Smooth-Trajectron++

Augmenting the Trajectron++ Behaviour Prediction Model with Smooth Attention

Westerhout, Frederik S.B.; Schumann, Julian F.; Zgonnikov, Arkady

DOI
10.1109/ITSC57777.2023.10421838

Publication date
2023

Document Version
Final published version

Published in
Proceedings of the IEEE 26th International Conference on Intelligent Transportation Systems, ITSC 2023

2023 IEEE 26th International Conference on
Intelligent Transportation Systems (ITSC)
24-28 September 2023. Bilbao, Bizkaia, Spain

Smooth-Trajectron++: Augmenting the Trajectron++ behaviour prediction model with smooth attention

Frederik S.B. Westerhout
Cognitive Robotics
TU Delft
Delft, Netherlands
frederikwesterhout@gmail.com

Julian F. Schumann
Cognitive Robotics
TU Delft
Delft, Netherlands
j.f.schumannn@tudelft.nl

Arkady Zgonnikov
Cognitive Robotics
TU Delft
Delft, Netherlands
a.zgonnikov@tudelft.nl

2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC) | 979-8-3503-9946-2/23/$31.00 ©2023 IEEE | DOI: 10.1109/ITSC57777.2023.10421838

***Abstract*— Understanding traffic participants' behaviour is crucial for predicting their future trajectories, aiding in developing safe and reliable planning systems for autonomous vehicles. Integrating cognitive processes and machine learning models has shown promise in other domains but is lacking in the trajectory forecasting of multiple traffic agents in large-scale autonomous driving datasets. This work investigates the state-of-the-art trajectory forecasting model Trajectron++ which we enhance by incorporating a smoothing term in its attention module. This attention mechanism mimics human attention inspired by cognitive science research indicating limits to attention switching. We evaluate the performance of the resulting Smooth-Trajectron++ model and compare it to the original model on various benchmarks, revealing the potential of incorporating insights from human cognition into trajectory prediction models.**

I. Introduction

In a world where the demand for intelligent vehicles is increasing rapidly [1], the concern for the safety of passengers and other road users should grow with it [2]. According to the World Health Organization, approximately 1.3 million people die each year due to road traffic accidents, and this number is expected to increase if proper measures are not taken [3]. Therefore, it should be paramount for future autonomous vehicles to improve traffic safety.

One of the most critical factors for ensuring a safe environment around intelligent vehicles is accurately predicting the future movements of surrounding traffic participants. These predictions allow for a better assessment of the environment and anticipation of potentially dangerous situations at an early stage, lowering the risk of accidents. The accurate predictions of interactive behaviours are especially important, as those comprise the most challenging situations.

Numerous methods have been used to tackle the human behaviour prediction problem [4]–[6], with examples ranging from reasoning-based methods to data-driven techniques. Over the last few years, data-driven approaches have shown great potential [7]–[13], using machine learning algorithms to learn from large amounts of data to predict the trajectories of traffic participants. One of these data-driven models is *Trajectron++* [10], which stands out due to its public code availability, the general applicability and the results it achieved on multiple datasets (including *nuScenes* [14], and *highD* [15]).

Other methods to predict future behaviour of traffic participants are based on theories from cognitive science. Instead of learning merely from data, the model is constructed to mimic human cognition. One class of such models based on the concept of evidence accumulation [16] have proved useful specifically in predicting binary decisions in traffic interactions [17], [18]. However, these models are not yet applicable to trajectory forecasting in a more general setting. Another example of using insights from cognitive science for behaviour prediction in traffic is the use of a quantum-like Bayesian model, a mathematical framework that combines elements of quantum theory and Bayesian probability theory to describe decision-making and information processing in complex and uncertain environments [19]. It is used in [20] to more accurately predict human street crossing behaviour, compared to the more data-driven model *Social-LSTM* [7]. Yet another insight from cognitive science suggests that the brain has a limited capacity for shifting attention rapidly between different tasks [21]. This is used in [22], where the application of the smoothing term to the attention module of a machine-learning prediction model – referred to as *Smooth-Attention* – which mimics human cognition, allows for better predictive performance.

