hierarchical recursive tree structure, instead of a flat, sequential layer of LSTM cells; and this approach is shown to outperform sequential memory architectures in path prediction tasks.



(a) Proposed Method

(b) Method proposed in [24]

Figure 1. Comparison between the proposed method and the current state-of-the-art approach proposed in [24]. In both methods the visual input is encoded with a CNN. The authors in [24] directly merge the CNN output with the speed and angular velocity trajectory and pass it through an LSTM layer. In contrast, in the proposed method, the encoded visual input and the steering wheel angle trajectory is passed through separate LSTM models and their long term history is mapped via two separate external memories. The final output is generated by merging current LSTM outputs together with the memory outputs.

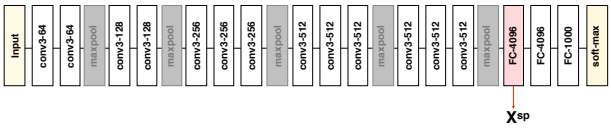


Figure 2. VGG-16 model [4] with 13 convolutional (Conv3-number-of-kernels) layers followed by 3 fully connected (FC-number-of-hidden-units) layers. We extract activations from the first fully connected layer (denoted X^{sp}).

3. Method

3.1. Spatial and trajectory input modules

3.1.1 Visual encoder

Each video frame in the database is encoded, using a visual encoder which encodes the spatial information in a discriminative manner. In the proposed model we utilise VGG-Net [4] pre trained on ImageNet. In order to capture discriminative driving actions more precisely, we have fine tuned the network with the frames in our datasets. Finally, as shown in Fig. 2, the first fully connected layer activations are extracted as inputs to the spatial LSTM.

Let $X_i^{sp} = [\mathbf{x_{i,1}^{sp}}, \mathbf{x_{i,2}^{sp}}, \ldots, \mathbf{x_{i,S}^{sp}}]$ be the S length sequence of the first fully connected layer activations from the VGG network for the i^{th} example. The spatial input module computes a vector representation for the input sequence via a LSTM layer,

$$\mathbf{c_t^{sp}} = f_{LSTM}(\mathbf{x_t^{sp}}, \mathbf{c_{t-1}^{sp}}), \tag{1}$$

where $\mathbf{c_t^{sp}} \in \mathbb{R}^{k^{sp}}$, and k^{sp} is the embedding dimension of the LSTM.

3.1.2 Trajectory LSTM

Let $X_i^{tr} = [\mathbf{x_{i,1}^{tr}}, \mathbf{x_{i,2}^{tr}}, \ldots, \mathbf{x_{i,S}^{tr}}]$ be the S length sequence of steering angle for the i^{th} example. The trajectory input module computes a vector representation for the input sequence via a LSTM layer,

$$\mathbf{c_t^{tr}} = f_{LSTM}(\mathbf{x_t^{tr}}, \mathbf{c_{t-1}^{tr}}), \tag{2}$$

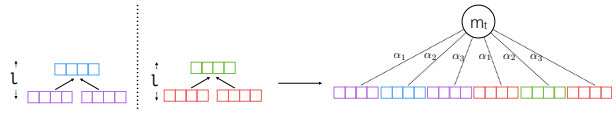
where $\mathbf{c_t^{tr}} \in \mathbb{R}^{k^{tr}}$, and k^{tr} is the embedding dimension of the LSTM.

3.2. Spatial and trajectory memory modules

In order to map the coherence between spatial sequences we utilise a spatial memory module. Consider N to be a queued sequence of historical LSTM embeddings for the spatial input, with length p and embedding dimension k ($N \in \mathbb{R}^{p \times k}$). Based on the exemplary results obtained by Fernando et al. [9] for long term dependency modelling, we adapt the same S-LSTM memory model as proposed in [9].

