Parting with Misconceptions about Learning-based Vehicle Motion Planning

Daniel Dauner[1,2] Marcel Hallgarten[1,3] Andreas Geiger[1,2] Kashyap Chitta[1,2]
[1]University of Tübingen [2]Tübingen AI Center [3]Robert Bosch GmbH
https://github.com/autonomousvision/nuplan_garage

Abstract—**The release of nuPlan marks a new era in vehicle motion planning research, offering the first large-scale real-world dataset and evaluation schemes requiring both precise short-term planning and long-horizon ego-forecasting. Existing systems struggle to simultaneously meet both requirements. Indeed, we find that these tasks are fundamentally misaligned and should be addressed independently. We further assess the current state of closed-loop planning in the field, revealing the limitations of learning-based methods in complex real-world scenarios and the value of simple rule-based priors such as centerline selection through lane graph search algorithms. More surprisingly, for the open-loop sub-task, we observe that the best results are achieved when using only this centerline as scene context (i.e., ignoring all information regarding the map and other agents). Combining these insights, we propose an extremely simple and efficient planner which outperforms an extensive set of competitors, winning the nuPlan planning challenge 2023.**

I. Introduction

Despite learning-based systems' success in vehicle motion planning research [9, 10, 26, 33, 7], a lack of standardized large-scale datasets for benchmarking holds back their transfer from research to applications [4, 19, 2]. The recent release of the nuPlan dataset and simulator [5], a collection of 1300 hours of real-world vehicle motion data, has changed this, enabling the development of a new generation of learned motion planners, which promise reduced manual design effort and improved scalability. Equipped with this new benchmark, we perform the first rigorous empirical analysis on a large-scale, open-source, and data-driven simulator for vehicle motion planning, including a comprehensive set of state-of-the-art (SoTA) planners [28, 25, 16] using the official metrics. Our analysis yields several surprising findings:

Open- and closed-loop evaluation are misaligned. Most learned planners are trained through the supervised learning task of forecasting the ego vehicle's future motion conditioned on a given goal location. We refer to this setting as ego-forecasting [10, 26, 23, 6]. In nuPlan, planners can be evaluated in two ways: by directly measuring ego-forecasting accuracy using distance-based metrics in an open-loop evaluation or by assessing driving-relevant closed-loop metrics such as progress and collision rates in a simulated setting, termed closed-loop evaluation. Our primary contribution lies in uncovering the performance trade-off between the open-loop and closed-loop evaluation schemes. While previous work on the simplistic CARLA simulator [14] has shown that open- and closed-loop evaluation can have little correlation [8], our results indicate a *negative correlation* exists when using nuPlan's metrics.

Rule-based planning generalizes. We surprisingly find that an established rule-based planning baseline from over twenty years ago [31] surpasses *all SoTA learning-based methods* in terms of closed-loop evaluation metrics on our benchmark. This contradicts the prevalent motivating claim used in most research on learned planners that rule-based planning faces difficulties in generalization. This was previously only verified on simpler benchmarks [33, 28, 25]. As a result, most current work on learned planning only compares to other learned methods, ignoring rule-based baselines [26, 7, 17].

A centerline is all you need for ego-forecasting. We implement a naïve learned planning baseline which does not incorporate any input about other agents in the scene and merely extrapolates the ego state conditioned on a centerline representation of the desired route. This baseline *sets the new SoTA* for open-loop evaluation on our benchmark. It does not require intricate scene representations (e.g. lane graphs, vectorized maps, rasterized maps, tokenized objects), which have been the central subject of inquiry in previous work [28, 25, 16]. None of these prior studies considered a simple centerline-only representation as a baseline, perhaps due to its extraordinary simplicity.