Recent work demonstrated that integrating insights from cognitive science is a promising way of improving the performance of trajectory prediction models, but such cognitively inspired models need to be explored in a much more comprehensive way. Specifically, Cao et al. [22] emphasize the need to combine smooth attention with more advanced interaction modelling network architectures. Here we aim to address this challenge by applying smooth attention to a state-of-the-art behaviour prediction model. Namely, we aim to improve upon the performance of *Trajectron++ (T++)* [10] by leveraging the method of *smooth attention* proposed

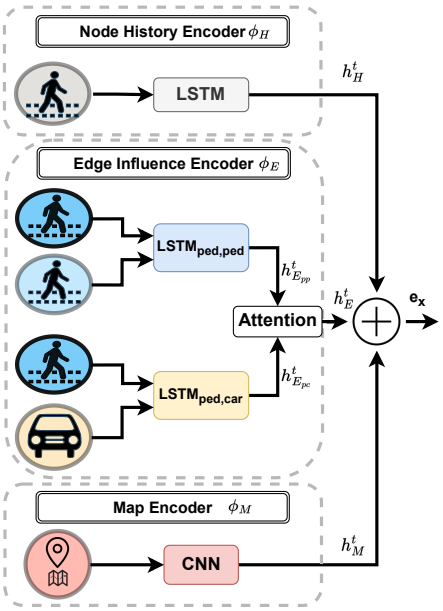

Fig. 1. Encoder part of *Trajectron++* [10] that encodes various past input information into the representation vector e_x.

in [22]. Applying a smoothness constraint on the attention module significantly reduces changes in attention, which mimics human cognitive processing. We name our approach of this combined model *Smooth-Trajectron++*. We test this new model on the *nuScenes* [14] and *highD* [15] datasets.

II. Methods

In this section, we provide an overview of the various elements of the *Trajectron++* model (*T++*) as well as the functioning of the *Smooth-Attention* module.

A. Trajectron++

We use *Trajectron++* [10] as the baseline model, for several reasons. Firstly, the model showed state-of-the-art performance on various public datasets [10] while including an attention module. Secondly, the authors have made the source code publicly available, including proper documentation regarding its application. This offers the opportunity to potentially reproduce the originally reported results while minimizing deviations from the original setup used by the authors. Lastly, the model has been tested on a large-scale public autonomous driving dataset, *nuScenes* [14]. This provides evidence of the applicability of the model to real-world scenarios concerning interactions between multiple traffic participants.

Trajectron++ is a graph-based conditional variational autoencoder model comprising an encoder and a decoder. The encoder uses various modules representing different influences on the trajectory forecast Figure 1. First, the past location and speed of the chosen traffic agent with multiple input time steps $(x_1^0, x_1^1, ...)$ are fed into the Node History Encoder ϕ_H, whose main component is a long-short-term memory cell (LSTM). This ensures that the past positional data is used for future predictions. The output of the LSTM is the hidden state h_H^t. Secondly, the *T++* model makes use of road map data to make its predictions more feasible and realistic. The Map Encoder module ϕ_M takes relevant environmental information, which is fed to a convolutional neural network (CNN), which has the hidden state h_M^t as output. Thirdly, the Edge Influence Encoder ϕ_E is used for taking into account to what extent other traffic agents in the scene are influencing the prediction. This module contributes to the "social awareness", which means that the behaviour of the agent gets influenced by other traffic participants. To encode graph edges, the method follows a two-step process. Firstly, edge information is collected from neighbouring agents belonging to the same semantic class, such as pedestrian-pedestrian and car-car semantic classes. Summation is used for feature aggregation inside these classes to handle varying numbers of neighbouring nodes while preserving count information, following [23]. The encodings of all the connections between the modelled agent and its neighbours are combined to create an "influence" representation vector, which represents the overall impact of the neighbouring nodes, which is done using an additive attention module [24]. Finally, the output is concatenated with the node's history and road map data to produce a unified node representation vector e_x fed to a decoder that constructs a predicted trajectory.

