Selective Temporal Fusion using Recurrent Attention for End-to-End Autonomous Driving

Andreas Bentzen Winje, Florian Wintel, Gabriel Hanssen Kiss, and Frank Lindseth

Department of Computer Science, Norwegian University of Science and Technology (NTNU), Høgskoleringen 1, 7034 Trondheim, Norway
{andreas.b.winje,florian.wintel,gabriel.kiss,frankl}@ntnu.no

Abstract. In end-to-end autonomous driving (E2E-AD), understanding the complex and dynamic environment of the driving scene is crucial. Temporal information supports this by extending perception beyond what is observable in a single frame. While some E2E-AD architectures, like TransFuser++, operate without temporal modeling, various methods for temporal fusion have been explored, from frame-stacking to memory-based methods and most recently, attention-based recurrent methods. However, existing recurrent attention methods lack a mechanism for forgetting information, distributing attention across all past features even when they are no longer relevant. In this paper, we present a recurrent attention-based temporal fusion module (TFM) with selective forgetting, designed as a drop-in extension for E2E-AD architectures. The TFM fuses current and past information using cross-attention, enabling temporal modeling with minimal impact on inference time, and allows for interpretable retention through attention weight visualization. We integrate a selection mechanism using a void token to allow selective forgetting of irrelevant past information. Applied to the TransFuser++ architecture, our method achieves a driving score of 83.69% on the closed-loop Bench2Drive benchmark and provides qualitative insights into how models retain past information. These results demonstrate its potential as a temporal extension to otherwise temporally unaware architectures.

Keywords: End-to-End Autonomous Driving · Recurrent Neural Networks · Imitation Learning · Attention · Temporal Processing · CARLA.

1 Introduction

End-to-end autonomous driving (E2E-AD) is a motion planning and control task in which a driving agent directly maps raw sensor observations to driving actions through a single learned model. E2E architectures have achieved promising improvements in navigating dynamic traffic interactions in simulated and real-world environments. Building on this, we focus on *temporal fusion* in E2E-AD, examining how models can integrate and utilize information over time.

A variety of temporal fusion strategies has been explored in recent years. In particular, streaming-based temporal fusion has emerged as a promising approach, where past information is recurrently integrated into the present feature

representation using the attention mechanism [18,15]. These methods treat historic features as a hidden state that is queried at each time step, allowing the model to condition current predictions on past observations. However, existing work lacks an explicit mechanism for discarding outdated or irrelevant information, which can lead to inefficient representations. In addition, E2E-AD models often lack interpretable outputs, which remains a key challenge in the field [6].

In this work, we extend a strong TransFuser++ [13] baseline agent with a streaming-based temporal fusion module. We then introduce a *selective forgetting* mechanism that allows the model to forget irrelevant or uninformative spatiotemporal features. By adding a *void token* to historic features, the model can learn to attend to this token, disregarding past information and leaving more room to use current spatial features. We visualize the attention weights of the temporal fusion module, demonstrating how the module retains information. Our contributions are as follows.

- We present a recurrent, attention-based temporal fusion module (TFM) and implement it in the TransFuser++ [13] architecture.
- Extending the TFM, we implement a simple selective forgetting mechanism by adding a void token to the historic tokens.
- We evaluate the impact of the TFM and selective forgetting on the closed-loop Bench2Drive benchmark [14] and provide qualitative visualizations of information retention behavior.

2 Related Work

Driving is a complex task, requiring agents to anticipate the movements of dynamic objects (vehicles, pedestrians, etc.) and hazards. Some hazards may only be apparent in earlier frames, such as a temporarily occluded pedestrian emerging from behind a vehicle [19]. While some E2E-AD architectures, like LAV [5] or TransFuser++ [13], do not explicitly model temporal information, Temporal fusion has been explored in various forms, aiming to expand the driving model's context by encoding information from multiple frames. A prevalent technique is frame-stacking, where consecutive frames or frame features are processed jointly, commonly either by 2D [20,21,4,3] or 3D [10,11] convolutions to provide short-term temporal context. However, frame-stacking operates on only a fixed set of temporal frames and is limited in capturing long-term dependencies.