Long-horizon prediction adds little value. Using the insights gained, we propose novel rule-based and learned planners that achieve SoTA results on the open- and closed-loop evaluation metrics, respectively. We maintain simple input representations so that both of these planners demonstrate significant efficiency. Finally, we introduce a corrective strategy that involves predicting learned offsets to a rule-based plan resulting in a hybrid planner. While this hybrid planning strategy significantly boosts performance in open-loop evaluation compared to the closed-loop planner, we find that it yields no added benefits for closed-loop evaluation due to the incorporated learned component. This outcome suggests that accurate long-horizon prediction, often considered crucial in complex planning scenarios [27, 11], offers little to no additional value for closed-loop planning.

A preliminary version of our approach won the inaugural nuPlan challenge. Given its simplicity, it provides a robust starting point for future motion planning research on nuPlan.

II. Ego-forecasting and Planning are Misaligned

In this section, we provide the relevant background regarding the data-driven simulator nuPlan [5]. We then describe two baselines used in a preliminary experiment to demonstrate that although ego-forecasting and planning are often considered related tasks, they are not well-aligned given their definitions on nuPlan. Improvements in one task can often lead to degradation in the other.

nuPlan. Simulators like nuPlan enable rapid prototyping and testing of motion planners, facilitating swift iteration of ideas. nuPlan constructs a simulated environment as closely as possible to a real-world driving setting through data-driven simulation [1, 3, 29, 35, 32, 34, 15]. This method extracts road maps, traffic patterns, and object properties (positions, orientations, and speeds) from a pre-recorded dataset consisting of 1,300 hours of real-world driving. These elements are then used to initialize scenarios, which are 15-second simulations employed to assess open-loop and closed-loop driving performance. In the open-loop simulation, the entire log is merely replayed (for both the ego vehicle and other actors). Conversely, in closed-loop simulation, the ego vehicle operates under the control of the planner being tested. There are two versions of closed-loop simulation: non-reactive, where all other actors are replayed along their original trajectory, and reactive, where other vehicles employ an `IDM` planner [31], which we describe in more detail in the following.

Metrics. nuPlan offers three official evaluation metrics: open-loop score (OLS), closed-loop score non-reactive (CLS-NR), and closed-loop score reactive (CLS-R). Although CLS-NR and CLS-R are computed identically, they differ in background traffic behavior. Each score is a weighted combination of established metrics, scaled within a range of 0-100. A high open-loop score necessitates low displacement and heading errors between the planner's and recorded trajectories over an extended period (8 seconds). The closed-loop score mandates that the generated plan remains on the road, adheres to the traffic's direction, avoids collisions where the planner is at fault, makes progress, maintains comfort, and observes the speed limit, all of which rely on an accurate short-term trajectory. For additional information on the metrics' composition, please refer to [20].

Intelligent Driver Model. The simple planning baseline `IDM` [31] serves not only as the mechanism for simulating other actors in the CLS-R evaluation within nuPlan, but also as a baseline for the ego-vehicle's planning. The nuPlan map is provided as a graph, with centerline segments functioning as nodes. After choosing a set of such nodes to follow via a graph search algorithm, `IDM` infers a longitudinal trajectory along the selected centerline. Given the current longitudinal position x, velocity v, and distance to the leading vehicle s along the centerline, it iteratively applies the following policy to calculate a longitudinal acceleration:

$$\frac{\mathrm{d}v}{\mathrm{d}t} = a\left(1 - \left(\frac{v}{v_0}\right)^{\delta} - \left(\frac{s^*}{s}\right)^2\right). \tag{1}$$

The acceleration limit a, target speed v_0, safety margin s^*, and exponent δ are manually selected. Intuitively, the policy uses an acceleration a unless the velocity is already close to v_0 or the leading vehicle is at a distance of only s^*.