B. Smooth Attention

The *smooth attention* approach [22] presents a novel perspective on attention modules. Unlike traditional methods, it applies attention at each time step, following [25]. Emulating human attention during deliberate tasks incorporates a smoothness constraint based on the hypothesis that attention does not frequently change over time. Previous research [21] shows that deliberate attention shifts are slower due to internal limitations. This implies that attention does not frequently fluctuate during driving, as it falls under intentional shifts. By incorporating the smoothness constraint, the *smooth attention* approach enhances the attention mechanism, improving the selection of important information while disregarding less relevant input variables and aligning better with the characteristics of human attention.

III. Smooth-Trajectron++

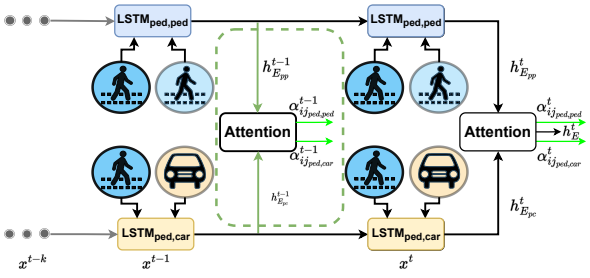

Fig. 2. Edge Influence Encoder including Smooth-Attention (in green). The model is trained to produce smooth attention values α using the loss function (1).

In this section, we propose a way to apply the smooth attention module [22] specifically to the *Trajectron++* model. To do this, we alter the Edge Influence Encoder

module of *Trajectron++* Figure 1, as this is the part where the social interactions are modelled, and the attention is applied.

Our approach[2], which we call *Smooth-Trajectron++*, is illustrated in Figure 2. At a high level, the original Edge Influence Encoder is expanded by applying the attention module at each time step in a similar fashion as in the *smooth attention* model (the green highlighted box in Figure 2), where the outputs $\alpha^{\tau}_{i,j_{ab}}$ are the attention weights that are used to rank the importance the human agent i assigns to the semantic class j_{ab} for neighbouring agents of types a and b (a and b can stand for agent types such as cars or pedestrians) at the time τ. All these attention weights from every time step are then used as an input for the added smoothing term in the loss function to incorporate the regularising of the attention by imposing a vectorial total variation penalty:

$$\mathcal{L}_{\text{smooth}}(\alpha) = \sum_{\tau=t-T+1}^{t} \sum_{i=1}^{N} \sqrt{\sum_{j} \left(\alpha^{\tau}_{ij} - \alpha^{\tau-1}_{ij}\right)^2}. \quad (1)$$

To ensure that the attention weights are utilised during the model training process, we incorporate $\mathcal{L}_{\text{smooth}}$ into the original loss function $\mathcal{L}_0$ [10]:

$$\mathcal{L}_{new} = \mathcal{L}_0 + \beta \mathcal{L}_{smooth}. \quad (2)$$

The scaling factor β is introduced to fine-tune the influence of $\mathcal{L}_{smooth}$. By adjusting β, the model can be trained to effectively balance the contribution of the attention weights with the original loss function. The β value of 0.01 is used as a starting point, following [22]. Additionally, this scaling factor is expanded up until β=10 to study its impact on performance. The intermediate steps are chosen logarithmic to cover a wide range of possibilities, with β=0.5 being an exemption to find an optimum between two promising values.

The extra loss term $\mathcal{L}_{smooth}$ and associated additional calls to the attention module increase the number of computations and therefore have an effect on the training time, which is approximately 1.5 times slower compared to the original version of *Trajectron++*.

