A Hybrid Autonomous Driving Navigation Architecture for the 2024 CARLA Autonomous Driving Challenge

Luis Alberto Rosero
lrosero@usp.br

Iago Pachêco Gomes
iagogomes@usp.br

Ana Huaman Reyna
arhuamanr@usp.br

Victor Hugo Sillerico Justo
vhsillerico@usp.br

Vinicius Gustierrez Neves
viniciusgustierrez@usp.br

Denis Fernando Wolf
denis@icmc.usp.br

Fernando Santos Osório
fosorio@icmc.usp.br

Institute of Mathematics and Computer Science
University of São Paulo
Ave. Trabalhador São-Carlense, 400, São Carlos, 13564-002, SP, Brazil

Abstract

This report presents a hybrid autonomous driving architecture that integrates modular components and data-driven path planning. The system generates dense trajectories using sensor fusion in the BEV (Bird's Eye View) space, combining stereo camera data, LiDAR point clouds, high-level commands, and historical traffic participant trajectories. Evaluated during the 2024 CARLA Autonomous Driving Challenge, the architecture outperformed baselines but requires improvement in executing evasive maneuvers and increasing agility in dynamic environments. Integrating historical traffic data into path planning enhanced interaction feature extraction and system robustness.

1. Introduction

Developing an autonomous system involves integrating software components and algorithms from fields such as machine learning, computer vision, decision theory, and probability theory. This integration adds complexity to both development and evaluation processes [5]. Modular architectures are effective in scenarios with detailed high-definition maps or dense waypoints, and allow comprehensive evaluation of each component [10, 12, 19, 21]. However, the extensive coupling of components increases error propagation risks, making maintenance complex and costly.

In contrast, the end-to-end approach leverages deep learning techniques to map sensor input directly to driving actions (i.e., steering, braking, and throttle), bypassing explicit task decomposition. This method simplifies the system architecture and reduces manual feature engineering but requires vast amounts of training data and lacks transparency in decision-making.

Autonomous navigation has progressed with both modular and end-to-end approaches, each with limitations. This has led to the development of a hybrid model combining their strengths. Modular systems, with many components, struggle to adapt to diverse, unpredictable situations, especially when relying on precise map data. End-to-end models, while effective, often operate as black boxes, making their decision-making processes hard to interpret and requiring extensive training data. The hybrid model integrates the advantages of both approaches, balancing the interpretability of modular systems and the learning capabilities of end-to-end models. It offers potential improvements in system robustness, flexibility, adaptability, and overall performance by combining deliberative and reactive behaviors.

This report presents the development and findings of an autonomous driving software architecture using a hybrid methodology that integrates modular perception and control modules with data-driven path planning. The architecture was designed and evaluated during the 2024 CARLA Autonomous Driving Challenge (CADCH). Based on the work of Rosero et al. [16], winner of the 2023 CADCH challenge on the SENSOR track, we incorporated the historical trajectories of traffic participants into the data-driven path planning. This allowed us to include information about the motion of these agents and extract relevant interaction features. The primary contributions of this architecture are as follows:

- A hybrid software architecture for autonomous vehicles, combining modular perception and control modules with data-driven path planning;
- Evaluation of autonomous driving performance in diverse and hazardous traffic events within urban environments;
- Integration of historical trajectory of traffic participants into a data-driven path planning, allowing the network to account for motion patterns and interaction features.

2. Related Works

Autonomous systems can be developed using different approaches based on the technologies used and component structure. A modular architecture organizes software components hierarchically, with each layer providing services to adjacent layers. For instance, following Paden et al.'s hierarchical navigation stack [13], the initial layer handles road and lane-level route planning. The behavior layer then makes tactical decisions, such as interactions with other traffic participants, adherence to traffic rules, and high-level maneuvers. The motion planning layer estimates short-term, feasible, and collision-free trajectories, translated into low-level commands by the control layer [1, 6, 10, 11, 20, 25]. However, as vehicle autonomy increases, the number of components grows, leading to error propagation and increased maintenance complexity and costs.

The end-to-end approach uses neural networks and deep learning to map sensor input directly to control outputs, eliminating the need for manual feature tuning. This method simplifies the navigation stack and enhances adaptability across traffic scenarios. Research on end-to-end navigation focuses on input representation [4, 17, 18, 26, 30], model design, sensor fusion [3, 27, 30], interaction with traffic agents [18, 30], deep learning technologies [3, 17, 18, 30], decision-making [4, 26], and output accuracy [3, 4, 18]. Despite its advantages, end-to-end navigation faces challenges with transparency, explainability, and extensive training data requirements.

