Generating Explanatory Saliency Maps for Mixed Traffic Flow using a Behaviour Cloning Model

Yasin M. Yousif[1] and Jörg P. Müller[1]

Department of Informatics, Clausthal University of Technology.
{yasin.yousif,joerg.mueller}@tu-clausthal.de

Abstract. Multi-agent mixed traffic modelling and simulation are needed for safety estimation in traffic situations. Many of the accurate traffic prediction models use deep learning methods. However, most of these models are considered black box models, which means that the output cannot be directly interpreted based on the input. However, such interpretation can be valuable, for example, in providing explanatory information about the model predictions for a simulation or a real-world dataset.
On the other hand, formulating the prediction problem as input to output mapping problem by defining it as markov decision process (MDP) is a more realistic approach to fully imitate the traffic entity behaviour.
Therefore, the presented method here implements a behaviour cloning approach with memoryless architecture. As a result, it is easier to link the output with the input using saliency maps extraction methods.
The calculated salience maps highlight the traversable areas for the agent to reach its destination, avoiding collision with other agents and obstacles. They also show the salient roads edges that influence the direction of the predicted movement.

Keywords: Mixed Traffic Modelling, Behaviour Cloning, Saliency Maps

1 Introduction

Traffic simulation is needed for estimation, optimization, prediction and understanding of vehicular or pedestrian traffic [1]. For example, it can help city planners evaluate the safety level in any proposed urban structure. However, the case of mixed traffic is more challenging than single mode traffic due to the higher complexity in the interactions between the different modes (e.g. vehicles, pedestrians, or cyclists).

Many deep learning models for mixed traffic prediction was proposed in the literature [2–5]. Although they provide low errors for many traffic datasets [6–8], they still lack a direct and accurate way to provide explanations of their predictions. This is due to the complexity and the size of their network architecture, which make the task of interpreting the output based on the input more difficult.

These interpretations are needed in traffic simulation programs because the simulation output is only the trajectories of the different agents in the traffic

area (The agent represents here the traffic entity model used in the simulation). For example, if the simulation shows that a pedestrian stops at the edge of a road while some cars are passing by, then there's no way to tell if the pedestrian is waiting for the cars or if he is just turning to change his direction. To be sure of the reason, the output should be linked in a meaningful way to the input.

The need for generating interpretations is also present in another field, namely, self-driving cars [9]. One important use case is to explain the software logic which led to an accident. It is also noted that traffic prediction modules are used as part of the self-driving cars software [10], therefore the same need for interpretation is present for these modules. However, the topic of interpretation in traffic prediction [11] did not receive as much attention as in the case of self-driving cars [12, 13].

An important point to consider when designing the traffic agent model is matching the model's input and output with the real input and output of the traffic participant. Therefore, a model is proposed here where the output is determined using only the current input, thus fulfilling the Markov property. The expected result is getting clear and meaningful saliency maps. These maps are grayscale images highlighting the highest influencing regions of the input, which contributed towards the prediction. Although, this isn't a full interpretation, it still provides insights into the model reasoning process.

After training this model on the mixed traffic datasets, the evaluation is done with respect to the accuracy of its predictions, and by analysing a representative set of examples and showing its corresponding saliency maps.

The paper contribution is in proposing a model architecture capable of predicting realistic waypoints for the traffic entity and at the same time generating saliency maps with insights into the model inner reasoning.

In the next section, a review of deep learning traffic models and of visual explanation methods is presented. After that, a perspective on a multi-agent implementation of the model is shown. Section 3 presents the neural network architecture, as well as the implemented explanation method. The results in quantitative way, and a set of visual explanatory saliency maps are shown in Section 4. These results are discussed and general remarks are made in Section 5. Finally, Section 6 concludes this paper with an overview of the contribution.

2 Related Work

In the following, a review of a number of important papers in the field of mixed traffic prediction, followed by a review of the most suitable saliency map extraction methods are presented. Finally, a multi-agent prospective of implementing this model is discussed.