IV. Results

Our method is evaluated on two publicly available datasets: *nuScenes* [14] and *highD* [15]. In both scenarios, we trained and assessed both the original *Trajectron++* model (which is a special case of *Smooth-Trajectron++* for $\beta = 0$) and the expanded model, with multiple versions of the latter, differentiated by five β-values ranging from 0.01 to 10. We also compared the obtained results with those originally reported for *T++* [10], as these turned out to be substantially different from the results we obtained after directly reproducing *T++* using the available source code and the same hyperparameters mentioned in the original article for the model's training.

[2]The source code is available at https://github.com/fwesterhout/Smooth-Trajectron

The experiments were performed on DelftBlue, the high-performance cluster of Delft University of Technology.

A. nuScenes dataset

The *nuScenes* dataset [14] consists of 1000 driving scenes in Boston and Singapore, characterised by their high traffic volumes and challenging driving situations. The driving scenes span 20 seconds each and are annotated at 2 Hz.

For this dataset, we evaluated the models according to three metrics: Final Displacement Error (FDE), Average Displacement Error (ADE) and Kernel Density Estimation of Negative Log Likelihood (KDE-NLL). These metrics are chosen as they were used in the original paper [10]. First, FDE indicates how far off the model's predicted location is from the actual location at the end of a predicted trajectory. Second, ADE is particularly useful for evaluating the overall accuracy of a model's trajectory predictions, as it considers the entire predicted trajectory rather than just the final location. Finally, KDE-NLL is a valuable metric for evaluating the uncertainty of a model's predictions, as it measures how well the model can capture the true distribution of the data. Following [10], we calculate the three above metrics at prediction horizons of 1, 2, 3, and 4s. The FDE and the ADE outputs comprise the most likely single trajectory prediction, using the "Most Likely" output configuration as in [10].

There are two main agent classes in *nuScenes*, pedestrians and vehicles. As their behaviour is significantly different, we evaluate the models on these classes separately.

1) Pedestrian-only predictions: Leftmost numbers in each column of Table I-Table III show the results for the predicted pedestrian trajectories. The numbers in bold represent the lowest metric values per prediction horizon, compared to the reproduced T++ results (the "T++ (rep)" row), which serve as a reference for comparative analysis.

First, we found a significant gap between the reproduced T++ performance and the results reported and *T++* paper. The FDE and ADE exhibit notable differences, especially at shorter prediction horizons. The reproduced KDE-NLL values also diverge significantly from the reported values. Several factors may contribute to this deviation; for example, a different version of *nuScenes*, a discrepancy in used and reported hyper-parameters and model settings, or possible deviations introduced during the reproduction process, such as data downloading or package installation. Future research should more closely examine the reproducibility of the original results and clarify potential causes of mismatches with the original findings.

Second, the *smooth attention* extensions of the reproduced *T++* ($\beta = 0.01$ to $\beta = 10$) consistently outperform the baseline reproduced version of *T++*. Tuning the scaling factor β influences the error. Regarding the FDE, the parameter $\beta = 0.1$ has the lowest error in all cases, except for the shared lowest error at the first prediction horizon (@1s) of $\beta = 1.0$. The higher the β-value, the more it resembles the "T++ (rep)" reference values. However, in Table III, the opposite

seems to be happening; the "T++ (rep)" row shows the lowest values for almost all cases. An exception is the smooth version with $\beta = 0.01$ @4s, where a marginal performance increase is seen. However, in general, in this pedestrian-only case, smooth attention does not improve this metric, although the decline for the smooth versions is minimal. The smoothing term might decrease the variety of predicted trajectory distributions, affecting the average and making it less similar to the ground truth. Further research is needed to explore this hypothesis in other pedestrian-only scenarios.