Data-driven path planning, which uses machine learning to learn collision-free paths from large datasets [14], has gained attention [28]. This approach inherits adaptability from deep learning models, allowing planning under diverse road and traffic conditions. It often involves recurrent neural networks for temporal features [23], integration of semantic data like traffic rules, and post-processing for trajectory smoothness [9, 23]. However, it also faces challenges in transparency, addressing global planning, handling dangerous scenarios, and ensuring the feasibility of planned trajectories.

Finally, sensor fusion enhances the reliability of perception and data-driven path planning by integrating data from various sensors like cameras, LiDAR, and radar [7, 24]. Each sensor has unique strengths, and combining their data provides a robust environmental representation, leading to safer trajectories [29].

This report introduces a hybrid autonomous vehicle architecture combining modular pipelines with data-driven path planning. By integrating modular perception and control components, the system delivers reliable information to the data-driven path planning module, ensuring kinematically feasible trajectories. An early-fusion approach enhances spatial and semantic representation by fusing LiDAR Bird's Eye View images, stereo camera projections, high-level commands, and the historical trajectories of traffic agents enabling interaction representation. The results demonstrate the network's generalization capabilities with real-time inference, highlighting the architecture's reliability.

3. Software Architecture

Figure 1 shows the general software architecture, referred to as *CaRINA Agent*, designed for the modular architecture submitted to the MAP track of the competition, and used as the basis for the hybrid architecture submitted to the SENSOR track of the competition. The architecture consists of the following layers: sensing, perception, map, risk assessment, navigation, control, and vehicle. The robotic framework ROS (Robot Operating System) provided the communication interface between components using the Publish/Subscribe pattern for message passing

- **Perception**: The *CaRINA Agent* uses frontal camera images, stereo camera, and a LiDAR sensor for robust environmental perception. The components rely on a multi-sensor late-fusion approach for accurate object detection in 2D/3D images and point clouds.
  - *Obstacle Detection*: we combine three techniques to increase the accuracy and reliability of the environmental perception.

    ▶ The first method employs a *height-map* generated from the LiDAR point cloud [22]. This method analyzes height differences of a point cloud projected into a grid, identifying obstacles exceeding a threshold. This detection mechanism serves as a backup for emergency situations, such as when the deep learning-based method fails to detect (i.e., False Negative).

    ▶ The second method employs the *Mask R-CNN* for instance segmentation on images, extracting bounding boxes and masks for each object instance in the image, categorizing them into eight categories: car, bicycle, pedestrian, red/yellow/green traffic lights, stop, and emergency vehicle.

    ▶ The third method employs *PointPillars* algorithm for 3D detection using LiDAR point cloud. This deep learning model provides the regression of 3D bounding

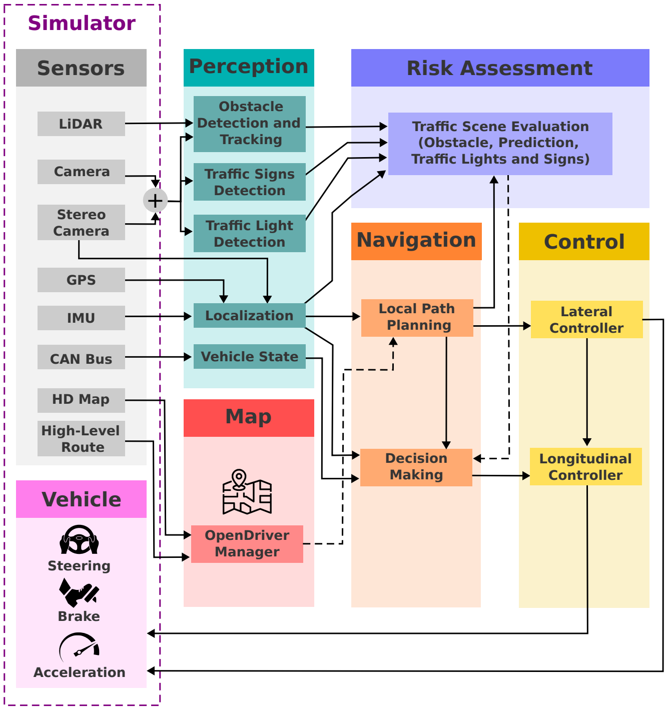

Figure 1. General design of the proposed modular architecture.

boxes and orientation related to the ego vehicle, categorizing them into three categories: pedestrian, car, and bicycle.