Centerline-conditioned ego-forecasting. We now propose the Predictive Driver Model (Open), i.e., `PDM-Open`, which is a straightforward multi-layer perceptron (MLP) designed to predict future trajectories. The inputs to this MLP are the centerline ($\mathbf{c}$) extracted by `IDM` and the ego history ($\mathbf{h}$). To accommodate the high speeds (reaching up to 15 m/s) and ego-forecasting horizons (extending to 8 seconds) observed in nuPlan, the centerline is sampled with a resolution of 1 meter up to a length of 120 meters. Meanwhile, the ego history incorporates the positions, velocities, and accelerations of the vehicle over the previous two seconds, sampled at a rate of 5Hz. Both $\mathbf{c}$ and $\mathbf{h}$ are linearly projected to feature vectors of size 512, concatenated, and input to the MLP $\phi_{\texttt{Open}}$ which has two 512-dimensional hidden layers. The output is the future waypoints for an 8-second horizon, spaced 0.5 seconds apart, expressed as $\mathbf{w}_{\texttt{Open}} = \phi_{\texttt{Open}}(\mathbf{c}, \mathbf{h})$. The model is trained using an L_1 loss on our training dataset of 177k samples (elaborated upon in Section III). By design, `PDM-Open` is considerably simpler than existing learned planners [28, 16].

OLS vs. CLS. In Table I, we benchmark the `IDM` and `PDM-Open` baselines using the nuPlan metrics. We present two `IDM` variants with different maximum acceleration values (the default $a = 1.0\mathrm{ms}^{-2}$ and $a = 0.1\mathrm{ms}^{-2}$) and four `PDM-Open` variants based on witholding different inputs. While `IDM` demonstrates strong closed-loop performance, `PDM-Open` excels in open-loop. Reducing `IDM`'s acceleration improves OLS but negatively impacts CLS. For `PDM-Open`, adding the centerline significantly contributes to ego-forecasting performance, with minor enhancements when adding a 2-second state history. However, including the ego-history leads to a drop in CLS. A clear trade-off between CLS and OLS indicates a misalignment between the goals of ego-forecasting and planning. This sort of inverse correlation on nuPlan is unanticipated, considering the increasing use of ego-forecasting in current planning literature [26, 28, 16, 25].

Method	**Centerline**	**History**	**CLS-R ↑**	**CLS-NR ↑**	**OLS ↑**
`IDM` [31] $a = 1.0$	✓	-	**77**	**76**	38
$a = 0.1$	✓	-	54	66	48
`PDM-Open`	-	-	51	50	69
	-	✓	38	34	72
	✓	-	54	53	85
	✓	✓	54	50	**86**

TABLE I: **OLS-CLS Tradeoff.** Baseline scores on nuPlan.

Improving closed-loop driving. We now extend `IDM` by incorporating several concepts from model predictive control, including forecasting, proposals, simulation, scoring, and selection, as illustrated in Fig. 1 (top). We call this model `PDM-Closed`. Note that as a first step, we still require a graph search to find a sequence of lanes along the route and extract their centerline, as in the `IDM` planner.

Forecasting. In nuPlan, the simulator provides an orientation vector and speed for each dynamic agent such as a vehicle or pedestrian. By linearly extrapolating these values, we can approximate the future positions of the agents up to the