2.1 Mixed Traffic Prediction Methods

In the topic of mixed traffic models, there are numerous available methods. Some depend on rule-based methods [14, 1], while others use black box deep

learning approach [2, 3, 5]. The latter category showed lower prediction errors in many traffic datasets [7, 8, 6]. However, this accuracy came at the cost of a more complex structure, bigger size of networks, and eventually lower explainability.

One of the first deep learning models with good accuracy is Social-LSTM [2]. This method uses a social pooling layer where the vectors of different Long Short-Term Memory (LSTM) networks (each one correspond to a nearby traffic participant) is taken to predict the agent future trajectory.

Other later work [3] employed Generative Adversarial Network (GAN) architectures. It depended on random sampling to generate multiple plausible trajectories. Variational Auto-encoders Architecture (VAE) was used as well [15] where a random sampling step is also performed. However, in these two cases (GAN and VAE) the output is dependent on random factor which cannot be explained.

Recently the work in [16], used Inverse Reinforcement Learning network as a first part to predict the coarse goal and trajectory, and then other recurrent networks with attention layers were used to get the finer trajectory.

In order to implement a deep learning model for mixed traffic modelling and simulation with less complexity and with memoryless architecture, a direct architecture of successive Convolutional Neural Network (CNN) layers is adopted here. This will make the generated saliency maps more meaningful. Otherwise, if an architecture with LSTM (which contains a memory) or with GAN (where a random sampling step is done) is used, the saliency maps will be affected by other random or previous input values, not just the current agent input.

Some other previous methods did use similar architecture of successive CNN layers. One example is [17], where the input is a group of RGB images and the output is flattened to form a group of vectors representing multiple plausible paths. Another paper is Y-Net [5], where a U-Net architecture is used to predict the pedestrians trajectories and goals, with an input consisting of the scene image and the previous trajectory of the agent. However, the work here, has a different shape of output and input, targeted to make the process of generating the output more transparent.

2.2 Saliency Maps Extraction Methods

For the network architecture proposed here, a number of suitable methods for saliency maps extraction are reviewed in the following.

Attention maps [18] are heatmaps generated using an additional layer added at the beginning of the network, and trained with it. This layer is multiplied with its input, in inference time, in order to maximize the effect of the important areas while minimizing the effect of other areas in the input.

The Visual Backprop [13] method depends on the usage of the most influencing information found in the higher layers, then by alternating between a deconvolution operation and element-wise multiplication with the activation matrix of the previous layer, moving from the output towards the input, it can link the output with the input as shown in Figure 1, taken from [13]. The end result is a grayscale image of the most salient parts of the input.

Recently the work in [12], called attentional bottleneck, proposes the usage of an attention layer while fusing the input into a smaller layer where the data is forced into smaller size making the heatmap focuse more on the most salient parts of the input.

All of the previous methods can generate visual explanations in the form of heatmaps. In this work, the method of visual backprob is selected to generate the maps, due to its suitability to the architecture. For instance, it was implemented with the similar architecture of PilotNet [19] .

In the field of traffic prediction, previous attempts to generate visual explanations were done. One example, is the paper in [4] where an attention layer is used to get saliency maps. The result highlighted the traversability parts of the scene. Another work [11] used a Layer-wise Relevance Propagation (LRP) [20] method to assign a percentage to each agent in the scene representing its influence on the prediction output. These two methods reported some benefits of applying post-hoc explanation methods to the traffic prediction models.

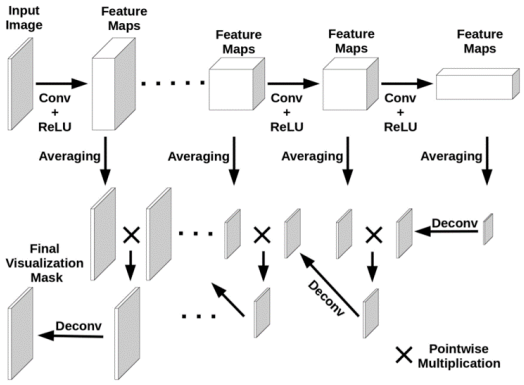


Fig. 1. VisualBackProp explanation method workflow (taken from [13] page 4703)