Recent methods have explored stateful techniques, such as recurrent neural networks [12,9,16] and memory banks [19] to retain and fuse temporal context, allowing for, in theory, arbitrary context lengths. Most recently, attention-based recurrent modules have been proposed. Rao *et al.* [18] implement a *streaming-based* recurrent attention decoder that fuses spatial information with a historic feature that is iteratively copied from previous predictions [18]. DriveTransformer [15] applies a similar approach using attention recurrently [15].

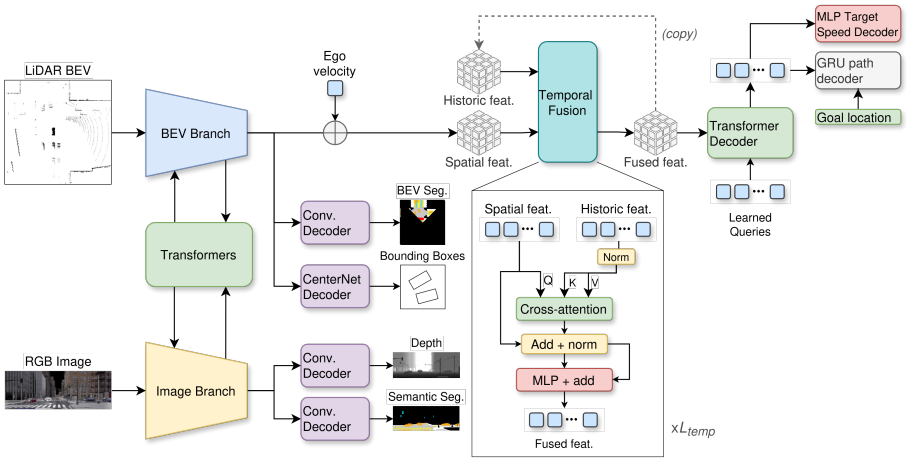


Fig. 1: **Extended TransFuser++**. We extend TF++ [13] by inserting a temporal fusion module between the backbone and the planning head.

3 Method

3.1 TransFuser++

Our method directly extends the popular imitation learning-based E2E-AD architecture TransFuser++ (TF++) [13] (which itself builds on TransFuser [7]). TF++ achieves competitive driving performance, with a driving score of 84.21% on the CARLA-based Bench2Drive benchmark [14,22]. The perception backbone of TF++ consists of two branches, which process RGB images (from three front-facing cameras) and LiDAR birds-eye-view (BEV) images (top-down representations of the scene around the ego vehicle, captured from a LiDAR depth sensor), respectively. The two branches exchange information through a transformer-based sensor fusion module. Each branch produces a spatial feature map used for perception outputs (bounding boxes, depth prediction, and semantic and BEV segmentation) and downstream planning. The subsequent planning module consumes the feature map of the BEV branch (BEV features) and predicts a future path and target speed. Finally, a PID-based control logic translates the plan into steering and acceleration commands. Our extension is inserted between the spatial backbone and planning heads of TF++. The temporal module captures spatiotemporal information by iteratively fusing the BEV features from current and historic time frames. The fused features are then passed to the planning head. The full extended TF++ architecture can be seen in fig. 1.

3.2 Implementation

Temporal Fusion Module. Our temporal fusion module (TFM) builds directly on streaming-based approaches [18,15]. We use a cross-attention layer

(CrossAttn) to integrate past and present information recurrently. At each time step t, the perception backbone produces a set of spatial features Q_t used as query tokens. These are fused with a set of historic features H_t acting as key-value (KV) tokens, which propagate through time and serve as a recurrent memory of past spatiotemporal context. The resulting feature tokens are then normalized with layer normalization (LayerNorm) and passed to a token-wise multilayer perceptron (MLP) to obtain the final fused spatiotemporal features F_t. Between each layer, we use residual connections to stabilize training. Formally, this can be expressed as

$$\begin{aligned} \hat{H}_t &= \text{LayerNorm}(H_t) \\ C_1 &= \text{CrossAttn}(Q = Q_t, KV = \hat{H}_t) \\ C_2 &= \text{LayerNorm}(C_1 + Q_t) \\ F_t &= \text{MLP}(C_2) + C_2, \end{aligned} \tag{1}$$

where C_1 and C_2 are intermediate calculations and $\hat{H}_t$ are the normalized historic features. Conceptually, H_t functions as a hidden state, where cross-attention is used as an update rule and provides the output to the downstream planning head at each time step. Between time steps, F_t is simply copied and reused as the next historic features, formally

$$H_{t+1} = F_t. \tag{2}$$

We avoid using self-attention to prevent mixing the query tokens from the backbone and maintain a spatial correspondence to the BEV grid. On the first time step ($t = 0$), where there is no past context yet, the historic tokens are initialized by a learnable tensor.