When computing an output at time instance t we extract out the tree configuration at time instance $t-1$. Let $\mathbf{M_{t-1}^{sp}} \in \mathbb{R}^{k \times 2^l - 1}$ be the memory matrix resultant from concatenating nodes from the top of the tree to $l = [0, \ldots]$ depth. The motivation behind using multiple nodes instead of a single node is to capture the different levels of abstraction that exist in the memory network. When considering dense inputs such as fully connected layer activations from a CNN, this can be extremely useful to increase the capacity of the model.

(a) Individual attention based fusion with LSTM memory cells



(b) Shared attentions based fusion

(c) Hierarchical fusion

Figure 3. Comparison between different fusion methods. The extracted memory cells of both the spatial memory module and the trajectory memory module are shown on the left. The fusion methods are shown on the right. Memory cells at different levels are shown in different colours.

The temporal dependencies between the steering angles can be mapped in a similar way, and as such a second memory model is used to model these, with data extracted in the same manner. Following the method presented in [9], at each time step we update the content of both spatial and temporal memory units.

3.3. Feature fusion

After feeding data to the memory modules, we have four data sources: the LSTM encodings for the two inputs (spatial and steering angle), and the memory outputs for these same two data sources. The structure that we utilise in order to incorporate information from these different sources is a vital factor for the modelling process. Even though the spatial and trajectory memories have the ability to capture the long term behaviour of the human drivers, if the fusion approach does not consider the features, in order to understand what the salient aspects of the memory components and how useful they are under certain contexts, then the long term planning process will be ineffective.

Therefore in order to fuse these neural networks we consider three approaches: i) a naive approach that treats each modality separately (see Section 3.3.1) ii) a joint approach that shares the attention between the two modalities (see Section 3.3.2) and iii) a hierarchical approach that enables deep layer wise fusion of the two memory outputs (see Section 3.3.3) Fig. 3 illustrates these different feature fusion architectures which are further described in the following sub-sections.

3.3.1 Individual attention fusion (fu-ia)

As shown in Fig. 3 (a), we have simply flattened the extracted memory cells and generated individual attention values for each memory cell in the following manner. Let f^{score} be an attention scoring function which can be implemented as a multi-layer perceptron [18],

$$m_t^{sp} = f^{score}(\mathbf{M_{t-1}^{sp}}, \mathbf{c_t^{sp}}), \tag{3}$$

$$\alpha^{sp} = \mathrm{softmax}(\mathrm{m_t^{sp}}). \tag{4}$$

The most relevant memory items for the current input are extracted via an attention mechanism derived using Eq. 3 and Eq. 4,

$$z_t^{sp} = \mathbf{M_{t-1}^{sp}}[\alpha^{sp}]^T. \tag{5}$$

Similarly, the attention of the trajectory memory module can be evaluated as,

$$m_t^{tr} = f^{score}(\mathbf{M_{t-1}^{tr}}, \mathbf{c_t^{tr}}), \tag{6}$$

$$\alpha^{tr} = \mathrm{softmax}(\mathrm{m_t^{tr}}), \tag{7}$$

$$z_t^{tr} = \mathbf{M_{t-1}^{tr}}[\alpha^{tr}]^T. \tag{8}$$

Then the final output y, which is the steering angle in degrees normalised between 0 and 1, can be represented as,

$$y_t = ReLU(W^{sp} z_t^{sp} + (1 - W^{sp}) c_t^{sp} + W^{tr} z_t^{tr} + (1 - W^{tr}) c_t^{tr}), \tag{9}$$

where W^{sp} and W^{tr} are the respective output weights, learned during training.