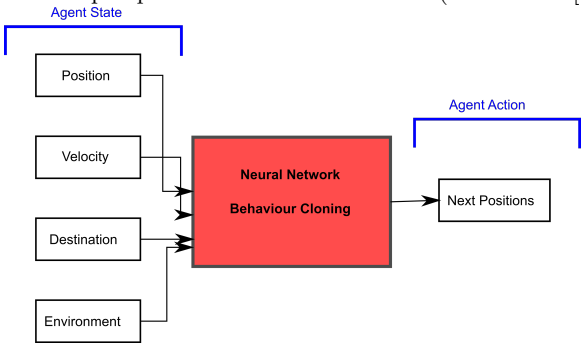


Fig. 2. Overview of the behaviour cloning approach for traffic modelling

2.3 Multi-agent Perspective of Mixed Traffic Modelling

The problem of mixed traffic modelling is a multi-agent problem by nature, where different agent types should be represented by different types of models. Additionally, each model should be customizable in order to represent the different agent attributes on microscopic level.

In the work here, the goal is to mimic the single traffic agent behavior, by presenting the problem as states to actions mapping. All the traffic types (pedestrians, cyclists and cars) are using the same model which has a multi-heads output, as shown in Figure 3, so each output branch can become specialized in a particular traffic type. Therefore, in principle, the model can be used in multi-agent simulation software such as SUMO [21], or M3 [22] to drive the agents, but each output branch should be trained only on particular agent type.

3 Method

An overview of the proposed method of using behavior cloning in the problem of microscopic mixed traffic modelling is shown in Figure 2

Behavior cloning is a supervised learning method implemented within MDP settings, where an agent model is supposed to imitate the behaviour of the expert (usually human) using the expert demonstrations as the training data [23].

The state is represented in several top-view images of the last eight step of the agent. Position, velocity, environment and destination are represented in those images.

The network architecture, as shown in Figure 3, consists of successive CNN layers with relu activation functions but softmax activation function for the last layer. The input is several RGB and grayscale images, specifically, one RGB image at the start, six grayscale images in the middle and a last RGB image at the end. They form together the single three dimensional input array.

The input is taken as bird's eye view from Stanford Drone Dataset [6]. The agent is always at the same position in the images coordinates, and it is color-coded as shown in Figure 4 at the right. The green box is a pedestrian, the blue box is a cyclist and the red box is a vehicle.

The standard prediction and trajectory history times for this dataset, as used in many other works [24, 4, 5], are 3600 and 2400 milliseconds respectively.

As shown also in Figure 3, the input step 8 image contains a yellow dot representing an approximate future position of the agent, i.e. the destination. If the destination fell outside the image borders, then it will not be added. This means that the model is trained on mixed examples, some with destination and others without.

The output has twelve grayscale images, representing the agent probability of being in a given position for the respective twelve steps. This is similar to the output format used in [10]. In other words, the (x,y) coordinate of the predicted future position is encoded in 2D array as a peak value.

The nature of the trajectory prediction problem is not deterministic [5]. This means that there isn't one possible correct prediction but many plausible outputs. To represent this feature, different works used different approaches, for example with generative architecture [3], variational auto-encoders [15], or multi head architecture [17].

In this work a multi-head output is implemented, as shown in Figure 3. The exact number of output paths for Stanford drone dataset is either 20 or 5 based on the standard split used in other works [24].

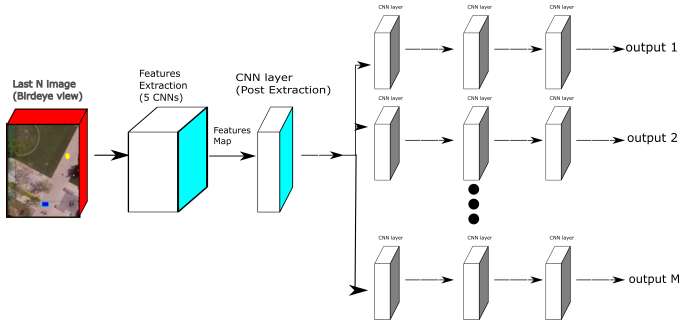


Fig. 3. Network architecture

After the training phase, the predictions are calculated from the model for the test split of the dataset. Simultaneously the explanatory saliency maps are generated for each output using the visual backprop method.

