Social Attention for Autonomous Decision-Making in Dense Traffic

Edouard Leurent, Jean Mercat

Social Attention for Autonomous Decision-Making in Dense Traffic

Edouard Leurent[*]
SequeL team, INRIA Lille – Nord Europe
Renault Group, France
edouard.leurent@inria.fr

Jean Mercat[*]
Laboratoire des signaux et des systèmes, Centrale-Supélec
Renault Group, France
jean.mercat@renault.com

Abstract

We study the design of learning architectures for behavioural planning in a dense traffic setting. Such architectures should deal with a varying number of nearby vehicles, be invariant to the ordering chosen to describe them, while staying accurate and compact. We observe that the two most popular representations in the literature do not fit these criteria, and perform badly on an complex negotiation task. We propose an attention-based architecture that satisfies all these properties and explicitly accounts for the existing interactions between the traffic participants. We show that this architecture leads to significant performance gains, and is able to capture interactions patterns that can be visualised and qualitatively interpreted. Videos and code are available at https://eleurent.github.io/social-attention/.

1 Introduction

In the last decades, the problem of *behavioural planning* – that is, high-level decision-making in the context of autonomous driving – has arguably received less attention and seen less progress than the other components of the typical robotics pipeline: perception and control (González *et al.*, 2016). Indeed, the vast majority of existing systems still rely on hand-crafted rules encoded as Finite State Machines (Paden *et al.*, 2016). As a result, only a narrow set of specified use-cases are addressed and these methods cannot scale to more complex scenes, especially when the decision-making involves interacting with other human drivers whose behaviours are uncertain and difficult to model explicitly.

This observation has led the community to turn to learning-based methods, which bear the promise of leveraging data to automatically learn a complex driving policy. In the imitation learning approach, a policy can be trained in a supervised manner to imitate human driving decisions (e.g. Pomerleau, 1989; Ross *et al.*, 2011; Bojarski *et al.*, 2016; Xu *et al.*, 2016; Eraqi *et al.*, 2017; Codevilla *et al.*, 2018; Rehder *et al.*, 2018; Rezagholiradeh and Haidar, 2018; Rhinehart *et al.*, 2018; Bansal *et al.*, 2018; Rhinehart *et al.*, 2019). Because the cost of human driving data collection at large scale can be prohibitive, another promising approach is train a policy in simulation using reinforcement learning (e.g. Cardamone *et al.*, 2009; Ross *et al.*, 2011; Mukadam *et al.*, 2017; Chen *et al.*, 2017; Isele *et al.*, 2018; Ha and Schmidhuber, 2018; Kendall *et al.*, 2019).

[*]Equal contribution.

Machine Learning for Autonomous Driving Workshop at the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

Beyond the choice of reinforcement learning algorithm, the formalization of the problem as a Markov Decision Process plays an important part in the design of the system. Indeed, the definition of the state space involves choosing a representation of the driving scene. In this work, we focus in on how the vehicles are represented. In particular, we claim that the two most-widely used representations both suffer from different drawbacks: on the one hand, the *list of features* representation is compact and accurate but has a varying-size and depends on the choice of ordering. On the other hand, the *spatial grid* representation addresses these concerns but in return suffers from an accuracy-size trade-off.

Our contributions are the following: first, we propose an attention-based architecture for decision-making involving social interactions. This architecture allows to satisfy the variable-size and permutation invariance requirements even when using a *list of features* representation. It also naturally accounts for interactions between the ego-vehicle and any other traffic participant. Second, we evaluate our model on a challenging intersection-crossing task involving up to 15 vehicles perceived simultaneously. We show that our proposed method provides significant quantitative improvements, and that it enables to capture interaction patterns in a way that is visually interpretable.

2 Background and Related Work

Model-free deep reinforcement learning Reinforcement Learning is a general framework for sequential decision-making under uncertainty. It frames the learning objective as the optimal control of a Markov Decision Process (S, A, P, R, γ) with measurable state space S, action space A, unknown reward function $R \in \mathbb{R}^{S\times A}$, and unknown dynamics $P \in \mathcal{M}(S)^{S\times A}$, where $\mathcal{M}(\mathcal{X})$ denotes the probability measures over a set $\mathcal{X}$. The objective is to find a policy $\pi \in \mathcal{M}(A)^S$ with maximal expected γ-discounted cumulative reward, called the value function V^π. Formally,

$$V^\pi(s) \overset{\text{def}}{=} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \,\middle|\, s_0 = s, a_t \sim \pi(a_t|s_t), s_{t+1} \sim P(s_{t+1}|s_t, a_t)\right]$$
$$Q^\pi(s, a) \overset{\text{def}}{=} R(s, a) + \gamma \mathop{\mathbb{E}}_{s'\sim P(s'|s,a)} V^\pi(s')$$

The optimal action-value function $Q^* = \max_\pi Q^\pi(s)$ satisfies the Bellman Optimality Equation:

$$Q^*(s, a) = (\mathcal{T}Q^*)(s, a) \overset{\text{def}}{=} \mathop{\mathbb{E}}_{s'\sim P(s'|s,a)} \max_{a'\in A} [R(s, a) + \gamma Q^*(s', a')]$$

As Q^* is a fixed-point of the Bellman Operator $\mathcal{T}$ (Bellman, 1956) – which is a contraction –, it can be computed by applying $\mathcal{T}$ in a fixed-point iteration fashion. The *Q-learning* algorithm (Watkins and Dayan, 1992) follows this procedure by applying a sampling version $\mathcal{T}$ to a batch of collected experience. When dealing with a continuous state space S, we need to employ function approximation in order to generalise to nearby states. The *Deep Q-Network* (DQN) algorithm (Mnih *et al.*, 2015) implements this idea by using a neural network model to represent the action-value function Q.