TABLE I
RESULTS *nuScenes* T++ PEDESTRIAN-ONLY/VEHICLE-ONLY: FDE (M)

Model	@1s	@2s	@3s	@4s
T++ [10]	0.014/0.07	0.17/0.45	0.37/1.14	0.62/2.20
T++ (rep.)	0.168/0.430	0.369/1.168	0.608/2.323	0.886/3.868
$\beta = 0.01$	0.157/**0.413**	0.353/1.102	0.586/2.141	0.855/3.546
$\beta = 0.1$	**0.155**/0.419	**0.350/1.081**	**0.580/2.122**	**0.842/3.496**
$\beta = 0.5$	0.159/0.421	0.354/1.123	0.588/2.181	0.857/3.560
$\beta = 1.0$	**0.155**/0.448	0.351/1.128	0.582/2.165	0.845/3.507
$\beta = 10$	0.160/0.425	0.366/1.149	0.607/2.190	0.876/3.539

TABLE II
RESULTS *nuScenes* T++ PEDESTRIAN-ONLY/VEHICLE-ONLY: ADE (M)

Model	@1s	@2s	@3s	@4s
T++ [10]	0.021/-	0.073/-	0.15/-	0.25/-
T++ (rep)	0.126/0.307	0.221/0.632	0.329/1.092	0.450/1.689
$\beta = 0.01$	0.116/**0.296**	0.208/0.602	0.314/1.021	0.432/1.559
$\beta = 0.1$	**0.114/0.302**	**0.206/0.597**	**0.311/1.012**	**0.427/0.543**
$\beta = 0.5$	0.118/0.301	0.210/0.613	0.316/1.041	0.434/1.580
$\beta = 1.0$	**0.114**/0.319	0.207/0.630	0.312/1.048	0.428/1.575
$\beta = 10$	0.118/0.303	0.215/0.628	0.325/1.055	0.445/1.586

TABLE III
RESULTS *nuScenes* T++ PEDESTRIAN-ONLY/VEHICLE-ONLY: KDE NLL

Model	@1s	@2s	@3s	@4s
T++ [10]	-5.58/-4.17	-3.96/-2.74	-2.77/-1.62	-1.89/-0.71
T++ (rep)	**-2.575**/-1.760	**-1.530**/-0.604	**-0.797**/0.235	-0.230/0.927
$\beta = 0.01$	-2.560/-1.861	-1.519/**-0.726**	-0.795/**0.108**	**-0.240/0.801**
$\beta = 0.1$	-2.541/-1.856	-1.502/-0.679	-0.776/0.176	-0.216/0.875
$\beta = 0.5$	-2.542/**-1.885**	-1.505/-0.690	-0.779/0.150	-0.211/0.818
$\beta = 1.0$	-2.549/-1.880	-1.507/-0.679	-0.785/0.173	-0.226/0.868
$\beta = 10$	-2.480/-1.861	-1.471/-0.661	-0.759/0.173	-0.205/0.845

2) Vehicle-only predictions: Rightmost numbers in each column of Table I-Table III show the results for the predicted vehicle trajectories. Similarly to the previous pedestrian-only case, a general FDE and ADE decline is seen along the β-versions of the *Smooth-Trajectron++*. The ADE values of the *T++* paper are missing in Table II, as they are not reported by the authors in the original article. The version with $\beta = 0.01$ holds the lowest value for the prediction horizon of 1 second, while $\beta = 0.1$ has a minor error for the remaining prediction horizons. Contrary to the pedestrian-only predictions in Table III, *Smooth-Trajectron++* on the vehicle-only forecasts has better KDE-NLL numbers than the reproduced model, which indicates that the model is better able to match the original distribution of predicted trajectories with the inclusion of the smooth-attention term in the loss function. Furthermore, this can be said for all β-factors, while in this case, the $\beta = 0.01$ has the lowest values.

B. highD dataset

To evaluate the models on the *highD* dataset, we used the previously proposed benchmarking framework [26]. This framework was designed to benchmark prediction models in *gap acceptance* scenarios, i.e. situations where drivers decide whether to enter a gap in traffic or wait for the next opportunity, such as when a car approaches an intersection and decides whether to turn left immediately or wait for a break in oncoming traffic. In case of the highD dataset, we investigated the predictions of gap acceptance in lane-change decisions using a restricted version of highD (see [26] for details).