Small modifications are done here on top of visual backprop method. Namely, for each activation layer and before performing the multiplication operation, the layer's values are normalized to the range (0-1), and a histogram equalization step is performed on the last heatmap corresponding to the input.

4 Experimental Evaluation

The result of training the network for single, 5 and 20 modes on Stanford Dataset is shown in Table 1 where the errors are in pixels, namely the Average Displacement Error (ADE), and Final Displacement Error (FDE).

The mode is the term used in many other papers [17, 25, 16] to indicate the number of plausible paths in the model output generated for a single agent state.

The saliency maps examples were chosen according to two criteria, in order to show a representative set of examples. First criterion, the ADE should be lower than twelve pixels for all the examples. Second criterion, these examples should be crowded, and diverse with respect to the agents' types.

Table 1. The errors in pixels for different numbers of modes on the Stanford drone dataset along with the result of other papers

Case	ADE (pixel)	FDE (pixel)	Epochs of training
BC[1] - 1 mode	30	450	111
BC - 5 modes	21.99	211.78	30
BC - 20 modes	17.99	273.9	36
BC - 20 modes (no destination)	29.47	317.25	36
Social GAN [2] [3] (20 modes)	27.23	41.44	-
CF-VAE [15] (20 modes)	12.60	22.30	-
Y-Net [2] [5] (20 modes)	7.85	11.85	-

*1: Behaviour Cloning (Ours)

*2: it predicts only pedestrians movement

The real and synthetic examples are shown in Figures (4 to 15), where the black path represents the ground truth, the pink paths represent the predictions, and the blue path represent the past trajectory.

Each figure from the dataset contains five images. First image on the left is the last input image. The next one is the saliency map from the model, and the third is the result of element-wise multiplying the first two images. The fourth and fifth images are the ground truth trajectory and the closest predicted waypoints to it respectively, represented as small white boxes in the path.

The following examples are taken from the test set of the dataset and presented here along with their saliency maps corresponding to the last input image and its associated prediction in Figures 4 to 12.

Figures 4 to 7 belong to the same traffic situation, but with output for networks of 20, 5 and single modes as well as 20 modes without destination, respectively. The most accurate output is for the 20 modes with destination as shown in Figure 4, where the cyclists are avoided and blackened out in the saliency maps, but a path very close to the ground truth is taken.

The five-modes network predicted a path near the ground truth also, as shown in figure 5 and the same is true for the one-mode output. Without destination, the network also excludes the area next to the cyclists.

In Figures 8 and 9, the pedestrian wants to cross between two cyclists. The saliency maps shows a thin line between the pedestrian and the destination in both of the two networks. However, in both of them there is a jump in the best output prediction which is clearly due to the closeness of the passing by cyclists.

Figures 10 and 11 are for two separate examples. Here it is clear also that not only the nearby cyclists' and pedestrians' rectangles are blackened out, but also the road in front of them where they may pass in the future.