3.3.2 Shared attention fusion (fu-sa)

In this fusion method we have used a shared attention for both memory modules. Fig. 3 (b) depicts the idea. Let f_s^{score} be an attention scoring function which accepts both memory inputs and current contexts and generates attention weights similar to the pattern shown in the figure,

$$m_t = f_s^{score}(\mathbf{M_{t-1}^{sp}}, \mathbf{M_{t-1}^{tr}}, \mathbf{c_t^{sp}}, \mathbf{c_t^{tr}}), \tag{10}$$

$$\alpha = \mathrm{softmax}(\mathrm{m_t}), \tag{11}$$

$$z_t^{sp} = \mathbf{M_{t-1}^{sp}}[\alpha]^T. \tag{12}$$

$$z_t^{tr} = \mathbf{M_{t-1}^{tr}}[\alpha]^T. \tag{13}$$

Then the final output can be represented as,

$$y_t = ReLU(W^{sp} z_t^{sp} + (1 - W^{sp}) c_t^{sp} + W^{tr} z_t^{tr} + (1 - W^{tr}) c_t^{tr}), \tag{14}$$

where W^{sp} and W^{tr} are the respective learned output weights.

3.3.3 Hierarchical fusion (fu-h)

In this method we map the extracted memory outputs hierarchically using an S-LSTM module (See Fig. 3 (c)). The motivation behind the hierarchical fusion mechanism is to retain the salient features from the two memory networks by representing them with different levels of abstraction. For example, the work by Zhu et al. in [26] has shown that recursive LSTM model is capable of achieving state-of-the-art performance on semantic composition tasks, outperforming flat, sequential LSTM structures. By combining the activations from spatial and trajectory memories hierarchically, we aim to capture how human drivers adapt to different conditions in a logical and orderly fashion.

Furthermore, in [5], the authors investigate the usage of different information sources together, and their evaluations revealed that hierarchically modelling features using an S-LSTM module results in more accurate predictions. This is further evidence to verify that different feature streams should not be combined directly. Instead, they should be merged in a hierarchical form as these different streams may complement each other, providing vital information under different contexts, in order to form a very strong system. Following this reasoning, we extract a fused feature as follows,

$$c^*{}_t = f_{S-LSTM}(\mathbf{M^{sp}_{t-1}}, \mathbf{M^{tr}_{t-1}}). \tag{15}$$

Then the final output can be represented as,

$$y_t = ReLU(W^c c^*_t + W^{sp} c^{sp}_t + W^{tr} c^{tr}_t), \tag{16}$$

where W^c, W^{sp} and W^{tr} are the respective learned output weights.

4. Experimental results

4.1. Datasets

4.1.1 Comma.ai dataset

The comma.ai dataset [2] is a publicly available driving dataset with 7.25 hours of driving data. The video data is recorded from a dashboard camera at 20 FPS. The dataset also has several sensor recordings (i.e. car speed, steering angle, GPS, gyroscope) that are measured at different frequencies and synchronised to the same sampling rate. As a preprocessing step we extract non-overlapping video subsequences 36 frames in length. From the resultant video sequences along with corresponding steering wheel angles, the first 20,883 examples are chosen for training and the remaining 8,951 for testing. Video frames are down-sampled to 224 x 224 for processing by the VGG-16 model network.

4.1.2 Udacity's self-driving car dataset

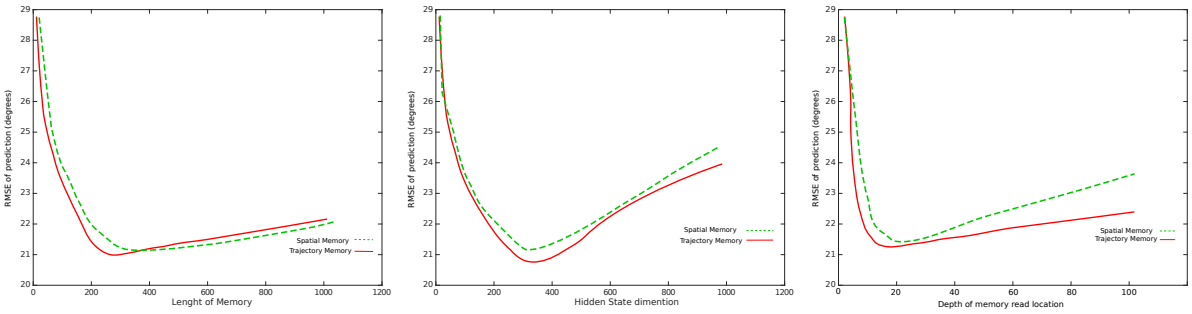
The Udacity self-driving car data set [11] contains video frames from three front-facing cameras (left, centre, and right) and vehicle measurements such as speed and steering wheel angle, recorded from the vehicle driving on the road. As the video and vehicle measurements are sampled at different sampling rates a synchronisation process is required. Therefore we perform the following operations as pre-processing: i) synchronising video frames with measurements from the vehicle, ii) selecting only centre camera frames, iii) down sample the video frames to 224 x 224. iv) extract video sequences of 36 frames in length. The resultant dataset contains 77,308 samples with corresponding steering wheel angles. We selected the first 54,115 samples for training and the remaining 23,193 examples are used for testing.