State-representation for social interactions In order to apply a reinforcement learning algorithm such as DQN to an autonomous driving problem, a state space S must first be chosen, that is, a representation of the scene. When social interactions are relevant to the decision, the state should at least contain a description of every nearby vehicle. A vehicle driving on a road can be described in the most general way by it's continuous position, heading and velocity. Then, the joint state of a road traffic with one ego-vehicle denoted s_0 and N other vehicles can be described by a list of individual vehicle states:

$$s = (s_i)_{i\in[0,N]} \qquad \text{where} \qquad s_i = [x_i \quad y_i \quad v_i^x \quad v_i^y \quad \cos\psi_i \quad \sin\psi_i]^T \tag{1}$$

This representation, that we call *list of features*, is illustrated in Figure 1 (left) and was used for instance in (Bai *et al.*, 2015; Gindele *et al.*, 2015; Song *et al.*, 2016; Sunberg *et al.*, 2017; Paxton *et al.*, 2017; Galceran *et al.*, 2017; Chen *et al.*, 2017).

This encoding is efficient in the sense that it uses the smallest quantity of information necessary to represent the scene. However, it lacks two important properties. First, its size varies with the number

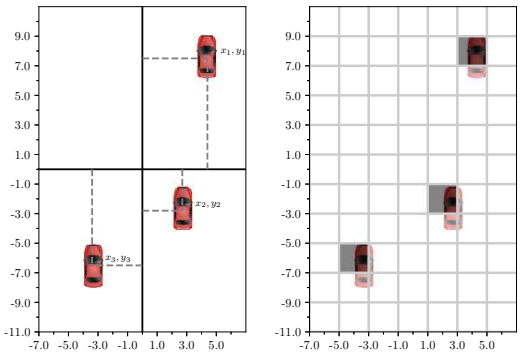

Figure 1: The *list of features* (left) and *spatial grid* (right) representations

of vehicles which can be problematic for the sake of function approximation which often expects constant-sized inputs. Second, we expect a driving policy π to be *permutation invariant*, i.e. not to be dependent on the order in which other traffic participants are listed. Ideally, this property should be enforced and not approximated by relying on the coverage of the $N!$ possible permutations τ of any given traffic state. Formally, we require that:

$$\pi(\cdot|(s_0, s_1, \dots, s_N)) = \pi(\cdot|(s_0, s_{\tau(1)}, \dots, s_{\tau(N)})) \qquad \forall \tau \in \mathfrak{S}_N \tag{2}$$

A popular way to address this limitations is to use a *spatial grid* representation. Instead of explicitly representing spatial information as variables x, y along with other features f directly inside a state $\{s_i = (x_i, y_i, f_i)\}_{i \in [0,N]}$ indexed on the vehicles, they are instead represented implicitly through the layout of several feature variables f_{ij} organised in a tensor structure, where the (i, j) indexes refer to a quantisation of the 2D-space. This representation is illustrated in Figure 1 (right). Note that the size of this tensor is related to the area covered divided by the quantisation step, which reflects a trade-off between accuracy and dimensionality. In an occupancy grid, the f features contains presence information (0-1) and additional channels such as velocity and heading, as in (e.g. Isele *et al.*, 2018; Fridman *et al.*, 2018; Bansal *et al.*, 2018; Rehder *et al.*, 2018). Another example is the use of top-view RGB images (e.g. Bagnell *et al.*, 2010; Rehder *et al.*, 2017, 2018; Liu *et al.*, 2018).

This permutation invariance property (2) can also be implemented within the architecture of the policy π. A general technique to achieve this is to treat each entity similarly in the early stages – e.g. through weight sharing – before reducing them with a projection operator that is itself invariant to permutations, for instance a max-pooling as in (Chen *et al.*, 2017) or an average as in (Qi *et al.*, 2016). A particular instance of this idea is attention mechanisms.

Attention mechanisms The attention architecture was introduced to enable neural networks to discover inter-dependencies within a variable number of inputs. It has been used for pedestrian trajectory forecasting in Vemula *et al.* (2018) with spatiotemporal graphs and in Sadeghian *et al.* (2019) with spatial and social attention using a generative neural network. In Sadeghian *et al.* (2018), attention over top-view road scene images for car trajectory forecasting is used. Multi-head attention mechanism has been developed in Vaswani *et al.* (2017) for sentence translation. In Messaoud *et al.* (2019) a mechanism called non-local multi-head attention is developed. However, this is a spatial attention that does not allow vehicle-to-vehicle attention. In the present work, we use a multi-head social attention mechanism to capture vehicle-to-ego dependencies and build varying input size and permutation invariance into the policy model.