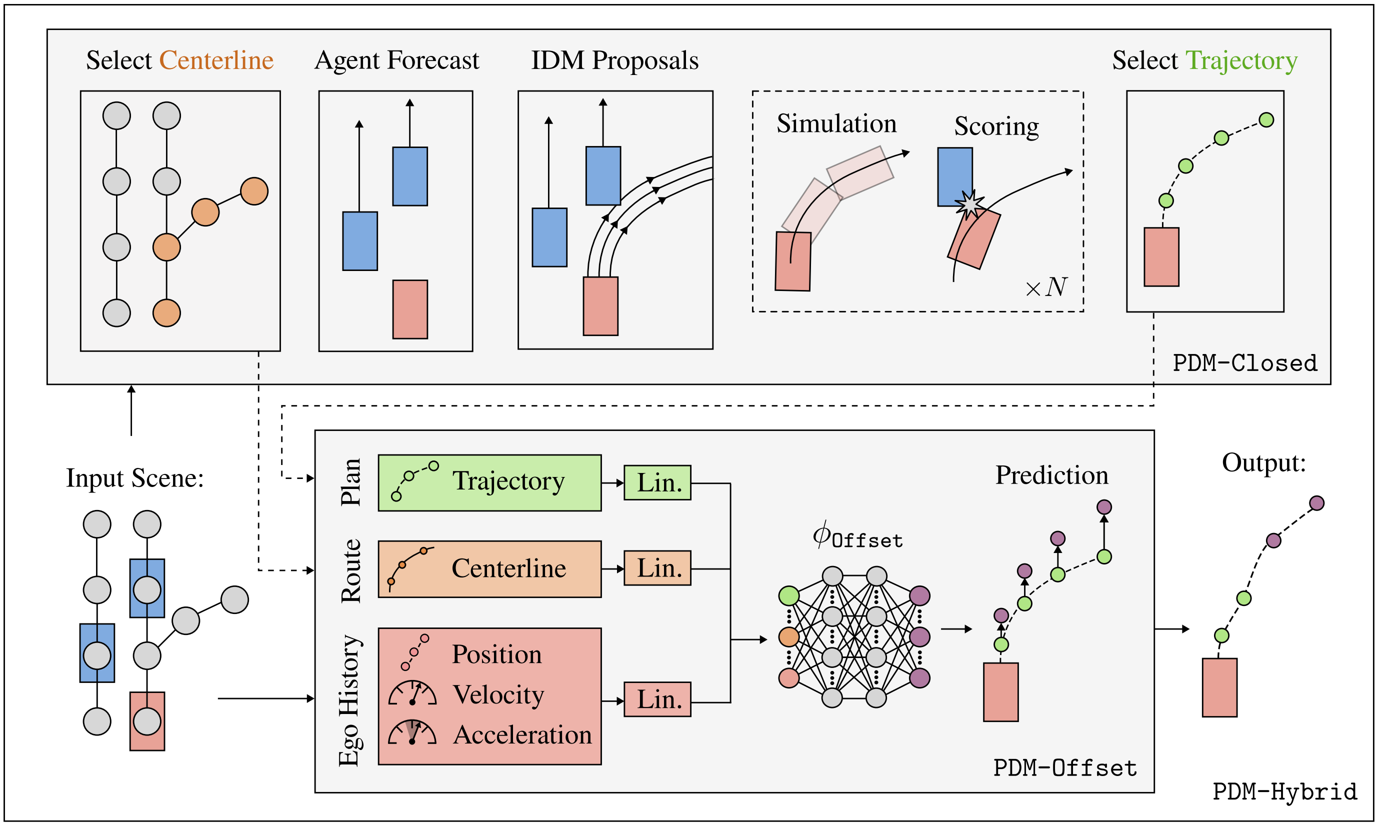

Fig. 1: **Architecture.** `PDM-Closed` selects a centerline, forecasts the environment, and creates varying trajectory proposals, which are simulated and scored for trajectory selection. The `PDM-Hybrid` module predicts offsets using the `PDM-Closed` centerline, trajectory, and ego history, correcting only long-term waypoints and thereby limiting the learned model's influence in closed-loop simulation.

forecasting horizon F of 8 seconds at 10Hz.

Proposals. In the process of calibrating the `IDM` planner, we observed a trade-off when selecting a single value for the target speed hyperparameter (v_0), which either yielded aggressive driving behavior or insufficient progress across various scenarios. Consequently, we generate a set of trajectory proposals by implementing `IDM` policies at five distinct target speeds, namely, $\{20\%, 40\%, 60\%, 80\%, 100\%\}$ of the designated speed limit. For each target speed, we also incorporate proposals with three lateral centerline offsets ($\pm$1m and 0m), thereby producing $N = 15$ proposals in total. To circumvent computational demands in subsequent stages, the proposals have a reduced horizon of H steps, which corresponds to 4 seconds at a 10Hz.