  - *Traffic Signs/Light Detection*: This approach combines the object mask from instance segmentation with the reconstructed point cloud from the stereo cameras, to identify the position and state of traffic lights. Moreover, a geometric analysis assigns the correspondent traffic light to the trajectory of the ego-vehicle.
  - *Disparity/Depth Estimation*: Disparity estimation generates an RGB-D image or RGB point cloud by processing stereo images. We use the Efficient Large-Scale stereo matching (ELAS) [8] algorithm for its computational efficiency and robustness in various scene complexities. Moreover, the modular pipeline allows the integration of other disparity algorithms, such as semi-global matching (SGM) or modern deep learning-based stereo-matching algorithms.
  - *Tracking*: We employ a tracking-by-detecting method, where Simple Online and Realtime Tracking (SORT) [2] uses Kalman Filter to estimate the position of the 3D obstacles and estimate their velocities.

- **Localization**: The ego-vehicle localization fuses odometry (stereo odometry), IMU orientation, and GNSS position using an Extended Kalman Filter (EKF). The inputs of this module include camera images, IMU orientation, GNSS coordinates, and sensor calibration. Outputs are the estimated pose and uncertainty. The EKF performs system prediction with each relative pose, updating the state when receives a global pose (GNNS).
- **Mapping and Path Planning**: In the modular architecture, we used the OpenDRIVE map standard to assist navigation and perception components. The path planning is divided into global and local planning. Global planning estimates a reference path, considering static features and intended routes. Local planning adjusts this path based on dynamic traffic variables. The autonomous agent receives from the simulator sparse waypoints representing the intended route. We use a lane-level localization to determine the lane and road ID for each waypoint using the OpenDRIVE map manager. This information helps estimate a lane-level route, identifying roads and lanes for the vehicle to traverse. Finally, segments of the reference path, each spanning 50-100 meters, are published in the ROS ecosystem based on the vehicle's speed and position.
- **Trajectory Prediction**: The architecture employs a short-term prediction-based approach to anticipate surrounding objects' movements, using a simple motion model based on their speeds. This model assumes constant velocity, providing an approximation of trajectories. The prediction module ensures computational efficiency, suitable for real-time applications.
- **Risk Assessment**: The architecture employs a dedicated collision risk assessment (CRA) module to continuously evaluate potential threats posed by both dynamic (cars, pedestrians, bicycles) and static surrounding objects. This module uses the current and future positions of surrounding objects and the ego-vehicle trajectory to detect collision based on a corridor strategy with different levels of risk. Figure 2 illustrates this strategy. The lengths of the two risk zones are estimated by the current speed of the ego-vehicle, where the first zone (red) has a fixed size of 8 meters, and the second zone (yellow) increases linearly according to increasing speed.
- **Decision-Making**: The decision-making module uses a synchronous Moore finite state machine (FSM) to plan actions based on CRA inputs. Each state governs specific speed control behaviors.
- **Control**: The control layer generates steering, throttle, and brake commands to keep the vehicle on the planned trajectory. This is achieved through closed-loop control mechanisms receiving desired trajectory information from the navigation layer. Longitudinal control uses a PID controller to follow the desired velocity, while lateral control employs model-based predictive control (MPC) to generate the steering signal. MPC optimizes control actions over a finite time horizon, continually adjusting based on updated measurements and conditions.

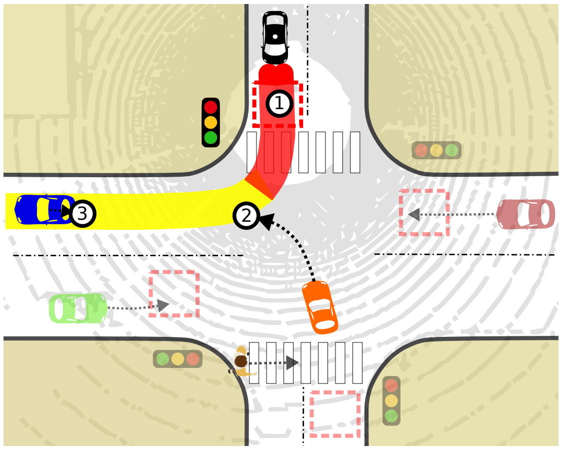

Figure 2. Risk Assessment with Path Prediction. **Point 1**: Zone of influence of a red traffic light. **Point 2**: The predicted trajectory of another car intersects the ego vehicle's path in the yellow zone. **Point 3**: A parked car within the yellow zone is identified as a potential obstacle but receives lower priority compared to threats in the red zone.