3 Model Architecture

Out of a complex scene description, the model should be able to filter information and consider only what is relevant for decision. In other words, the agent should *pay attention* to vehicles that are close or conflict with the planned route.

The proposed architecture is presented in Figure 2. It is used to represent the Q-function that will be optimized by the DQN algorithm. It is composed of a first linear encoding layer whose

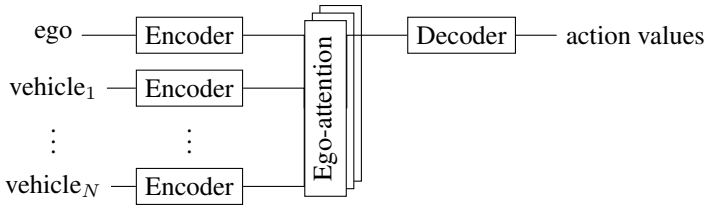

Figure 2: Block diagram of our model architecture. It is composed of several linear identical encoders, a stack of ego-attention heads, and a linear decoder.

weights are shared between all vehicles. At that point, the embeddings only contain individual features of size d_x. They are then fed to an ego-attention layer, composed of several heads stacked together. The *ego* prefix highlights that it similar to a multi-head self-attention layer (Vaswani *et al.*, 2017) but with only a single output corresponding to the ego-vehicle. Such an ego-attention head is illustrated in Figure 3 and works in the following way: in order to select a subset of vehicles depending on the context, the ego-vehicle first emits a single query $Q = [q_0] \in \mathbb{R}^{1\times d_k}$, computed with a linear projection $L_q \in \mathbb{R}^{d_x\times d_k}$ of its embedding. This query is then compared to a set of keys $K = [k_0, \dots, k_N] \in \mathbb{R}^{N\times d_k}$ containing descriptive features k_i for each vehicle, again computed with a shared linear projection $L_k \in \mathbb{R}^{d_x\times d_k}$. The similarity between the query q_0 and any key k_i is assessed by their dot product $q_0 k_i^T$. These similarities are then scaled by the inverse-square-root-dimension $1/\sqrt{d_k}$[2] and normalised with a softmax function σ across vehicles. We obtain a stochastic matrix called the *attention matrix*, which is finally used to gather a set of output value $V = [v_0, \dots, v_N]$, where each value v_i is a feature computed with a shared linear projection $L_v \in \mathbb{R}^{d_x\times d_v}$. Overall, the attention computation for each head can be written as:

$$\text{output} = \underbrace{\sigma\left(\frac{QK^T}{\sqrt{d_k}}\right)}_{\text{attention matrix}} V \tag{3}$$

The outputs from all heads are finally combined with a linear layer, and the resulting tensor is then added to the ego encoding as in residual networks. We can easily see that this process is permutation invariant: indeed, a permutation τ will change the order of the rows in keys K and values V in (3) but will keep their correspondence. The final result is a dot product of values and key-similarities, which is independent of the ordering.

4 Experiments

Environment In this application, we use the highway-env environment (Leurent, 2018) for simulated highway driving and behavioural decision-making. We propose a new task where vehicle-to-vehicle interaction plays a significant part: crossing a four-way intersection. The scene – composed of two roads crossing perpendicularly – is populated with several traffic participants initialised with random positions, velocities, and destinations. As in (Leurent, 2018), these vehicle are simulated with the Kinematic Bicycle Model, their lateral control is achieved by a low-level steering controller tracking a target route, and their longitudinal behaviour follows the Intelligent Driver Model (Treiber *et al.*, 2000). However, this model only considers same-lane interactions and special care was required to prevent lateral collisions at the intersection. To that end, we implemented the following simplistic behaviour: each vehicle predicts the future positions of its neighbours over a three-seconds horizon by using a constant velocity model. In case of predicted collision with a neighbour, the yielding vehicle is determined based on road priorities and brakes until the collision prediction ceases.

In this context, the agent must drive a vehicle by controlling its acceleration chosen from a finite set of actions $A = \{\texttt{SLOWER}, \texttt{NO-OP}, \texttt{FASTER}\}$. The lateral control is performed automatically by a low-level controller, such that the problem complexity is focused on the high-level interactions with other vehicles, namely the decision to either give or take way. The agent is rewarded by 1 when it drives at maximum velocity, 0 otherwise, and by -5 when a collision occurs.