Simulation. Trajectories in nuPlan are simulated by iteratively retrieving actions from a LQR controller [30] and propagating the ego vehicle with a kinematic bicycle model [24, 22]. We simulate the proposals with the same parameters and a faster re-implementation of this pipeline. Thereby, the proposals are evaluated based on the expected movement in closed-loop.

Scoring. Each simulated proposal is scored to favor traffic-rule compliance, progress, and comfort. By considering proposals with lateral and longitudinal variety, the planner can avoid collisions with agent forecasts and correct drift that may arise when the controller fails to accurately track the intended trajectory. Furthermore, our scoring function closely resembles the nuPlan evaluation metrics.

Trajectory selection. Finally, `PDM-Closed` selects the highest-scoring proposal which is extended to the complete forecasting horizon F with the corresponding `IDM` policy. If the best trajectory is expected to collide within 2 seconds, the output is overwritten with an emergency brake maneuver.

Enhancing long-horizon accuracy. To integrate the accurate ego-forecasting capabilities of `PDM-Open` with the precise short-term actions of `PDM-Closed`, we now propose a hybrid version of `PDM`, i.e., `PDM-Hybrid`. Specifically, `PDM-Hybrid` uses a learned module `PDM-Offset` to predict offsets to waypoints from `PDM-Closed`, as shown in Fig. 1 (bottom).

In practice, the LQR controller used in nuPlan relies exclusively on the first 2 seconds of the trajectory when determining actions in closed-loop. Therefore, applying the correction only to long-term waypoints (i.e., beyond 2 seconds by default, which we refer to as the correction horizon C) allows `PDM-Hybrid` to maintain closed-loop planning performance. The final planner output waypoints (up to the forecasting horizon F) $\{\mathbf{w}^t_{\texttt{Hybrid}}\}^F_{t=0}$ are given by:

$$\mathbf{w}^t_{\texttt{Hybrid}} = \mathbf{w}^t_{\texttt{Closed}} + \mathbb{1}_{[t>C]}\phi^t_{\texttt{Offset}}(\mathbf{w}_{\texttt{Closed}}, \mathbf{c}, \mathbf{h}). \quad (2)$$

Where $\mathbf{c}$ and $\mathbf{h}$ are the centerline and history (identical to the

Method	Rep.	CLS-R ↑	CLS-NR ↑	OLS ↑	Time ↓
`Urban Driver` [28]	Polygon	44	45	76	64
`GC-PGP` [16]	Graph	54	57	82	100
`PlanCNN` [25]	Raster	72	73	64	43
`IDM` [31]	Centerline	77	76	38	27
`PDM-Open`	Centerline	54	50	**86**	**7**
`PDM-Closed`	Centerline	**92**	**93**	44	91
`PDM-Hybrid`	Centerline	**92**	**93**	84	96
`PDM-Hybrid*`	Graph	**92**	**93**	84	172
Log Replay	*GT*	*80*	*94*	*100*	-

TABLE II: **Val14 benchmark.** We show the closed-loop score reactive/non-reactive (CLS-R/CLS-NR), open loop score (OLS) and runtime in ms for several planners. We specify the input representation (Rep.) used by each planner. `PDM-Hybrid` accomplishes strong ego-forecasting (OLS) and planning (CLS). *This is a preliminary version of `PDM-Hybrid` that combined `PDM-Closed` with `GC-PGP` [16], and was used in our online leaderboard submission (Table III)

inputs of `PDM-Open`). $\{\mathbf{w}^t_{\texttt{Closed}}\}_{t=0}^F$ are the `PDM-Closed` waypoints added to the hybrid approach, and $\phi_{\texttt{Offset}}$ is an MLP. Its architecture is identical to $\phi_{\texttt{Open}}$ except for an extra linear projection to accommodate $\mathbf{w}_{\texttt{Closed}}$ as an additional input.