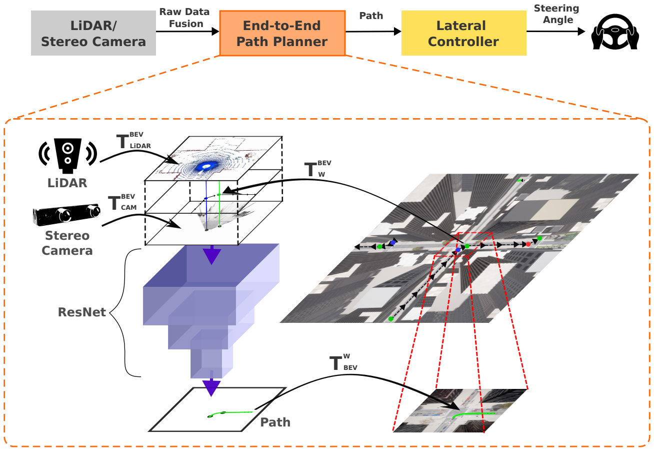

Figure 3. End-to-End Path Planning using CNN-Planner++

3.1. Data-driven Path Planning

Conventional map-based navigation architecture struggles in dynamic environments without accurate maps. We address this with an enhanced CNN-planner for mapless situations, generating waypoints using sensor fusion in the BEV (Bird's Eye View) space. Figure 3 illustrates the general design of the data-driven path planning model. The fusion method, BEVSFusion, based on [15], integrates data from four sources:

- **Stereo Camera**: Disparity maps are calculated using the ELAS algorithm and projected into a point cloud. This point cloud is transformed from the camera coordinate system to the BEV coordinate system.
- **LiDAR**: The LiDAR point cloud is directly transformed to the BEV coordinate system. LiDAR points are rasterized into an RGB image with height information encoded by color (blue for ground, yellow for above sensor). Empty pixels are filled with black.

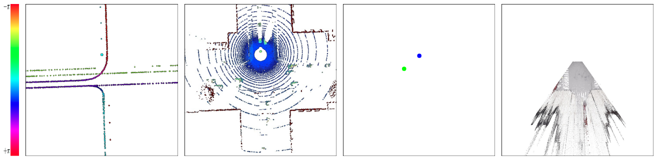

Figure 4. Rasterized inputs for the BEVSFusion structure: footprints (orientation angle mapped to hsv color), LiDAR point cloud, high-level commands, and stereo point cloud.

Table 1. Results Track MAP Leaderboard 2

Team	DS	RC	IS
Tuebingen_AI	5.564	11.815	0.467
LLM4AD (CarLLaVA)	5.040	15.761	0.431
Kyber-E2E	3.471	8.476	0.499
xproject	1.897	5.016	0.656
LRM (CaRINA modular)	0.707	4.758	0.415

- **High-Level Commands**: Global plan commands are converted from the world frame to the BEV frame using and rasterized as colored dots (blue for *right turn*, red for *left turn*, white for *straight*, green for *lane follow*). A line connecting adjacent commands enhances sequence representation.
- **Historical Trajectory of Traffic Participants**: We maintain a historical record of detections and their orientations for 200 frames, enabling the rasterization of the route traces (footprints) of surrounding vehicles. This input gives additional information to our trajectory planner to learn from other agents in its vicinity. Footprints' points are encoded using a HSV color map to represent the rotation angle of the historical poses.

These processed inputs are combined into a single 12-channel image (BEVSFusion), which serves as input to the CNN path planner. The network estimates a sequence of dense waypoints for the vehicle's trajectory. Figure 4 showcases an example of the inputs.

4. Results and Discussion

Tables 1 and 2 present the results of the MAP and SENSOR tracks of the 2024 CADCH, respectively. The results show that the architectures outperform the baseline, which uses privileged information from the simulation. However, more complex scenarios requiring evasive maneuvers remain challenging. The architecture needs improvements to become more reactive and agile in responding to events.