[2]This scaling is due to the fact that the dot-product of two independent random vectors with mean 0, variance 1, and dimension d_k, is a random variable with mean 0 and variance d_k

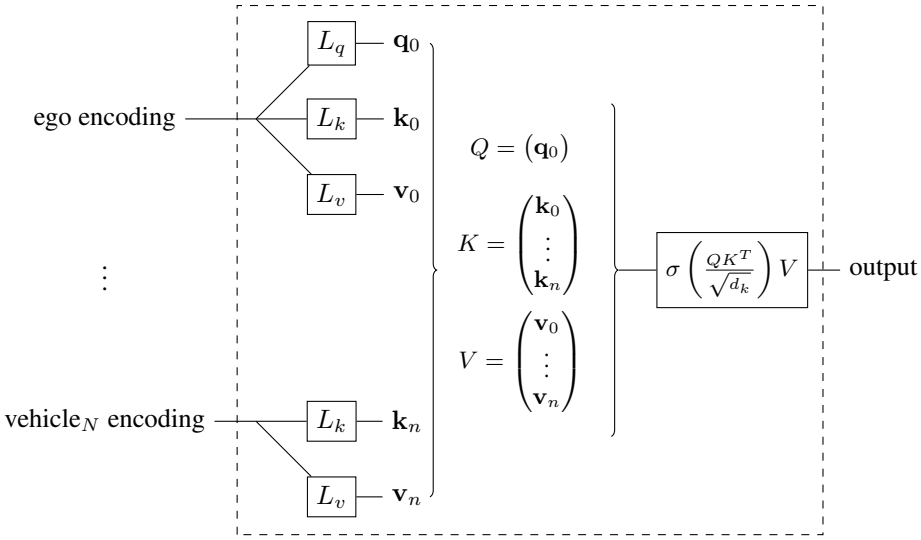

Figure 3: Architecture of an ego-attention head. The blocks L_q, L_k, L_v are linear layers. The keys K and values V are concatenated from all vehicles, while the query Q is only produced by the ego-vehicle.

Table 1: Characteristics of the agents

Architecture	`FCN/List`	`CNN/Grid`	`Ego-Attention`
Input sizes	[15, 7]	[32, 32, 7]	[$\cdot$, 7]
Layers sizes	[128, 128]	Convolutional layers: 3 Kernel Size: 2 Stride: 2 Head: [20]	Encoder: [64, 64] Attention: 2 heads $d_k = 32$ Decoder: [64, 64]
Number of parameters	3.0e4	3.2e4	3.4e4
Variable input size	No	No	Yes
Permutation invariant	No	Yes	Yes

Agents We evaluate three different agents, whose characteristics are summarised in Table 1.

- `FCN/List`: a *list of features* state representation is used, as described in Section 2. The model is a simple fully-connected network (FCN). Because this architecture requires a fixed-size input, we use zero-padding to fill the input tensor up to a maximum number $N = 14$ of observed vehicles, and add an additional *presence* feature to the coordinates described in (1) so as to identify active rows.
- `CNN/Grid`: a *spatial grid* representation is used, as described in Section 2, with a 32×32 grid where each cell represents a 2m $\times$ 2m square. The model is a convolutional neural network (CNN).
- `Ego-Attention`: a *list of features* state representation is used along with the Ego-Attention architecture described in Section 3. As this model supports varying-size inputs, zero-padding is not required.

These agents are all trained with the DQN algorithm using the same hyperparameters, and their architectures are scaled to admit about the same number of trainable parameters for fair comparison.

Performances We plot in Figure 4 the evolution of the total reward, episode length and average velocity during training, over 4000 episodes and repeated across 120 random seeds. The `FCN/List` agent learns to accelerate to earn short-term rewards, as shown by its high average velocity, but fails to exploit the information of other vehicles and crashes often, leading to short episodes. We obtain a risky and blind policy that is the worst performing. Conversely, the `CNN/Grid` architecture benefits

from its invariance to permutations and manages to learn to brake upon arrival at the intersection to avoid collisions, as we can see from its higher episode length. However, it only proceeds when the intersection has been fully cleared, as reflected by its low average velocity. This results in an overly cautious policy – a common trait colloquially known as the *freezing robot problem* (Trautman and Krause, 2010) – with a slight increase in performance. In stark contrast, the `Ego-Attention` policy quickly learns both when it must slow down at the intersection (see the high episode length), but also when it can exploit the gaps in the traffic and take way to vehicles that are far or slow enough (see the higher average velocity than `CNN/Grid`). This translates as a significant performance improvement, and the overall resulting behaviour is qualitatively more nuanced and human-like.