The framework [26] allowed us to use two methods of splitting the highD data into training and testing sets: the random split and the critical split. The first method randomly splits the data for testing and training. In contrast, the second method deliberately selects the most unusual behaviour for testing, such as accepting a very small gap or rejecting a large gap. This latter testing scenario, therefore, tests the model's ability to extrapolate to situations that lie outside its training distribution, which is generally considered to be a more difficult task [27]. Also, small accepted gaps can be regarded as safety-critical scenarios, which is especially important when developing safe and reliable prediction models.

Furthermore, the framework allowed us to test the models with varying number of input time steps(n_I) to study the input-dependability of the tested models; we used $n_I = 2$ and $n_I = 10$.

In addition to the metrics used for the *nuScenes* dataset, the gap acceptance benchmark includes an additional metric, the Area under the Receiver-Operator Curve (AUC), used to evaluate the performance of binary classification models (here between accepted and rejected gaps).

First, we analysed the performance of the models at predicting lane-change decisions at initial gaps, i.e. at the start of the interaction (Figure 3, top row). Here, only in the case of $\beta = 0.5$ there is a notable increase in AUC for the random split compared to $T+$ in both n_I-instances. For the critical split, almost all AUC-values are lower than $T+$, except for $\beta = 0.1$ and $\beta = 0.5$ at $n_I = 10$ where it is slightly higher. Generally, the β-term does not seem to increase the performance of the base model.

Second, we investigated models' predictions of lane changes in *highD* at the fixed-sized gaps, as defined in

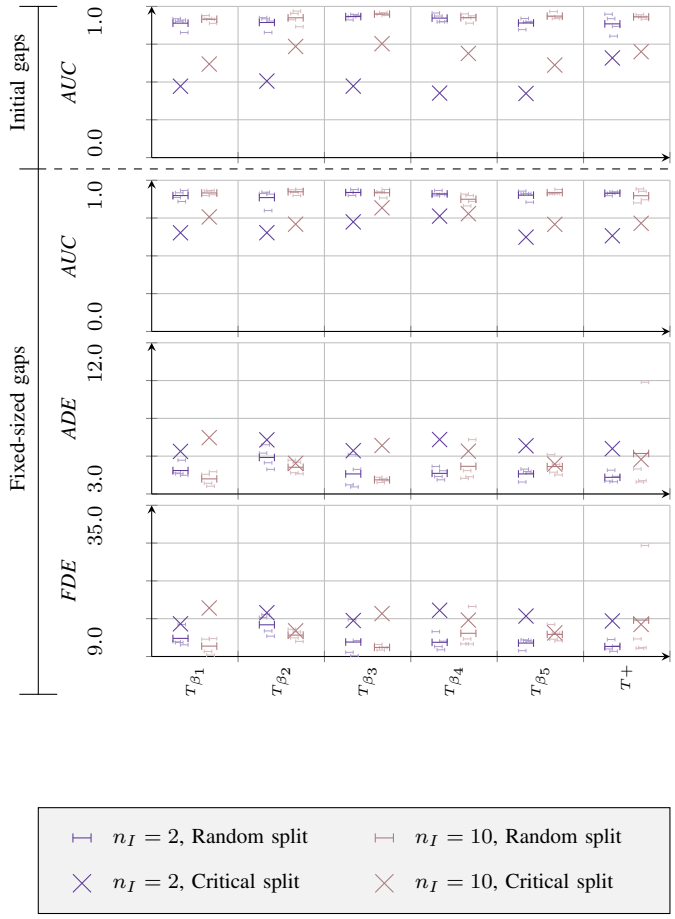

Fig. 3. The performance of *Smooth-Trajectron++* and *T++* on the *highD* dataset: β_{1-5} refer to the β-values of 0.01, 0.1, 0.5, 1.0 and 10 respectively

[26] (Figure 3, bottom three rows). Looking at the random splits, again for $\beta = 0.5$ there is an increase in AUC for both n_I-situations. The changes for the other β-versions are not consistently different when compared to $T++$, having minor fluctuations to perform slightly better or slightly worse. Concerning the critical split, all β-versions but $\beta = 10$ perform better than the base model, where $\beta = 10$ performs very similarly to $T++$. Also, the difference between $n_I = 2$ and $n_I = 10$ is logical, as the latter consistently has a higher AUC than the former.