Fig. 4. Example 1 - 20 modes output

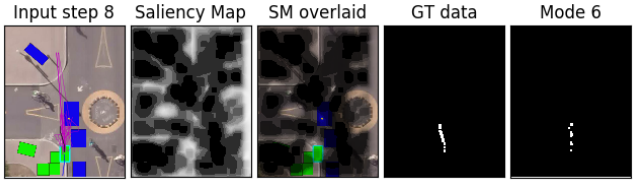


Fig. 5. Example 1 - 5 modes output

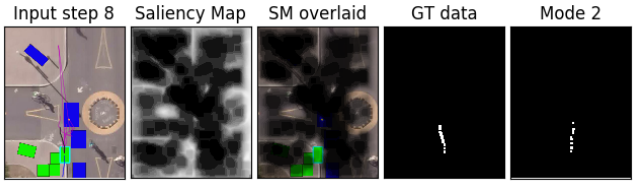


Fig. 6. Example 1 - single mode output

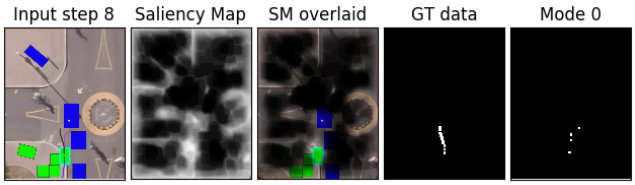


Fig. 7. Example 1 - 20 modes output without destination

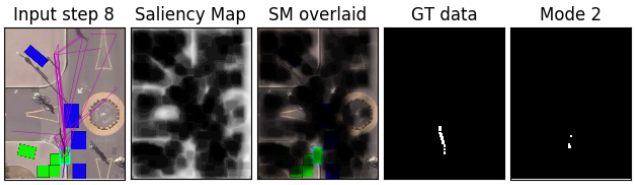


Fig. 8. Example 2 - 20 modes output

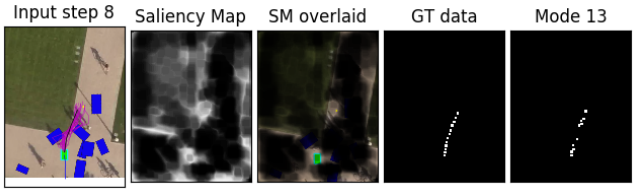


Fig. 9. Example 2 - 5 modes output

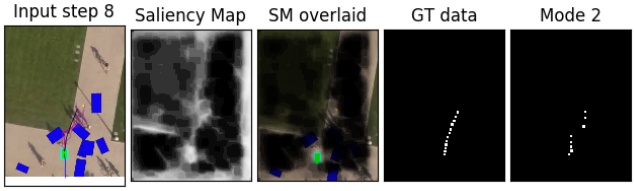


Fig. 10. Example 3 - 20 modes output

Fig. 11. Example 4 - 20 modes output

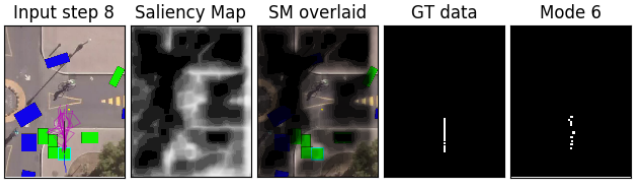


Fig. 12. Example 5 - 20 modes without destination

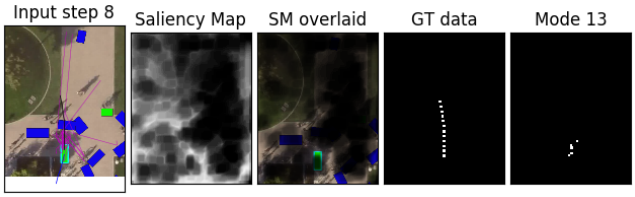


Figure 12 shows that even without destination the output is in the direction of the ground truth, and as the saliency maps shows, big black areas are in the way because of the cyclists.

Figure 13 is a synthetic example, where three pedestrians try to cross the road and a fast car has just passed them. The prediction is a jump to the sidewalk, and a few points suggest a movement to the left towards the destination. Here the sidewalk is highlighted along with the car and the pedestrians rectangles. Figure 14 also shows a focus on the sidewalk as well as further movement in the direction of the sidewalk. Figure 15, the model correctly avoided the cyclists and tried to take a turn behind the obstacle to the destination. The agent, the obstacle and the destination are all highlighted.