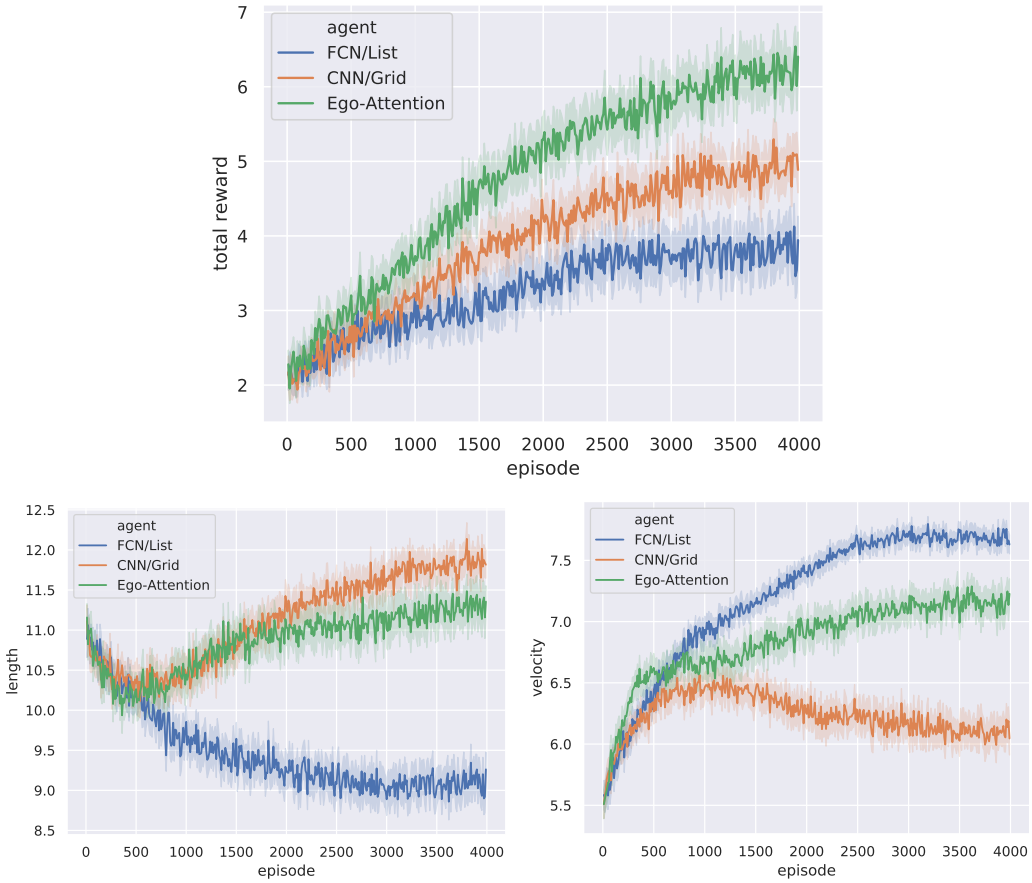

Figure 4: The episode total rewards, lengths, and average velocity (higher is better). We display the mean values – along with their 95% confidence interval – averaged over 120 random seeds.

Attention interpretation In any given state, the attention matrix can be visualised in the following way: we connect the ego-vehicle to every vehicle by a line of width proportional to the corresponding attention weight. Since the architecture can contain several ego-attention heads, we use different colours to distinguish them. In our experiments, two attention heads were used and will be represented in green and blue. We observe in Figure 5 that they specialised to focus on different areas: the green head is only watching the vehicles coming from the left, while the blue head restricts itself to vehicles in the front and right directions. However, we notice that both heads exhibit a common behaviour: they direct their attention to incoming vehicles that are likely to collide with the ego-vehicle, depending on their current position, heading, velocity, and ignore those that are too far or in a conflict-less situation. In particular, the attention tends to increase when vehicles get closer, as shown in Figure 6. It can also be very sensitive to small variations in the traffic state, as reflected in Figure 7. A full episode showcasing interactions with several vehicles is shown in Figure 8.

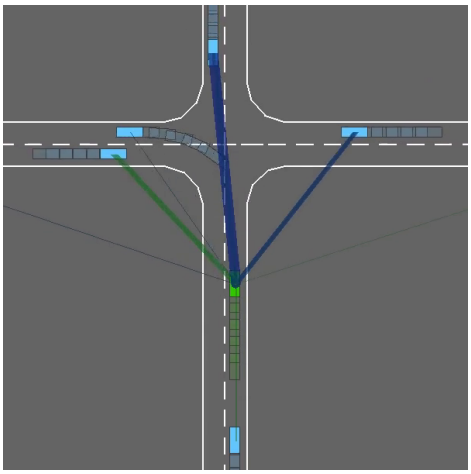

Figure 5: The attention heads specialised in different areas: left and front/right.

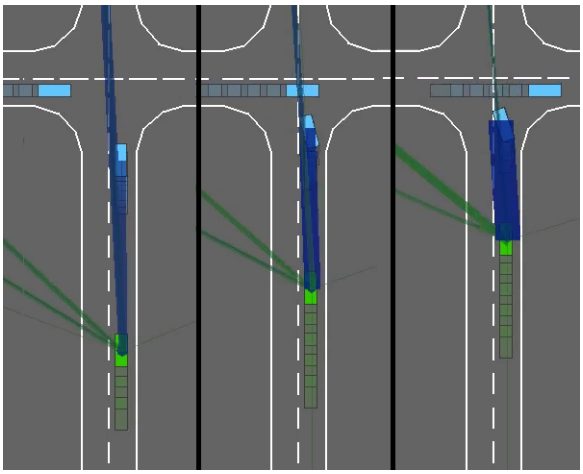

Figure 6: The attention paid to a vehicle tends to increase as it gets closer.