Selective Forgetting via Implicit Gating. While the default TFM enables recurrent integration of past features, it also forces each query to attend to some element of the historic tokens, regardless of whether any past context contains useful information. Due to normalization, the attention weights must sum to one, forcing each query to allocate its full attention mass across the KV-tokens, which can ultimately lead to inefficient use of the current context.

To address this, we introduce a mechanism for *selective forgetting* that allows the model to discard irrelevant historic context. We draw inspiration from the use of special tokens in transformer-based approaches, such as DETR's "no-object" token [2], to allow the model to ignore parts of the attention mass. Concretely, we augment the historic tokens H_t with an additional *void token*, a zero-vector appended to the KV set of the cross-attention, formally

$$H_t^+ = [H_t; \mathbf{0}].$$

During cross-attention, the model may route attention mass to the void token, effectively ignoring the past when it provides no useful information. The mechanism is shown in fig. 2. The original query information is preserved by the residual connection following the cross-attention layer shown in eq. (1), which

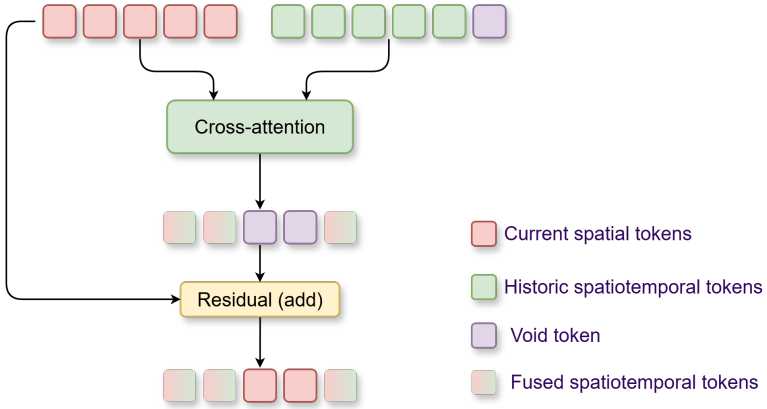


Fig. 2: **Selective forgetting mechanism**. A *void token* is added to the historic tokens to allow non-attention to the historic context. In essence, the void tokens erase past information when attended to, creating space for the current information, which is reintroduced through a residual connection.

re-adds the current features Q_t. Intuitively, the void token acts as an *implicit gating mechanism*. When queries attend to it, past information is forgotten, and the current spatial features are preserved. When queries attend to the historic tokens, past spatiotemporal information is retained. This implicit gating provides the model with flexibility in deciding what to remember or forget, without requiring explicit gate parameters as in recurrent networks like LSTMs or GRUs. It further allows for interpretable retention behavior via visualization of attention weights, allowing for inspection of the model's attention to the void token.

Efficient Training by Freezing in Time. To alleviate some of the computational burden of *backpropagation through time* [17], we take inspiration from previous works [16,15] where parts of the model are frozen during training. Specifically, we train only the TFM through time, and only backpropagate through the backbone on the last time step of each sequence. We find that this has minimal to no effect on performance while lowering VRAM usage and training time.

3.3 Training and Evaluation Details

Similar to TF++, a two-step training process is used. First, the backbone is trained alone using only the 4 perception outputs for 31 epochs, then the model is trained for another 31 epochs using the full architecture. The models are trained with a sequence length of 5 time steps, including the current time step, and a step size of 500 ms, leading to a total lookback of 2 s. Training one model takes about 4 days using 2 A100 40GB GPUs. The TFM uses multihead attention with

4 attention heads and $L_{temp} = 8$ fusion layers. For evaluation, we use ensembles of 3 trained models by averaging over model outputs [13]. We evaluate each ensemble 3 times and present the mean score. We train on the same data set as TF++ [13].

4 Experiments

We evaluate the performance of the temporal TransFuser++ architecture on the closed-loop Bench2Drive [14] benchmark, which uses the popular CARLA simulator [8]. Bench2Drive provides a comprehensive suite of driving scenarios, including various weather conditions, traffic densities, and complex urban environments. We report results on the original TF++ (reproduced), TF++ with our TFM, and finally TF++ with TFM and a void token. For comparison, we also list the original TF++ results from its technical report [22].