It is important to note that `PDM-Hybrid` is designed with high modularity, enabling the substitution of individual components with alternative options when diverse requirements emerge. Given its overall simplicity, one interesting approach to explore involves incorporating modular yet differentiable algorithms as components, as seen in [18]. Exploring the integration of these modules within unified multi-task architectures is another interesting direction. We reserve such exploration for future work.

III. Experiments

We now outline our proposed benchmark and highlight the driving performance of our approach.

Val14 benchmark. We offer standardized data splits for training and evaluation. Training uses all 70 scenario types from nuPlan, restricted to a maximum of 4k scenarios per type, resulting in ∼177k training scenarios. For evaluation, we use 100 scenarios of the 14 scenario types considered by the leaderboard, totaling 1,118 scenarios. Despite minor imbalance (all 14 types do not have 100 available scenarios), our validation split aligns with the online leaderboard evaluation (Table II and Table III), confirming the suitability of our Val14 benchmark as a proxy for the online test set.

Baselines. We include several additional SoTA approaches adopting ego-forecasting for planning in our study. `Urban Driver` [28] encodes polygons with PointNet layers and predicts trajectories with a linear layer after a multi-head attention block. `GC-PGP` [16] clusters trajectory proposals based on route-constrained lane-graph traversals before returning the most likely cluster center. `PlanCNN` [25] predicts waypoints using CNN from rasterized grid features without ego-state input. It shares several similarities to `ChauffeurNet` [2], a seminal work in the field. A preliminary version of `PDM-Hybrid`, which won the nuPlan competition, used `GC-PGP` as its ego-forecasting component, and we include this as a baseline.

Results. Our results are presented in Table II. Intriguingly, PlanCNN achieves the best CLS among learned planners, possibly due to its design choice of removing ego state from input, trading OLS for enhanced CLS. Contrary to the community's growing preference for graph- and vector-based scene representations in prediction and planning [13, 25, 21, 12], these results show no clear advantage in adopting these methods for the closed-loop task, with the raster-based PlanCNN also offering a lower runtime. Surprisingly, the simplest rule-based approach in our study, `IDM`, outperforms the best learned planner, `PlanCNN`. Moreover, we observe `PDM-Closed`'s advantages over `IDM` in terms of CLS: an improvement from 76-77 to 92-93 as a result of the ideas from Section II.

Notably, the centerline representation serves as a highly valuable prior for achieving the SoTA OLS of 86 with `PDM-Open` in a runtime of only 7ms. Next, despite `PDM-Closed`'s unsatisfactory 44 OLS, `PDM-Hybrid` successfully combines `PDM-Closed` with `PDM-Open`. Both the centerline and graph versions of `PDM-Hybrid` achieve identical scores in our evaluation. However, the final centerline version, using `PDM-Open` instead of `GC-PGP`, is more efficient during inference.

Finally, the privileged approach of outputting the ground-truth ego future trajectory (log replay) fails to achieve a perfect CLS, in part due to the nuPlan framework's LQR controller occasionally drifting from the provided trajectory. `PDM-Hybrid` compensates for this by evaluating proposals based on the expected controller outcome, causing it to match/outperform log replay in closed-loop evaluation.

Method	CLS-R ↑	CLS-NR ↑	OLS ↑	Score ↑
`PDM-Hybrid*`	**93**	**93**	83	**90**
hoplan	89	88	85	87
pegasus_multi_path	82	85	**88**	85
`Urban Driver` [28]	68	70	86	75
`IDM` [31]	72	75	29	59

TABLE III: **2023 nuPlan Challenge.**

Challenge. The 2023 nuPlan challenge saw the preliminary (graph) version of `PDM-Hybrid` rank first out of 25 participating teams. The leaderboard considers the mean of CLS-R, CLS-NR, and OLS. While open-loop performance lagged slightly, closed-loop performance excelled, resulting in an overall SoTA score. Unfortunately, due to the closure of the leaderboard, our final (centerline) version of `PDM-Hybrid` that replaces `GC-PGP` with the simpler `PDM-Open` module could not be benchmarked.