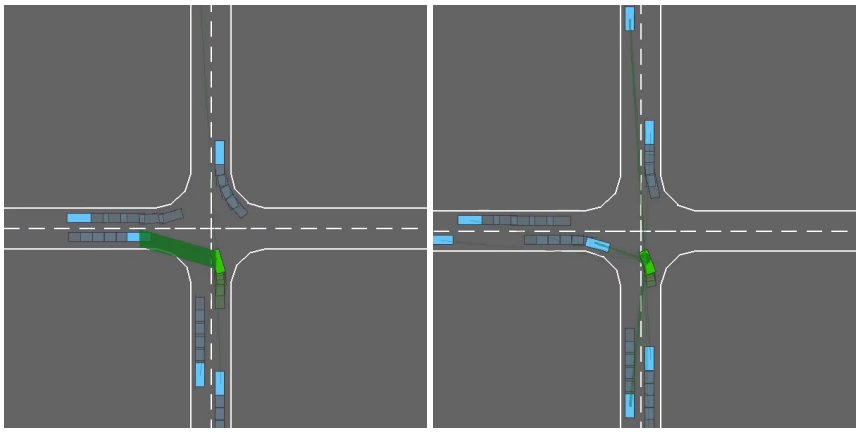

Figure 7: Sensitivity to uncertainty. *Left*: the agent has stopped at the intersection, its attention is focused on an incoming vehicle whose destination is still uncertain. *Right*: as soon as the vehicle orientation changes, revealing its intention of turning right, the attention drops and the agent starts accelerating right away.

Exploiting interaction patterns The agent decisions regarding right of way are not enforced through rewards but interactions: based on the defined road priorities, some vehicles will take way to the ego-vehicle while others will not. By changing which is a priority road, we can influence the rules of interactions which affects the learnt behaviour. In Figure 9, we compare two policies placed in the exact same initial state and observe how their decisions are affected by their internal model of how incoming vehicles interact with them. This difference showcases the ability of our proposed architecture to discover and exploit such interaction patterns.

Goal conditioning In the previous examples, we trained a policy tailored for left-turns only because it is the hardest direction with the most conflict points and the lowest priority level. Two individual policies tailored for right turns and driving straight can be trained as well, with similar results. Training a generic intersection policy would be less efficient without any prior information on where the ego-vehicle is headed. To remedy this problem, the destination could be added as additional features in (1), for instance encoded as a desired direction (d_x, d_y). This destination feature could also be used for other traffic participants to encode blinker information when available. This should result in a more efficient and generic policy.

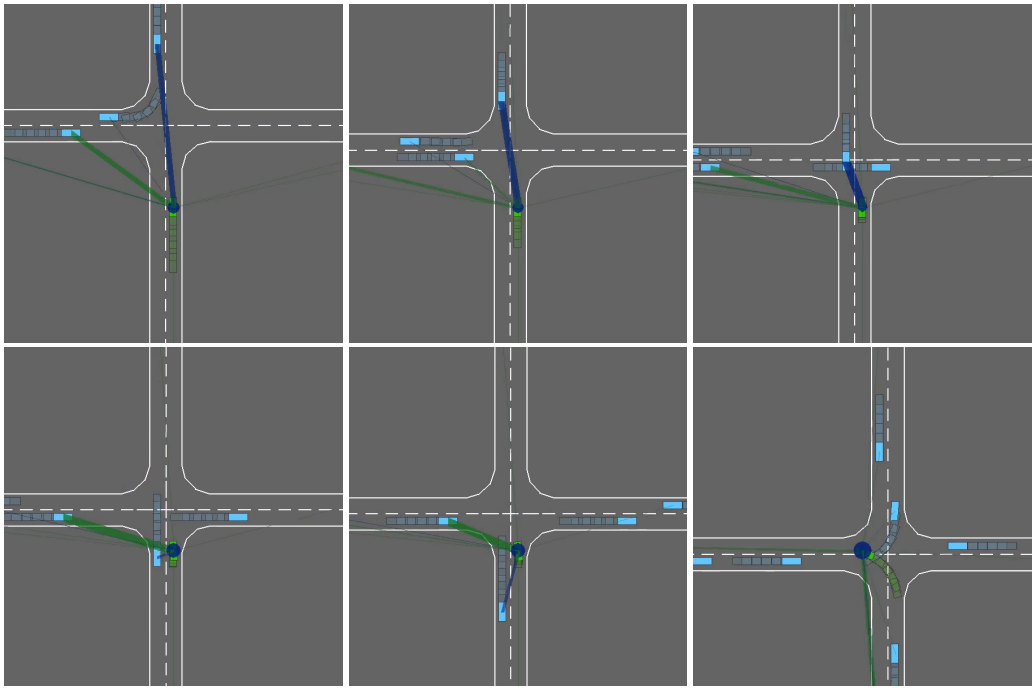

Figure 8: A complete episode. *From left to right, top to bottom*: 1. The green and blue heads direct their attentions to the left and front vehicles, respectively. 2. The left-vehicle is passing and is no longer a threat 3. Immediately, the green attention head switches to the next vehicle coming from the left. 4. The front vehicle has now passed, and the blue attention head is now focused on the ego-vehicle. 5. The ego-vehicle waits for one last vehicle coming from the left. 6. The ego-vehicle can finally proceed, and its attention is focused on itself.