The key metric, driving score (DS) (%), is a composite of *route completion* (%) and an *infraction score* (0-1), based on how much of each route is completed and the number and type of infractions the model has committed (More information on the metrics can be found on the CARLA leaderboard [1] and in the Bench2Drive paper [14].).

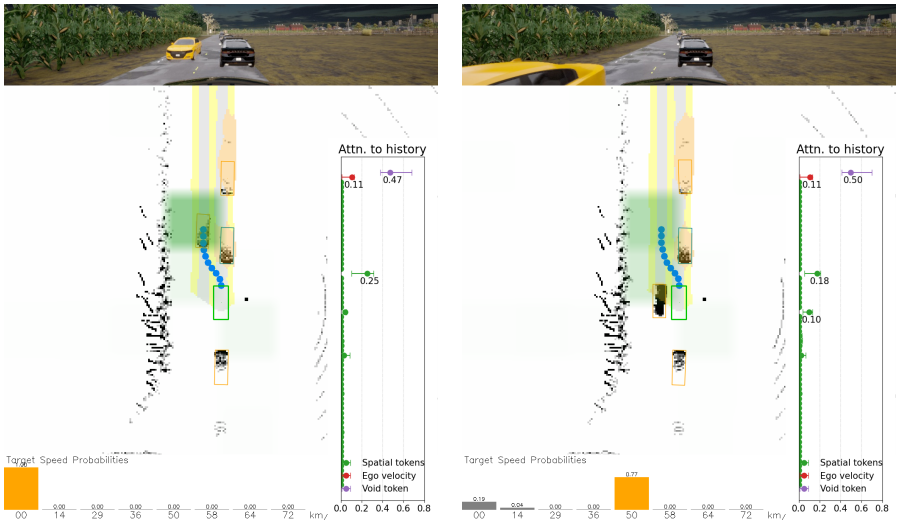
4.1 Bench2Drive Results

Table 1: **Bench2Drive benchmark scores.** We report the overall driving score (DS) and success rate (SR), in addition to the multi-ability scores merging (Merge), overtaking (Ovrtk), emergency brake (EmBrk), give way (GvWay), and traffic signs (TSign). (**Bold**=best score; Underline=second-best score).

Method	**Overall ↑**		**Multi-Ability↑**					
	DS	SR	Merge	Ovrtk	EmBrk	GvWay	TSign	Mean
`TF++` (reproduced)	82.71	63.94	56.67	<u>48.15</u>	75.56	<u>40.00</u>	**84.21**	60.92
`TF++` (w/`TFM`)	81.47	61.67	56.59	45.64	<u>80.56</u>	**50.00**	79.30	62.42
`TF++` (w/`TFM+void`)	<u>83.69</u>	<u>64.85</u>	**62.47**	42.97	**83.33**	**50.00**	<u>83.33</u>	**64.42**
`TF++` [22]	**84.21**	**67.27**	<u>58.75</u>	**57.77**	**83.33**	<u>40.00</u>	82.11	<u>64.39</u>

Table 1 shows the quantitative results on Bench2Drive [14] in the DS, SR, and multi-ability metrics. Although it remains slightly below the original TF++'s DS, the void model (`TFM+void`) outperforms the reproduced TF++ baseline by +0.98 DS. Notably, the void model performs better than or on par with the non-void TFM model in almost all metrics, including merging, traffic signs, and emergency braking, only falling behind in overtaking, and achieves a +2 higher mean multi-ability score. This demonstrates the void token's positive effect on the agent's driving performance.

4.2 Visualizing Temporal Attention



(a) Ego waiting for approaching vehicle. (b) Ego starts driving after vehicle passed.

Fig. 3: **Output visualizations**. Two outputs from the model showing an RGB image (top), LiDAR BEV (mid), target speed classifications (bottom), and attention to history (right). The attention to history plot shows the attention mass distribution across the historic spatial tokens (green), ego velocity token (red), and void token (purple). We overlay the BEV grid with a green tint according to the average attention of the corresponding spatial token.