5 Discussion

The error values were slightly higher for the model here than some other methods shown in Table 1, even with destination. However, the model accuracy is still reasonable, for example, in the case of 20 modes with destination, the ADE error value is around 18 pixels which is in most cases less than the width of the agent's bounding box. Additionally, the extraction of meaningful saliency maps should be achievable with this level of accuracy. The network does the prediction by detecting patterns of agents positions and environment structure

Fig. 13. Example 6 - synthetic example 20 modes

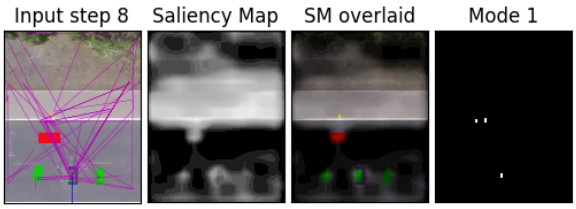


Fig. 14. Example 7 - synthetic example 5 modes

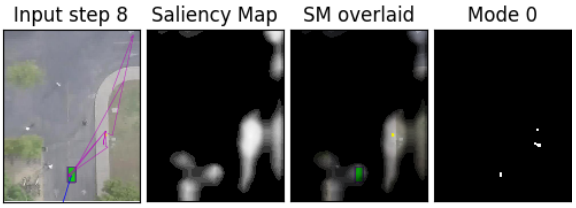
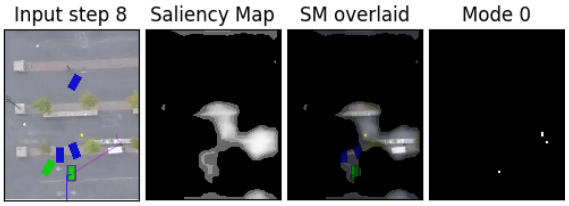


Fig. 15. Example 8 - synthetic example single mode



in the input images. These patterns should be highlighted in the saliency maps, and by analyzing these maps, it is noted that:

- They highlight the agent of interest, the yellow dot of destination and the edges of the road and sidewalks
- They blacken out the other faster agents in the vicinity of the main agent and the area in their movement direction.

In general, it is easy for the network to detect the agent, because of its main color and position. It is also easy to detect the yellow destination point. The next step for the network is finding the correct path towards the destination. If the road is clear, it will be a straight path and it will be highlighted in the map. However, if other agents, like Figure 8, or some obstacles like Figure 15, were in the way, then a maneuver should be done, and the salience map will blacken out these parts of agents or obstacles.

The edge of the road are important features to learn for the model, for example in Figure 8, it is noted that the edge is tilted, the prediction path is also parallel to that, because it's more likely to walk in the direction of, or perpendicular to, the road direction than in other ways. This is also shown in Figure 4, when the pedestrian wants to cross the road.

6 Conclusion

In this work, a model for mixed traffic modelling is trained and tested on Stanford Drone Dataset for three agent types (cars, cyclists and pedestrians). This model formulated the problem in MDP framework, mapping states to actions using behavior cloning model trained with supervised learning from traffic data.

A post-hoc explanation method was used to get explanatory saliency maps for the predictions. These maps showed that the model attended to the agent, the destination and the possible areas to plan the predicted path, while avoiding collision with other agents and obstacles. The edges of the roads and sidewalks, which define the general direction of movement were also highlighted in consistency with the predictions.

Therefore, these maps can provide more information of why some area was avoided and why some direction was taken. These information is useful in the case of a simulation or in the case of automatic analysis of real-world dataset.

As a future work, a method for assigning each output branch of the model to specific traffic participant type should be investigated in order to use the model to drive different agents in a multi-agent based simulation framework.

References

1. Schönauer, R.: A microscopic traffic flow model for shared space. Graz University of Technology (2017)
2. Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L., Savarese, S.: Social lstm: Human trajectory prediction in crowded spaces. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2016) 961–971
3. Gupta, A., Johnson, J., Fei-Fei, L., Savarese, S., Alahi, A.: Social gan: Socially acceptable trajectories with generative adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2018) 2255–2264
4. Sadeghian, A., Kosaraju, V., Sadeghian, A., Hirose, N., Rezatofighi, H., Savarese, S.: Sophie: An attentive gan for predicting paths compliant to social and physical constraints. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2019) 1349–1358
5. Mangalam, K., An, Y., Girase, H., Malik, J.: From goals, waypoints & paths to long term human trajectory forecasting. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. (2021) 15233–15242
6. Robicquet, A., Sadeghian, A., Alahi, A., Savarese, S.: Learning social etiquette: Human trajectory understanding in crowded scenes. In: European conference on computer vision, Springer (2016) 549–565
7. Bock, J., Krajewski, R., Moers, T., Runde, S., Vater, L., Eckstein, L.: The ind dataset: A drone dataset of naturalistic road user trajectories at german intersections. In: 2020 IEEE Intelligent Vehicles Symposium (IV). (2020) 1929–1934
8. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. In: CVPR. (2020)
9. Zablocki, É., Ben-Younes, H., Pérez, P., Cord, M.: Explainability of vision-based autonomous driving systems: Review and challenges. arXiv preprint arXiv:2101.05307 (2021)