For both the FDE and ADE, all the β-versions are outperforming the $T++$-model at $n_I = 10$ on the random split. However, this seems due to one extremely high value of one of the random splits of $T++$ (each random split consists of three sub-splits, which are averaged to minimize the effect of randomness). This could be an outlier, caused by an error in the training process. At $n_I = 2$, the β-values under-perform compared to $T++$ for the random split, indicating no significant improvement. At the critical split, only at $\beta = 0.1$ and $\beta = 10$ both ADE and FDE values are lower at $n_I = 10$. In general, there is no clear improvement regarding these metrics across the various β-values.

Overall, in *highD* lane-change prediction experiments, there are instances of both better and worse performance of *Smooth-Trajectron++* compared to T++, indicating no consistent benefits of adding smooth attention to T++. This is in contrast to the nuScenes results, which may stem from fundamental differences in the datasets. While the *nuScenes* dataset encompasses a wide range of data with cars and pedestrians, *highD* mainly focuses on cars. The application of the smoothing term β in the *Smooth-Trajectron++* model relies on the attention module that compares different semantic classes of traffic participants. In datasets where one class dominates, the smoothing term may not yield tangible improvements.

V. Conclusion

This paper proposed *Smooth-Trajectron++*, a trajectory prediction model based on an existing state-of-the-art model *Trajectron++* [10] in which we incorporated a cognitively-inspired smooth attention module [22]. We demonstrated that our smooth-attention version of T++ can achieve increased performance on the large-scale dataset nuScenes, but does not result in tangible improvements on the highD dataset. This suggests that the smooth attention approach seems to be more suitable for large-scale multi-agent datasets with multiple agent types rather than on datasets with few traffic agents of mostly the same type. Hence, the concept of *smooth attention* might be better applied to models where the attention module is implemented over individual agents and not semantic classes. Nevertheless, our results further strengthen previous work [18], [20], [22], indicating that including cognitive insights can allow better predictions of human behaviour in traffic.

References

[1] A. Singh and S. Mutreja, "Autonomous vehicle market size, share, value, report, growth."

[2] P. Koopman and M. Wagner, "Autonomous vehicle safety: An interdisciplinary challenge," *IEEE ITSM*, vol. 9, no. 1, pp. 90–96, 2017.

[3] World Health Organization, "Road traffic injuries." https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries, 2021. Accessed on May 12, 2023.

[4] F. Camara, N. Bellotto, S. Cosar, F. Weber, D. Nathanael, M. Althoff, J. Wu, J. Ruenz, A. Dietrich, G. Markkula, *et al.*, "Pedestrian models for autonomous driving part ii: high-level models of human behavior," *IEEE Trans. on Intell. Transp. Syst.*, vol. 22, no. 9, pp. 5453–5472, 2020.

[5] A. Bighashdel and G. Dubbelman, "A survey on path prediction techniques for vulnerable road users: From traditional to deep-learning approaches," in *2019 IEEE ITSC*, pp. 1039–1046, 2019.

[6] A. Rudenko, L. Palmieri, M. Herman, K. M. Kitani, D. M. Gavrila, and K. O. Arras, "Human Motion Trajectory Prediction: A Survey," *The Int. J. Robotics Research*, May 2019.

[7] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social lstm: Human trajectory prediction in crowded spaces," in *Proc. IEEE conference on computer vision pattern recognition*, pp. 961–971, 2016.

[8] A. Mohamed, K. Qian, M. Elhoseiny, and C. Claudel, "Social-stgcnn: A social spatio-temporal graph convolutional neural network for human trajectory prediction," in *Proc. IEEE/CVF conference on computer vision pattern recognition*, pp. 14424–14432, 2020.