IV. Conclusion

We identify prevalent misconceptions in learning-based vehicle motion planning and provide evidence in our paper that challenges these notions. We also introduce `PDM-Hybrid` which surpasses a comprehensive set of competitors on nuPlan and claimed victory in the 2023 nuPlan challenge.

REFERENCES

[1] Matthias Althoff, Markus Koschi, and Stefanie Manzinger. Commonroad: Composable benchmarks for motion planning on roads. In *Proc. IEEE Intelligent Vehicles Symposium (IV)*, 2017.

[2] Mayank Bansal, Alex Krizhevsky, and Abhijit S. Ogale. Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst. In *Proc. Robotics: Science and Systems (RSS)*, 2019.

[3] Luca Bergamini, Yawei Ye, Oliver Scheel, Long Chen, Chih Hu, Luca Del Pero, Blazej Osinski, Hugo Grimmett, and Peter Ondruska. Simnet: Learning reactive self-driving simulations from real-world observations. In *Proc. IEEE International Conf. on Robotics and Automation (ICRA)*, 2021.

[4] 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.org*, 1604.07316, 2016.

[5] Holger Caesar, Juraj Kabzan, Kok Seang Tan, Whye Kit Fong, Eric M. Wolff, Alex H. Lang, Luke Fletcher, Oscar Beijbom, and Sammy Omari. nuplan: A closed-loop ml-based planning benchmark for autonomous vehicles. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2021.

[6] Kashyap Chitta, Aditya Prakash, and Andreas Geiger. Neat: Neural attention fields for end-to-end autonomous driving. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2021.

[7] Kashyap Chitta, Aditya Prakash, Bernhard Jaeger, Zehao Yu, Katrin Renz, and Andreas Geiger. Transfuser: Imitation with transformer-based sensor fusion for autonomous driving. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 2022.

[8] Felipe Codevilla, Antonio M. Lopez, Vladlen Koltun, and Alexey Dosovitskiy. On offline evaluation of vision-based driving models. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2018.

[9] Felipe Codevilla, Matthias Miiller, Antonio López, Vladlen Koltun, and Alexey Dosovitskiy. End-to-end driving via conditional imitation learning. In *Proc. IEEE International Conf. on Robotics and Automation (ICRA)*, 2018.

[10] Felipe Codevilla, Eder Santana, Antonio M. López, and Adrien Gaidon. Exploring the limitations of behavior cloning for autonomous driving. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2019.

[11] Alexander Cui, Abbas Sadat, Sergio Casas, Renjie Liao, and Raquel Urtasun. Lookout: Diverse multi-future prediction and planning for self-driving. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2021.

[12] Alexander Cui, Sergio Casas, Kelvin Wong, Simon Suo, and Raquel Urtasun. Gorela: Go relative for viewpoint-invariant motion forecasting. In *Proc. IEEE International Conf. on Robotics and Automation (ICRA)*, 2023.

[13] Nachiket Deo, Eric M. Wolff, and Oscar Beijbom. Multimodal Trajectory Prediction Conditioned on Lane-Graph Traversals. In *Proc. Conf. on Robot Learning (CoRL)*, 2021.

[14] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proc. Conf. on Robot Learning (CoRL)*, 2017.

[15] Lan Feng, Quanyi Li, Zhenghao Peng, Shuhan Tan, and Bolei Zhou. Trafficgen: Learning to generate diverse and realistic traffic scenarios. In *Proc. IEEE International Conf. on Robotics and Automation (ICRA)*, 2023.

[16] Marcel Hallgarten, Martin Stoll, and Andreas Zell. From Prediction to Planning With Goal Conditioned Lane Graph Traversals. *arXiv.org*, 2302.07753, 2023.

[17] Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, Lewei Lu, Xiaosong Jia, Qiang Liu, Jifeng Dai, Yu Qiao, and Hongyang Li. Planning-oriented autonomous driving. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[18] Peter Karkus, Boris Ivanovic, Shie Mannor, and Marco Pavone. Diffstack: A differentiable and modular control stack for autonomous vehicles. In *Proc. Conf. on Robot Learning (CoRL)*, 2022.