To gain further insight into how our method affects the model, we visualize the attention weights of the TFM in fig. 3. The two images show two frames of a scenario, where the ego vehicle waits to pass a parked car in its lane while an oncoming car (yellow) approaches. The 'Attention to history'-plot shows the min, max, and mean total attention received by the historic tokens from the queries across all attention heads and layers, allowing for observable retention behavior. As can be seen from the spatial tokens (green), mainly two tokens are attended to, which, from the green overlay, can be seen to be the spatial regions for which the oncoming car is located. One can also see a high attention to both the ego velocity (11%) and the void token (≈50%). This was observed across multiple scenarios, with the void token consuming about 50-80% of all attention. We further plot the maximum void attention for each time step across the same frames in fig. 4. We observe a generally high void attention until the pass-by event, with a peak at the pass-by, followed by an immediate drop. We interpret this as the model choosing to forget information about the oncoming vehicle as it passes by, as, intuitively, it is no longer relevant to the motion plan.

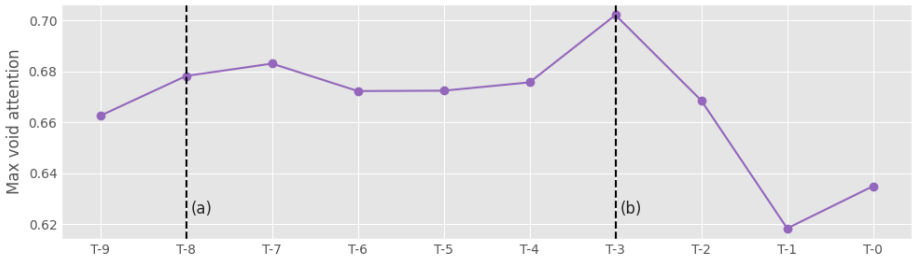


Fig. 4: **Void attention over time.** The maximum attention to the void token across ten frames. The corresponding frames from fig. 3 are marked (a) and (b).

Fig. 5: **Void attention by context.**

Finally, fig. 5 illustrates the effect of temporal context on the attention paid to the void token. In dense, moving traffic (top), void attention is lower (i.e., temporal context matters more) than when the road is free (bottom).

5 Discussion

Our experiments show that our proposed TFM-based agent achieves comparable driving performance to the non-temporal TransFuser++, with a 1% improvement in DS on the reproduced baseline. One interpretation is that the temporal context is not crucial to the model for many of the benchmarked scenarios; another is that it was still unable to exploit the information effectively. Visual inspection of the attention weights indicates the former: Particularly, the context-dependent spikes in void attention (e.g. on pass-by events) align intuitively with expected behavior: Less attention is paid to temporal context, when it is less important (compare figs. 3 and 5). This pattern was observed in multiple driving scenarios, suggesting that irrelevant information was in fact selectively forgotten.

Across all scenarios, void attention remained high (around 50-80%), with the model instead relying on current spatial tokens. This indicates that the historic tokens were largely uninformative to the model. Under this interpretation, the number of historic tokens could be reduced, e.g., by adaptively selecting the most important historic tokens using a top-K measure similar to [15].

Limitations. In some cases, the model remains stationary after a full stop. This behavior is not seen in the unaltered TF++ model and could indicate causal confusion, as it is reminiscent of one of the signs outlined in [6]. The model also consistently attended to the ego velocity token, suggesting an over-reliance on its own speed, further supporting this concern. Future work should investigate the impact of appending the ego velocity before and after temporal fusion to rule it out as a factor.

6 Conclusion

In this paper, we present a recurrent attention-based temporal fusion module for end-to-end autonomous driving, introducing a void token mechanism for selective forgetting, a mechanism currently lacking in recent related works. Integrated into TransFuser++, our method maintains competitive performance with the baseline while enabling analysis of how past information is retained or discarded. Qualitative results suggest that the mechanism can induce selective forgetting that aligns with expected behavior, although it also revealed expected limitations such as a possible overreliance on ego velocity. These findings highlight both the promise and the challenges of incorporating temporal fusion in E2E-AD. Beyond raw performance, our approach is a step towards more interpretable model design, a crucial challenge in the field of E2E-AD.

Disclosure of Interests. Florian Wintel received funding from the PERSEUS project, a European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No 101034240. He is also supported by the MoST (MobilitetsLab Stor-Trondheim) project (https://www.mobilitetslabstortrondheim.no/en/).