4.2. Quantitative evaluation

The VGG-16 feature extractor was pre-trained with an analog output unit to learn the recorded steering angle from randomly selected single frame images from Udacity's self-driving car dataset. After that offline training process the proposed network is added and we use stochastic gradient descent (SGD) with momentum of 0.99 and a batch size of 100. Models are evaluated using root mean square error (RMSE).

Hyper parameters, the length of the memory module, p, the embedding dimension, k, and the depth of extracted memory matrix, l, of the two memory modules are evaluated experimentally with reference to the hierarchical fusion model (fu-h). Fig. 4 (a) shows the variation in RMSE against p for the spatial and trajectory memory modules in solid red and dashed green lines respectively. For the spatial memory, the error converges around $p = 250$, and for trajectory memory, the error converges around $p = 310$. Therefore we set the value of p as 256. Fig. 4 (b) shows the variation of average RMSE against k for the two memory modules. As the error converges around 300 hidden units, the embedding dimension is set to 300. With the aid of similar experiments we evaluated the depth of memory read location, l, and the resultant error plots are shown in Fig. 4 (c) where we are able to conclude that $l = 20$ produces optimal results.

The experimental evaluations are tabulated in Tab. 1. In the Comma.ai dataset there aren't standard training and testing splits. Furthermore for Udacity's self-driving car datasets the ground truth labels for the testing split are not available. Therefore we evaluated all the baselines for our training-testing splits. For Udacity dataset, we divided the provided training set to training-testing splits. As the baseline models we use the deep architectures proposed in [24] and [20]. As a Behavioural Reflex method we utilise [20] and for the Mediated Perception Approach, as given in [24], we first compute the segmentation output of every frame in Comma.ai and Udacity's self-driving car datasets using the Multi-Scale Context Aggregation approach described in



(a) RMSE vs length of memory module p (b) RMSE vs embedding dimension k (c) RMSE vs depth of extracted matrix l

Figure 4. Parameter evaluation: We evaluate the length of the memory module, p (see (a)), the embedding dimension, k (see (b)) and the depth of the extracted matrix, l (see (c)).

Method	RMSE (in degrees)	
	Comma.ai	Udacity
Behavioural Reflex [20]	25.6	31.8
Mediated Perception [24]	23.7	29.5
Privileged training [24]	23.1	27.3
fu-ia	21.4	24.3
fu-sa	22.5	24.8
fu-h	**20.1**	**22.5**

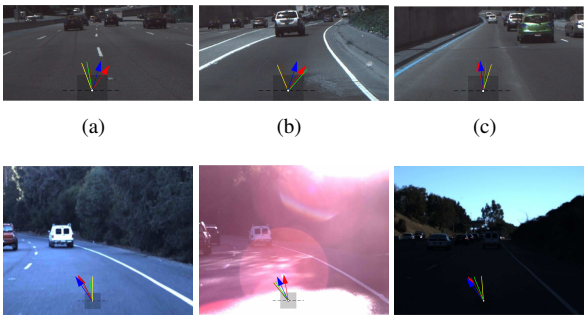
Table 1. Comparison of our results to the state-of-the-arts for autonomous driving on Comma.ai and Udacity datasets

[25]. For the segmentation mask we utilise the Cityscape [7] mask. Then mimicking the stage-by-stage training, we train the LSTM independently from the segmentation process. For the Privileged training approach we use the method proposed in [24].