Table 2. Results Track SENSORS Leaderboard 2

Team	DS	RC	IS
LLM4AD (CarLLaVA)	6.865	18.083	0.423
Tuebingen_AI	5.183	11.336	0.479
LRM (CaRINA hybrid)	1.024	9.645	0.222
CARLA_Leaderboard (baseline)	0.250	15.200	0.103

5. Conclusion

We propose a versatile autonomous driving architecture that integrates map-based and mapless paradigms within a unified framework. Our system excels in both navigation styles by combining modularity with end-to-end path planning, facilitating transparent debugging and continuous performance improvement. The architecture outperforms competitive baselines. However, challenges remain, such as executing evasive maneuvers in blocked traffic scenarios and increasing the system's agility in performing these maneuvers.

References

[1] Autoware. Architecture overview. Available online: https://autowarefoundation.github.io/autoware-documentation/main/design/autoware-architecture (accessed on 23 January 2024). 2

[2] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 3464–3468, 2016. 3

[3] Peide Cai, Sukai Wang, Yuxiang Sun, and Ming Liu. Probabilistic end-to-end vehicle navigation in complex dynamic environments with multimodal sensor fusion. *IEEE Robotics and Automation Letters*, 5(3):4218–4224, 2020. 2

[4] Sergio Casas, Abbas Sadat, and Raquel Urtasun. Mp3: A unified model to map, perceive, predict and plan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14403–14412, 2021. 2

[5] Pranav Singh Chib and Pravendra Singh. Recent advancements in end-to-end autonomous driving using deep learning: A survey. *IEEE Transactions on Intelligent Vehicles*, 2023. 1

[6] Haoyang Fan, Fan Zhu, Changchun Liu, Liangliang Zhang, Li Zhuang, Dong Li, Weicheng Zhu, Jiangtao Hu, Hongye Li, and Qi Kong. Baidu apollo em motion planner. *arXiv*, abs/1807.08048, 2018. Available online: http://arxiv.org/abs/1807.08048 (accessed on 23 January 2024). 2

[7] Jamil Fayyad, Mohammad A Jaradat, Dominique Gruyer, and Homayoun Najjaran. Deep learning sensor fusion for autonomous vehicle perception and localization: A review. *Sensors*, 20(15):4220, 2020. 2

[8] Andreas Geiger, Martin Roser, and Raquel Urtasun. Efficient large-scale stereo matching. In *Asian Conference on Computer Vision (ACCV)*, 2010. 3

[9] Shengchao Hu, Li Chen, Penghao Wu, Hongyang Li, Junchi Yan, and Dacheng Tao. St-p3: End-to-end vision-based autonomous driving via spatial-temporal feature learning. In *European Conference on Computer Vision*, pages 533–549. Springer, 2022. 2

[10] Kichun Jo, Junsoo Kim, Dongchul Kim, Chulhoon Jang, and Myoungho Sunwoo. Development of autonomous car—part ii: A case study on the implementation of an autonomous driving system based on distributed architecture. *IEEE Transactions on Industrial Electronics*, 62(8):5119–5132, 2015. 1, 2

[11] Christos Katrakazas, Mohammed Quddus, Wen-Hua Chen, and Lipika Deka. Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions. *Transportation Research Part C: Emerging Technologies*, 60:416–442, 2015. 2

[12] Shaoshan Liu, Liyun Li, Jie Tang, Shuang Wu, and Jean-Luc Gaudiot. *Creating Autonomous Vehicle Systems*. Morgan & Claypool Publishers, 2017. 1

[13] 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(1):33–55, 2016. 2

[14] Mohamed Reda, Ahmed Onsy, Ali Ghanbari, and Amira Y Haikal. Path planning algorithms in the autonomous driving system: A comprehensive review. *Robotics and Autonomous Systems*, page 104630, 2024. 2

[15] Luis Alberto Rosero, Iago Pachêco Gomes, Júnior Anderson Rodrigues da Silva, Tiago Cesar dos Santos, Angelica Tiemi Mizuno Nakamura, Jean Amaro, Denis Fernando Wolf, and Fernando Santos Osório. A software architecture for autonomous vehicles: Team LRM-B entry in the first CARLA autonomous driving challenge. *arXiv*, abs/2010.12598, 2020. Available online: https://arxiv.org/abs/2010.12598 (accessed on 23 January 2024). 4

[16] Luis Alberto Rosero, Iago Pachêco Gomes, Júnior Anderson Rodrigues Da Silva, Carlos André Przewodowski, Denis Fernando Wolf, and Fernando Santos Osório. Integrating modular pipelines with end-to-end learning: A hybrid approach for robust and reliable autonomous driving systems. *Sensors*, 24(7):2097, 2024. 1