10. Bansal, M., Krizhevsky, A., Ogale, A.: Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst. arXiv preprint arXiv:1812.03079 (2018)
11. Kothari, P., Kreiss, S., Alahi, A.: Human trajectory forecasting in crowds: A deep learning perspective. IEEE Transactions on Intelligent Transportation Systems (2021)
12. Kim, J., Bansal, M.: Attentional bottleneck: Towards an interpretable deep driving network. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. (2020) 322–323
13. Bojarski, M., Choromanska, A., Choromanski, K., Firner, B., Ackel, L.J., Muller, U., Yeres, P., Zieba, K.: Visualbackprop: Efficient visualization of cnns for autonomous driving. In: 2018 IEEE International Conference on Robotics and Automation (ICRA), IEEE (2018) 4701–4708
14. Johora, F.T., Müller, J.P.: Modeling interactions of multimodal road users in shared spaces. In: 2018 21st International Conference on Intelligent Transportation Systems (ITSC), IEEE (2018) 3568–3574
15. Bhattacharyya, A., Hanselmann, M., Fritz, M., Schiele, B., Straehle, C.N.: Conditional flow variational autoencoders for structured sequence prediction. arXiv preprint arXiv:1908.09008 (2019)
16. Deo, N., Trivedi, M.M.: Trajectory forecasts in unknown environments conditioned on grid-based plans. arXiv preprint arXiv:2001.00735 (2020)
17. Cui, H., Radosavljevic, V., Chou, F.C., Lin, T.H., Nguyen, T., Huang, T.K., Schneider, J., Djuric, N.: Multimodal trajectory predictions for autonomous driving using deep convolutional networks. In: 2019 International Conference on Robotics and Automation (ICRA), IEEE (2019) 2090–2096
18. Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., Bengio, Y.: Show, attend and tell: Neural image caption generation with visual attention. In: International conference on machine learning, PMLR (2015) 2048–2057
19. Bojarski, M., Yeres, P., Choromanska, A., Choromanski, K., Firner, B., Jackel, L., Muller, U.: Explaining how a deep neural network trained with end-to-end learning steers a car. arXiv preprint arXiv:1704.07911 (2017)
20. Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.R., Samek, W.: On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PloS one **10**(7) (2015) e0130140
21. Lopez, P.A., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flötteröd, Y.P., Hilbrich, R., Lücken, L., Rummel, J., Wagner, P., Wießner, E.: Microscopic traffic simulation using sumo. In: The 21st IEEE International Conference on Intelligent Transportation Systems, IEEE (2018)
22. Suzumura, T., Kanezashi, H.: Multi-modal traffic simulation platform on parallel and distributed systems. In: Proceedings of the Winter Simulation Conference 2014, IEEE (2014) 769–780
23. Osa, T., Pajarinen, J., Neumann, G., Bagnell, J.A., Abbeel, P., Peters, J.: An algorithmic perspective on imitation learning. arXiv preprint arXiv:1811.06711 (2018)
24. Sadeghian, A., Kosaraju, V., Gupta, A., Savarese, S., Alahi, A.: Trajnet: Towards a benchmark for human trajectory prediction. arXiv preprint (2018)
25. Mangalam, K., Girase, H., Agarwal, S., Lee, K.H., Adeli, E., Malik, J., Gaidon, A.: It is not the journey but the destination: Endpoint conditioned trajectory prediction. In: European Conference on Computer Vision, Springer (2020) 759–776