[19] 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. *arXiv.org*, abs/1807.00412, 2018.

[20] Motional. nuplan metrics, 2023. URL https://nuplan-devkit.readthedocs.io/en/latest/metrics_description.html.

[21] Nigamaa Nayakanti, Rami Al-Rfou, Aurick Zhou, Kratarth Goel, Khaled S. Refaat, and Benjamin Sapp. Wayformer: Motion forecasting via simple and efficient attention networks. In *Proc. IEEE International Conf. on Robotics and Automation (ICRA)*, 2023.

[22] Philip Polack, Florent Altché, Brigitte d'Andréa Novel, and Arnaud de La Fortelle. The kinematic bicycle model: A consistent model for planning feasible trajectories for autonomous vehicles? In *Proc. IEEE Intelligent Vehicles Symposium (IV)*, 2017.

[23] Aditya Prakash, Kashyap Chitta, and Andreas Geiger. Multi-modal fusion transformer for end-to-end autonomous driving. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[24] Rajesh Rajamani. *Vehicle dynamics and control*. Springer Science & Business Media, 2011.

[25] Katrin Renz, Kashyap Chitta, Otniel-Bogdan Mercea, Almut Sophia Koepke, Zeynep Akata, and Andreas Geiger. Plant: Explainable planning transformers via object-level representations. In *Proc. Conf. on Robot Learning (CoRL)*, 2022.

[26] Nicholas Rhinehart, Rowan McAllister, Kris Kitani, and

Sergey Levine. PRECOG: prediction conditioned on goals in visual multi-agent settings. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2019.

[27] Nicholas Rhinehart, Jeff He, Charles Packer, Matthew A Wright, Rowan McAllister, Joseph E Gonzalez, and Sergey Levine. Contingencies from observations: Tractable contingency planning with learned behavior models. In *Proc. IEEE International Conf. on Robotics and Automation (ICRA)*, 2021.

[28] Oliver Scheel, Luca Bergamini, Maciej Wolczyk, Błażej Osiński, and Peter Ondruska. Urban driver: Learning to drive from real-world demonstrations using policy gradients. In *Proc. Conf. on Robot Learning (CoRL)*, 2021.

[29] Simon Suo, Sebastian Regalado, Sergio Casas, and Raquel Urtasun. Trafficsim: Learning to simulate realistic multi-agent behaviors. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[30] Y. Tassa, N. Mansard, and E. Todorov. Control-limited differential dynamic programming. In *Proc. IEEE International Conf. on Robotics and Automation (ICRA)*, 2014.

[31] Martin Treiber, Ansgar Hennecke, and Dirk Helbing. Congested traffic states in empirical observations and microscopic simulations. *Physical review E*, 2000.

[32] Danfei Xu, Yuxiao Chen, Boris Ivanovic, and Marco Pavone. Bits: Bi-level imitation for traffic simulation. In *Proc. IEEE International Conf. on Robotics and Automation (ICRA)*, 2023.

[33] Wenyuan Zeng, Wenjie Luo, Simon Suo, Abbas Sadat, Bin Yang, Sergio Casas, and Raquel Urtasun. End-to-end interpretable neural motion planner. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[34] Zhejun Zhang, Alexander Liniger, Dengxin Dai, Fisher Yu, and Luc Van Gool. TrafficBots: Towards world models for autonomous driving simulation and motion prediction. In *Proc. IEEE International Conf. on Robotics and Automation (ICRA)*, 2023.

[35] Ziyuan Zhong, Davis Rempe, Danfei Xu, Yuxiao Chen, Sushant Veer, Tong Che, Baishakhi Ray, and Marco Pavone. Guided conditional diffusion for controllable traffic simulation. In *Proc. IEEE International Conf. on Robotics and Automation (ICRA)*, 2023.