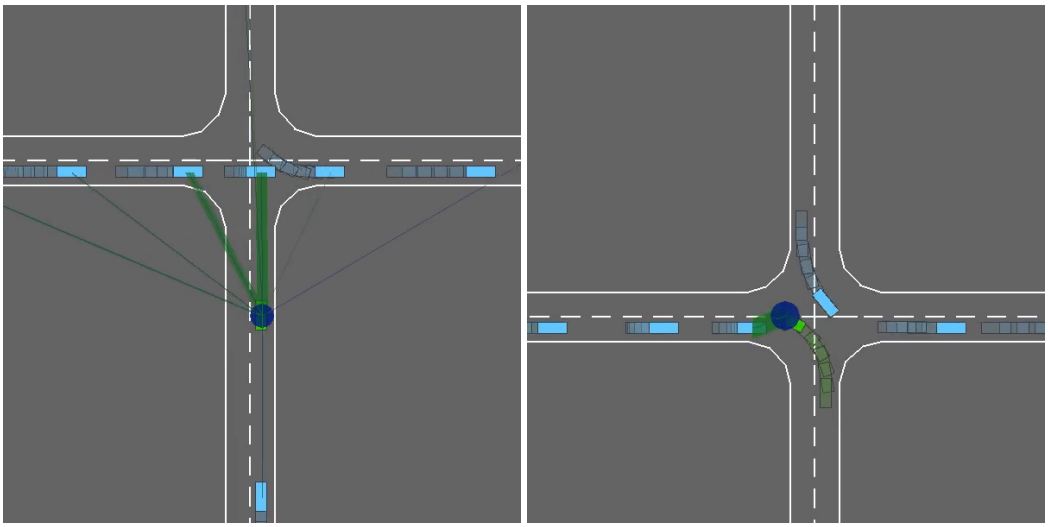

Figure 9: Effect of the right of way. *Left*: when trained on a non-priority road, the agent learns to yield to incoming vehicles. *Right*: when trained on a priority road, the agent expects other vehicles to give way and is consequently more aggressive.

5 Conclusion

In this work, we showed that the *list of features* representation, commonly used to describe vehicles in autonomous driving literature, is not tailored for use in a function approximation setting, in particular with neural networks. These concerns can be addressed by the *spatial grid* representation, but it comes at the price of an increased input size and loss of accuracy. In contrast, we proposed an attention-based neural network architecture to tackle the aforementioned issues of the *list of features* representation without compromising either size or accuracy. This architecture enjoys a better performance on a simulated negotiation and intersection crossing task, and is also more interpretable thanks to the visualisation of the attention matrix. The resulting policy successfully learns to recognise and exploit the interaction patterns that govern the nearby traffic.

References

James Bagnell, David Bradley, David Silver, Boris Sofman, and Anthony Stentz. Learning for autonomous navigation. *IEEE Robotics and Automation Magazine*, 17(2):74–84, 2010.

Haoyu Bai, Shaojun Cai, Nan Ye, David Hsu, and Wee Lee. Intention-aware online pomdp planning for autonomous driving in a crowd. *Proceedings - IEEE International Conference on Robotics and Automation*, 2015:454–460, 06 2015.

Mayank Bansal, Alex Krizhevsky, and Abhijit Ogale. Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst, 2018.

Richard Bellman. Dynamic programming and lagrange multipliers. In *Proceedings of the National Academy of Sciences of the United States of America*, 1956.

Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D. Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, Xin Zhang, Jake Zhao, and Karol Zieba. End to End Learning for Self-Driving Cars. *arXiv preprint*, 2016.

Luigi Cardamone, Daniele Loiacono, and Pier Luca Lanzi. Evolving competitive car controllers for racing games with neuroevolution. In *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation*, GECCO '09, pages 1179–1186. ACM, 2009.

Yu Fan Chen, Michael Everett, Miao Liu, and Jonathan P. How. Socially aware motion planning with deep reinforcement learning. *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep 2017.

Felipe Codevilla, Matthias Miiller, Antonio Lopez, Vladlen Koltun, and Alexey Dosovitskiy. End-to-End Driving Via Conditional Imitation Learning. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 4693–4700, 2018.

Hesham M. Eraqi, Mohamed N. Moustafa, and Jens Honer. End-to-End Deep Learning for Steering Autonomous Vehicles Considering Temporal Dependencies. In *Workshop on Machine Learning for Intelligent Transportation Systems, NeurIPS 2017*, 2017.

Lex Fridman, Jack Terwilliger, and Benedikt Jenik. Deeptraffic: Crowdsourced hyperparameter tuning of deep reinforcement learning systems for multi-agent dense traffic navigation. In *Deep Reinforcement Learning Workshop at NeurIPS 2018*, 2018.

Enric Galceran, Alexander G. Cunningham, Ryan M. Eustice, and Edwin Olson. Multipolicy decision-making for autonomous driving via changepoint-based behavior prediction: Theory and experiment. *Autonomous Robots*, 2017.

Tobias Gindele, Sebastian Brechtel, and Rudiger Dillmann. Learning driver behavior models from traffic observations for decision making and planning. *IEEE Intelligent Transportation Systems Magazine*, 7(1):69–79, 2015.

David González, Joshué Pérez, Vicente Milanés, and Fawzi Nashashibi. A Review of Motion Planning Techniques for Automated Vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 2016.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 2450–2462. Curran Associates, Inc., 2018.

David Isele, Reza Rahimi, Akansel Cosgun, Kaushik Subramanian, and Kikuo Fujimura. Navigating occluded intersections with autonomous vehicles using deep reinforcement learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018.