References

1. Carla autonomous driving leaderboard (2024), http://leaderboard.carla.org/leaderboard/
2. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European conference on computer vision. pp. 213–229. Springer (2020)
3. Casas, S., Luo, W., Urtasun, R.: Intentnet: Learning to predict intention from raw sensor data. In: Conference on robot learning. pp. 947–956. PMLR (2018)
4. Casas, S., Sadat, A., Urtasun, R.: Mp3: A unified model to map, perceive, predict and plan. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14403–14412 (2021)
5. Chen, D., Krähenbühl, P.: Learning from all vehicles. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 17222–17231 (2022)
6. Chen, L., Wu, P., Chitta, K., Jaeger, B., Geiger, A., Li, H.: End-to-end autonomous driving: Challenges and frontiers. IEEE Transactions on Pattern Analysis and Machine Intelligence (2024)

7. Chitta, K., Prakash, A., Jaeger, B., Yu, Z., Renz, K., Geiger, A.: Transfuser: Imitation with transformer-based sensor fusion for autonomous driving. IEEE transactions on pattern analysis and machine intelligence **45**(11), 12878–12895 (2022)
8. Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V.: Carla: An open urban driving simulator. In: Conference on robot learning. pp. 1–16. PMLR (2017)
9. Han, C., Yang, J., Sun, J., Ge, Z., Dong, R., Zhou, H., Mao, W., Peng, Y., Zhang, X.: Exploring recurrent long-term temporal fusion for multi-view 3d perception. IEEE Robotics and Automation Letters **9**(7), 6544–6551 (2024)
10. Hu, A., Murez, Z., Mohan, N., Dudas, S., Hawke, J., Badrinarayanan, V., Cipolla, R., Kendall, A.: Fiery: Future instance prediction in bird's-eye view from surround monocular cameras. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 15273–15282 (2021)
11. Hu, S., Chen, L., Wu, P., Li, H., Yan, J., Tao, D.: St-p3: End-to-end vision-based autonomous driving via spatial-temporal feature learning. In: European Conference on Computer Vision. pp. 533–549. Springer (2022)
12. Hu, Y., Yang, J., Chen, L., Li, K., Sima, C., Zhu, X., Chai, S., Du, S., Lin, T., Wang, W., et al.: Planning-oriented autonomous driving. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 17853–17862 (2023)
13. Jaeger, B., Chitta, K., Geiger, A.: Hidden biases of end-to-end driving models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8240–8249 (2023)
14. Jia, X., Yang, Z., Li, Q., Zhang, Z., Yan, J.: Bench2drive: Towards multi-ability benchmarking of closed-loop end-to-end autonomous driving. Advances in Neural Information Processing Systems **37**, 819–844 (2024)
15. Jia, X., You, J., Zhang, Z., Yan, J.: Drivetransformer: Unified transformer for scalable end-to-end autonomous driving. In: The Thirteenth International Conference on Learning Representations (2025), https://openreview.net/forum?id=M42KR4W9P5
16. Li, Z., Wang, W., Li, H., Xie, E., Sima, C., Lu, T., Yu, Q., Dai, J.: Bevformer: learning bird's-eye-view representation from lidar-camera via spatiotemporal transformers. IEEE Transactions on Pattern Analysis and Machine Intelligence (2024)
17. Lillicrap, T.P., Santoro, A., Marris, L., Akerman, C.J., Hinton, G.: Backpropagation and the brain. Nature Reviews Neuroscience **21**(6), 335–346 (2020)
18. Rao, Z., Cai, Y., Wang, H., Lian, Y., Zhong, Y., Chen, L., Li, Y.: Enhancing autonomous driving: a low-cost monocular end-to-end framework with multi-task integration and temporal fusion. IEEE Transactions on Intelligent Vehicles (2024)
19. Shao, H., Wang, L., Chen, R., Waslander, S.L., Li, H., Liu, Y.: Reasonnet: End-to-end driving with temporal and global reasoning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 13723–13733 (2023)
20. Toromanoff, M., Wirbel, E., Moutarde, F.: End-to-end model-free reinforcement learning for urban driving using implicit affordances. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 7153–7162 (2020)
21. Wu, P., Chen, S., Metaxas, D.N.: Motionnet: Joint perception and motion prediction for autonomous driving based on bird's eye view maps. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11385–11395 (2020)
22. Zimmerlin, J., Beißwenger, J., Jaeger, B., Geiger, A., Chitta, K.: Hidden biases of end-to-end driving datasets. arXiv preprint arXiv:2412.09602 (2024)