[9] F. Giuliari, I. Hasan, M. Cristani, and F. Galasso, "Transformer networks for trajectory forecasting," in *2020 25th ICPR*, pp. 10335–10342, 2020.

[10] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone, "Trajectron++: Dynamically-Feasible Trajectory Forecasting with Heterogeneous Data," in *ECCV 2020*, pp. 683–700, 2020.

[11] Y. Yuan, X. Weng, Y. Ou, and K. M. Kitani, "Agentformer: Agent-aware transformers for socio-temporal multi-agent forecasting," in *Proc. IEEE/CVF*, pp. 9813–9823, 2021.

[12] A. Kalatian and B. Farooq, "A context-aware pedestrian trajectory prediction framework for automated vehicles," *Transp. research part C: emerging technologies*, vol. 134, p. 103453, 2022.

[13] A. Rasouli, I. Kotseruba, and J. K. Tsotsos, "Are they going to cross? a benchmark dataset and baseline for pedestrian crosswalk behavior," in *2017 IEEE ICCVW*, pp. 206–213, 2017. ISSN: 2473-9944.

[14] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuScenes: A Multimodal Dataset for Autonomous Driving," in *2020 IEEE/CVF Conf. on Comput. Vis. Pattern Recognit. (CVPR)*, pp. 11618–11628, June 2020. ISSN: 2575-7075.

[15] R. Krajewski, J. Bock, L. Kloeker, and L. Eckstein, "The highd dataset: A drone dataset of naturalistic vehicle trajectories on german highways for validation of highly automated driving systems," in *2018 21st ITSC*, pp. 2118–2125, IEEE, 2018.

[16] J. I. Gold and M. N. Shadlen, "The neural basis of decision making," *Annu. Rev. Neurosci.*, vol. 30, pp. 535–574, 2007.

[17] A. Zgonnikov, D. Abbink, and G. Markkula, "Should I Stay or Should I Go? Cognitive Modeling of Left-Turn Gap Acceptance Decisions in Human Drivers," *Hum. Factors: The J. Hum. Factors Ergon. Soc.*, p. 001872082211445, Dec. 2022.

[18] J. F. Schumann, A. R. Srinivasan, J. Kober, G. Markkula, and A. Zgonnikov, "Using Models Based on Cognitive Theory to Predict Human Behavior in Traffic: A Case Study," May 2023. arXiv:2305.15187 [cs].

[19] C. A. P. Moreira, "Quantum probabilistic graphical models for cognition and decision," *D. Universidade de Lisboa*, 2017.

[20] Q. Song, W. Wang, W. Fu, Y. Sun, D. Wang, and Z. Gao, "Research on quantum cognition in autonomous driving," *Sci. reports*, vol. 12, no. 1, p. 300, 2022.

[21] J. M. Wolfe, G. A. Alvarez, and T. S. Horowitz, "Attention is fast but volition is slow," *Nature*, vol. 406, no. 6797, pp. 691–691, 2000.

[22] Z. Cao, E. Biyik, G. Rosman, and D. Sadigh, "Leveraging smooth attention prior for multi-agent trajectory prediction," in *2022 Int. Conf. on Robotics Autom. (ICRA)*, pp. 10723–10730, IEEE, 2022.

[23] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena, "Structural-rnn: Deep learning on spatio-temporal graphs," in *Proc. IEEE CVPR*, pp. 5308–5317, 2016.

[24] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[25] A. Vemula, K. Muelling, and J. Oh, "Social attention: Modeling attention in human crowds," in *2018 IEEE ICRA*, pp. 4601–4607, 2018.

[26] J. F. Schumann, J. Kober, and A. Zgonnikov, "Benchmarking behavior prediction models in gap acceptance scenarios," *IEEE Trans. on Intell. Veh.*, vol. 8, no. 3, pp. 2580–2591, 2023.

[27] E. Barnard and L. Wessels, "Extrapolation and interpolation in neural network classifiers," *IEEE Control Syst. Mag.*, vol. 12, no. 5, pp. 50–53, 1992.