Results are presented in Tab. 1, and we observe that all of our fusion methods outperform the current state-of-the-art. The behavioural reflex model [20] has the lowest accuracy verifying our assertion that they do not possess enough capacity to perform long term planing. The privileged training model of [24] obtains better performance but still it cannot completely model the driving style of the driver due to it's shallow LSTM structure, which in turn leads to erroneous predictions.

Our hierarchical fusion method (fu-h) has the lowest error illustrating that a deep layer wise fusion method is able to capture different levels of abstraction and semantics that are present in the different input modalities. Models **fu-ia** and **fu-sa** do not possess such ability but still **fu-ia** outperforms **fu-sa** demonstrating that different input streams possess information which requires different degrees of attention. Therefore in **fu-ia** we are able to learn that attention through back propagation where the model learns what acts as the main information queue and what the complementary information is. Model **fu-h** takes this concept to its next level where it offers layer wise merging of temporal semantics from the two information streams.

4.3. Qualitative evaluation



(a) (b) (c)

(d) (e) (f)

Figure 5. Qualitative evaluation: Results from Comma.ai dataset (a)-(c) and Udacity dataset (d)-(f), predictions of fu-h model in red, behavioural reflex approach in yellow, mediated perception approach in green, privileged training approach in brown and ground truths in blue.

In Fig. 5 we show prediction results from our hierarchical fusion model (fu-h model) in red, the behavioural reflex approach in yellow, the mediated perception approach in green and the privileged training approach in brown for Comma.ai and Udacity datasets. The blue arrow shows the driver's action. Lane change behaviour where the driver moves from the left lane to the right lane is shown in Fig. 5 (a) and in sub-figures (b) and (c) we show the lane following behaviour for left and right curved lanes. Challenges with the Udacity dataset are shown in sub figures (d)-(f), where the illumination conditions vary from normal to direct sunlight and shadows within few seconds. The results illustrate that regardless of such changes, illumination variations, discontinuous shoulder lines, lane merges, and divided highways, the proposed model is able to generate accurate pre-

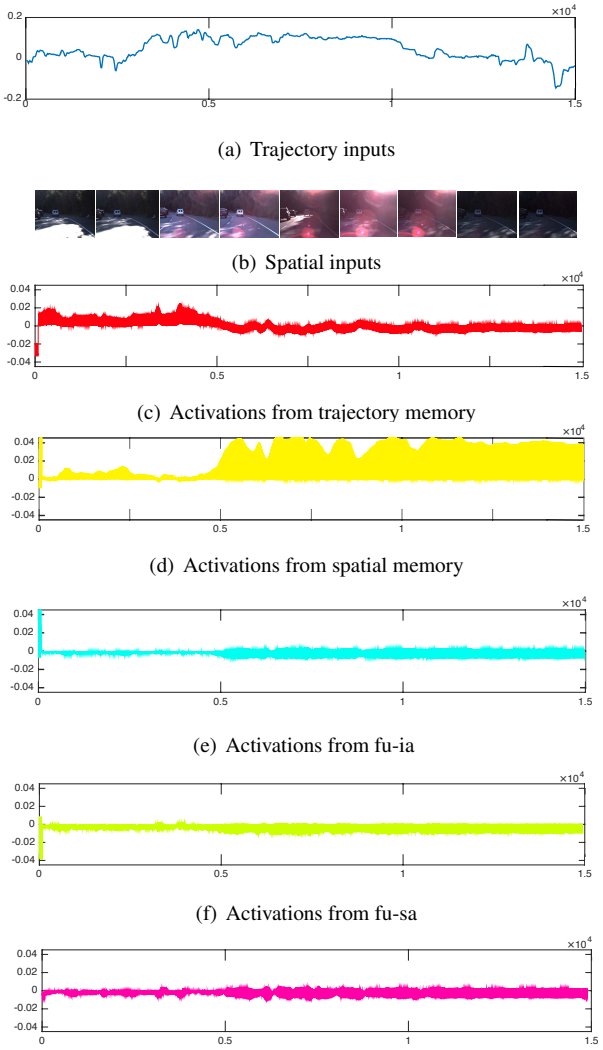
dictions and outperform all the baselines. For instance in Fig. 5 (a), due to the long term dependency modelling ability, the proposed model is able to anticipate the lane change behaviour of the driver and generate accurate predictions. In contrast, all the baselines have shown lane following behaviour.