[17] Hao Shao, Letian Wang, Ruobing Chen, Hongsheng Li, and Yu Liu. Safety-enhanced autonomous driving using interpretable sensor fusion transformer. In *Proceedings of The 6th Conference on Robot Learning*, pages 726–737. PMLR, 2023. 2

[18] Hao Shao, Letian Wang, Ruobing Chen, Steven L. Waslander, Hongsheng Li, and Yu Liu. Reasonnet: End-to-end driving with temporal and global reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13723–13733, 2023. 2

[19] Ardi Tampuu, Tambet Matiisen, Maksym Semikin, Dmytro Fishman, and Naveed Muhammad. A survey of end-to-end driving: Architectures and training methods. *IEEE Trans-*

actions on Neural Networks and Learning Systems, 33(4): 1364–1384, 2020. 1

[20] Ömer Şahin Taş, Niels Ole Salscheider, Fabian Poggenhans, Sascha Wirges, Claudio Bandera, Marc René Zofka, Tobias Strauss, J Marius Zöllner, and Christoph Stiller. Making bertha cooperate–team annieway's entry to the 2016 grand cooperative driving challenge. *IEEE Transactions on Intelligent Transportation Systems*, 19(4):1262–1276, 2018. 2

[21] Siyu Teng, Xuemin Hu, Peng Deng, Bai Li, Yuchen Li, Yunfeng Ai, Dongsheng Yang, Lingxi Li, Zhe Xuanyuan, Fenghua Zhu, et al. Motion planning for autonomous driving: The state of the art and future perspectives. *IEEE Transactions on Intelligent Vehicles*, 2023. 1

[22] Sebastian Thrun, Mike Montemerlo, Hendrik Dahlkamp, David Stavens, Andrei Aron, James Diebel, Philip Fong, John Gale, Morgan Halpenny, Gabriel Hoffmann, Kenny Lau, Celia Oakley, Mark Palatucci, Vaughan Pratt, Pascal Stang, Sven Strohband, Cedric Dupont, Lars-Erik Jendrossek, Christian Koelen, Charles Markey, Carlo Rummel, Joe van Niekerk, Eric Jensen, Philippe Alessandrini, Gary Bradski, Bob Davies, Scott Ettinger, Adrian Kaehler, Ara Nefian, and Pamela Mahoney. *Stanley: The Robot That Won the DARPA Grand Challenge*, pages 1–43. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007. 2

[23] Matt Vitelli, Yan Chang, Yawei Ye, Ana Ferreira, Maciej Wołczyk, Błażej Osiński, Moritz Niendorf, Hugo Grimmett, Qiangui Huang, Ashesh Jain, et al. Safetynet: Safe planning for real-world self-driving vehicles using machine-learned policies. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 897–904. IEEE, 2022. 2

[24] Zhangjing Wang, Yu Wu, and Qingqing Niu. Multi-sensor fusion in automated driving: A survey. *Ieee Access*, 8:2847–2868, 2019. 2

[25] Junqing Wei, Jarrod M Snider, Junsung Kim, John M Dolan, Raj Rajkumar, and Bakhtiar Litkouhi. Towards a viable autonomous driving research platform. In *Intelligent Vehicles Symposium (IV), 2013 IEEE*, pages 763–770. IEEE, 2013. 2

[26] Penghao Wu, Xiaosong Jia, Li Chen, Junchi Yan, Hongyang Li, and Yu Qiao. Trajectory-guided control prediction for end-to-end autonomous driving: A simple yet strong baseline. In *NeurIPS*, 2022. 2

[27] Yi Xiao, Felipe Codevilla, Akhil Gurram, Onay Urfalioglu, and Antonio M. López. Multimodal end-to-end autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 23(1):537–547, 2022. 2

[28] Zifan Xu, Xuesu Xiao, Garrett Warnell, Anirudh Nair, and Peter Stone. Machine learning methods for local motion planning: A study of end-to-end vs. parameter learning. In *2021 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pages 217–222. IEEE, 2021. 2

[29] De Jong Yeong, Gustavo Velasco-Hernandez, John Barry, and Joseph Walsh. Sensor and sensor fusion technology in autonomous vehicles: A review. *Sensors*, 21(6):2140, 2021. 2

[30] Qingwen Zhang, Mingkai Tang, Ruoyu Geng, Feiyi Chen, Ren Xin, and Lujia Wang. Mmfn: Multi-modal-fusion-net for end-to-end driving. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8638–8643. IEEE, 2022. 2