Alex Kendall, Jeffrey Hawke, David Janz, Przemyslaw Mazur, Daniele Reda, John-Mark Allen, Vinh-Dieu Lam, Alex Bewley, and Amar Shah. Learning to drive in a day. *2019 International Conference on Robotics and Automation (ICRA)*, May 2019.

Edouard Leurent. An environment for autonomous driving decision-making. https://github.com/eleurent/highway-env, 2018.

Jingchu Liu, Pengfei Hou, Lisen Mu, Yinan Yu, and Chang Huang. Elements of Effective Deep Reinforcement Learning towards Tactical Driving Decision Making. *arXiv preprint*, 2018.

Kaouther Messaoud, Itheri Yahiaoui, Anne Verroust-Blondet, and Fawzi Nashashibi. Non-local Social Pooling for Vehicle Trajectory Prediction. In *IV 19 - IEEE Intelligent Vehicles Symposium 2019*, Paris, France, June 2019.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

Mustafa Mukadam, Akansel Cosgun, Alireza Nakhaei, and Kikuo Fujimura. Tactical decision making for lane changing with deep reinforcement learning. In *Workshop on Machine Learning for Intelligent Transportation Systems, NeurIPS 2017*, 12 2017.

Brian Paden, Michal Čáp, Sze Zheng Yong, Dmitry Yershov, and Emilio Frazzoli. A survey of motion planning and control techniques for self-driving urban vehicles. *IEEE Transactions on Intelligent Vehicles*, 1, 04 2016.

Chris Paxton, Vasumathi Raman, Gregory Hager, and Marin Kobilarov. Combining neural networks and tree search for task and motion planning in challenging environments. In *Proc. of IROS 2017*, pages 6059–6066, 09 2017.

Dean a Pomerleau. Alvinn: An autonomous land vehicle in a neural network. *Advances in Neural Information Processing Systems 1*, pages 305–313, 1989.

Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. *Proceedings - 2016 4th International Conference on 3D Vision, 3DV 2016*, pages 601–610, 2016.

Eike Rehder, Jannik Quehl, and Christoph Stiller. Driving Like a Human: Imitation Learning for Path Planning using Convolutional Neural Networks. In *Workshop at International Conference on Robotics and Automation*, 2017.

Eike Rehder, Florian Wirth, Martin Lauer, and Christoph Stiller. Pedestrian Prediction by Planning Using Deep Neural Networks. In *IEEE International Conference on Robotics and Automation*, pages 5903–5908, 2018.

M. Rezagholiradeh and M. A. Haidar. Reg-gan: Semi-supervised learning based on generative adversarial networks for regression. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2806–2810, April 2018.

Nicholas Rhinehart, Rowan McAllister, and Sergey Levine. Deep imitative models for flexible inference, planning, and control. *arXiv preprint arXiv:1810.06544*, 2018.

Nicholas Rhinehart, Rowan McAllister, Kris Kitani, and Sergey Levine. PRECOG: PREdiction Conditioned On Goals in Visual Multi-Agent Settings. *arXiv preprint arXiv:1905.01296*, May 2019.

Stephane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In Geoffrey Gordon, David Dunson, and Miroslav Dudík, editors, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pages 627–635, Fort Lauderdale, FL, USA, 11–13 Apr 2011. PMLR.

Amir Sadeghian, Ferdinand Legros, Maxime Voisin, Ricky Vesel, Alexandre Alahi, and Silvio Savarese. Car-net: Clairvoyant attentive recurrent network. In *The European Conference on Computer Vision (ECCV)*, September 2018.

Amir Sadeghian, Vineet Kosaraju, Ali Sadeghian, Noriaki Hirose, Hamid Rezatofighi, and Silvio Savarese. Sophie: An attentive gan for predicting paths compliant to social and physical constraints. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Weilong Song, Guangming Xiong, and Huiyan Chen. Intention-Aware Autonomous Driving Decision-Making in an Uncontrolled Intersection. *Mathematical Problems in Engineering*, 2016, 2016.

Zachary N. Sunberg, Christopher J. Ho, and Mykel J. Kochenderfer. The value of inferring the internal state of traffic participants for autonomous freeway driving. *Proceedings of the American Control Conference*, pages 3004–3010, 2017.

Peter Trautman and Andreas Krause. Unfreezing the robot: Navigation in dense, interacting crowds. *International Conference on Intelligent Robots and Systems, IROS 2010*, pages 797–803, 2010.

Martin Treiber, Ansgar Hennecke, and Dirk Helbing. Congested traffic states in empirical observations and microscopic simulations. *Physical Review E - Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics*, 62(2):1805–1824, 2000.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc., 2017.

Anirudh Vemula, Katharina Muelling, and Jean Oh. Social attention: Modeling attention in human crowds. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018.

Christopher J. C. H. Watkins and Peter Dayan. Q-learning. *Machine Learning*, 1992.

Huazhe Xu, Yang Gao, Fisher Yu, and Trevor Darrell. End-to-end Learning of Driving Models from Large-scale Video Datasets. *arXiv preprint*, 2016.