Furthermore, as the illumination conditions fluctuate in sub figures (d)-(f), the predictions of the behavioural reflex approach become more erroneous as it directly maps pixels to an action. The mediated perception approach and privileged training approach have been able to tolerate that to a certain extent using the sequence modelling ability of the LSTMs but still the predictions aren't accurate when compared to the predictions of the **fu-h** model. As the proposed model has enough capacity to model how the human divers behave under different road conditions and illumination conditions, it has accurately anticipated the driver's behaviour.

4.4. Visualisation of memory activations

In 6 we visualise the temporal evolution of the 2 memory networks and the fusion methods. The trajectory of the steering wheel angle is illustrated in 6 (a). In subfigure (b) we visualise every 100^{th} frame that is given to the model as the spatial input. We have randomly selected a hidden unit from the top most layer of the trajectory and spatial memories and subfigures (c)-(d) shows activations from respective memories; and subfigures (e)-(g) visualise the activations from the proposed **fu-ia**, **fu-sa** and **fu-h** fusion methods respectively.

From the illustrations presented in Fig. 6 it is evident that both spatial and trajectory memory activations possess vital information. For instance between time steps 0.5 and 1.5 x10^4 the spatial input (6 (b)) becomes noisy with changes in lighting conditions and hence the memory activations show sudden fluctuations. In such scenarios, with the **fu-sa** fusion method where we are using the same attention weights for both the spatial and trajectory inputs, the model loses the chance to obtain complementary information that is available from the trajectory history. Therefore the small spikes of activations that are visible in the trajectory memory are not incorporated in the final memory output and it is further evident that the fusion process is mainly driven by the spatial memory output. With the individual attention mechanism (**fu-ia**) we allow the model to learn the different degrees of attention that different modalities of input should receive. The hierarchical structure of **fu-h** grants the opportunity for deep layer wise fusion of the salient features, allowing the model to pay careful attention towards different levels of abstraction. As the model gains enough capacity in order to jointly backpropagate and learn the instances in which it should vary it's attention, we observe a unique distribution of activations in Fig. 6 (g). Note that 0.5 to 1.5 x10^4 there exist several peaks and valleys of activations in Fig. 6 (g) that are not present in (e) and (f). This further illustrates our motivation as the model has obtained specific information from both spatial and trajectory memories demonstrating their importance.



(a) Trajectory inputs

(b) Spatial inputs

(c) Activations from trajectory memory

(d) Activations from spatial memory

(e) Activations from fu-ia

(f) Activations from fu-sa

(g) Activations from fu-h

Figure 6. Visualisation of activations of memory and fusion methods. First 2 rows show the trajectory and spatial inputs at different time steps. Rows (c)-(d) shows activations from trajectory and spatial memory respectively. In rows (e)-(g) we visualise the activations from fu-ia, fu-sa and fu-h fusion methods respectively.

5. Conclusion

This paper proposed a novel deep learning architecture for autonomous driving. Instead of blind pixel to action mapping or shallow planning, the proposed model incorporates both visual input as well as the steering wheel trajectory and attains a long term planning capability via neural Tree Memory Networks. Furthermore, we introduced three

fusion techniques to combine these multimodal information sources enabling optimal utilisation of the vital information that they provide. We tested our models in two challenging publicly available datasets and the experimental evaluations demonstrate that the proposed architectures have outperformed the current state-of-the-art and generate human like driving behaviour with the aid of long term modelling.

References

[1] M. Aly. Real time detection of lane markers in urban streets. In *IEEE Intelligent Vehicles Symposium*, pages 7–12. IEEE, 2008.

[2] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.

[3] C. Chen, A. Seff, A. Kornhauser, and J. Xiao. Deepdriving: Learning affordance for direct perception in autonomous driving. In *ICCV*, pages 2722–2730, 2015.

[4] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*, 2014.

[5] Q. Chen, X. Zhu, Z. Ling, S. Wei, and H. Jiang. Enhancing and combining sequential and tree lstm for natural language inference. *arXiv preprint arXiv:1609.06038*, 2016.

[6] X. Chen and C. Lawrence Zitnick. Mind's eye: A recurrent visual representation for image caption generation. In *CVPR*, pages 2422–2431, 2015.

[7] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, pages 3213–3223, 2016.

[8] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE T-PAMI*, 32(9):1627–1645, 2010.

[9] T. Fernando, S. Denman, A. McFadyen, S. Sridharan, and C. Fookes. Tree memory networks for modelling long-term temporal dependencies. *arXiv preprint arXiv:1703.04706*, 2017.

[10] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

[11] J. F. Heyse and M. Bouton. End-to-end driving controls predictions from images. 2016.

[12] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[13] A. Joulin and T. Mikolov. Inferring algorithmic patterns with stack-augmented recurrent nets. In *Advances in Neural Information Processing Systems*, pages 190–198, 2015.

[14] Ł. Kaiser and I. Sutskever. Neural gpus learn algorithms. *arXiv preprint arXiv:1511.08228*, 2015.

[15] A. Kumar, O. Irsoy, J. Su, J. Bradbury, R. English, B. Pierce, P. Ondruska, I. Gulrajani, and R. Socher. Ask me anything: Dynamic memory networks for natural language processing. *arXiv preprint arXiv:1506.07285*, 2015.

[16] P. Lenz, J. Ziegler, A. Geiger, and M. Roser. Sparse scene flow segmentation for moving object detection in urban environments. In *IEEE Intelligent Vehicles Symposium*, pages 926–932. IEEE, 2011.

[17] M. Malinowski and M. Fritz. A multi-world approach to question answering about real-world scenes based on uncertain input. In *Advances in Neural Information Processing Systems*, pages 1682–1690, 2014.

[18] T. Munkhdalai and H. Yu. Neural tree indexers for text understanding. *arXiv preprint arXiv:1607.04492*, 2016.

[19] D. Neil, J. H. Lee, T. Delbruck, and S.-C. Liu. Delta networks for optimized recurrent network computation. *arXiv preprint arXiv:1612.05571*, 2016.

[20] P. Penkov, V. Sriram, and J. Ye. Applying techniques in supervised deep learning to steering angle prediction in autonomous vehicles. 2016.

[21] D. A. Pomerleau. Alvinn, an autonomous land vehicle in a neural network. Technical report, Carnegie Mellon University, Computer Science Department, 1989.

[22] D. A. Pomerleau. *Neural network perception for mobile robot guidance*, volume 239. Springer Science & Business Media, 2012.

[23] J. Weston, S. Chopra, and A. Bordes. Memory networks. *arXiv preprint arXiv:1410.3916*, 2014.

[24] H. Xu, Y. Gao, F. Yu, and T. Darrell. End-to-end learning of driving models from large-scale video datasets. *arXiv preprint arXiv:1612.01079*, 2016.

[25] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.

[26] X.-D. Zhu, P. Sobhani, and H. Guo. Long short-term memory over recursive structures. In *ICML*, pages 1604–1612, 2015.