Route Conditioned Motion Planning using Natural Language

Marlon Steiner* , Hanfeng Wu† , Lingguang Wang‡ , Willi Poh§ , Ömer Şahin Taş¶ , and Christoph Stiller‖

Abstract: Mapless driving lacks globally consistent lane-level route information from an HD map. At the same time, motion planning still depends on local lane structure produced online by the perception and mapping stack. We address this gap by conditioning planning on natural-language route descriptions that encode global routing intent and aligning them with learned embeddings of the local map. To study this setting at scale, we extend nuPlan with 60-second natural-language route descriptions for 1 000 000 training scenarios and the Val14 split, train language-conditioned models on this extension, and evaluate on Val14. Compared to route-conditioned baselines, our language-conditioned models achieve a better trade-off between driving quality and route compliance, with particularly strong gains in long-horizon navigation compliance. In our experiments, we use ground-truth lane-level local maps in place of online mapping outputs to isolate the effect of route-language conditioning. The dataset is available at `github.com/KIT-MRT/nuplan-language-based-routing-dataset`.

Keywords: Deep-learning, Flow-matching, Language-guidance, Motion Planning, Routing

1 Introduction

Autonomous driving systems rely on high-definition maps and explicit route information to guide motion planning. In mapless driving settings, however, such global routing information is not available, and planning must operate solely on an online-perceived local map. This creates a fundamental challenge: how to convey global routing intent to a learned motion planner without access to map-based routing graphs. We address this problem by representing routing information through natural language and aligning it with learned embeddings of the locally perceived map.

Natural language provides an interface for specifying routing intent. Traditional planning representations such as waypoints [25] or rule-based graphs [13] lack semantic transparency and are difficult for humans to interpret or specify. In contrast, language naturally captures high-level navigation intent and abstractions. Recent datasets and benchmarks

*Marlon Steiner is Ph.D. student at the Institute of Measurement and Control Systems (MRT), Karlsruhe Institute of Technology (KIT) (contact: marlon.steiner@kit.edu).

†Hanfeng Wu is Ph.D. student at BMW Group.

‡Lingguang Wang is Postdoc at MRT, KIT.

§Willi Poh is Ph.D. student at MRT, KIT.

¶Ömer Şahin Taş is group leader at FZI Research Center for Information Technology.

‖Christoph Stiller is Head of Institute and Professor at MRT, KIT

increasingly explore language as a conditioning signal for driving behavior, including ProSim-Instruct-520k [18] for the Waymo Open Motion Dataset [17], NAVSIM [5] as a nuPlan [3] based dataset with discrete routing commands, language-annotated extensions [15, 22, 16] of nuScenes [2] and CARLA [6], and the vision-language-action dataset CoVLA [1]. These works demonstrate a growing community interest in language beyond descriptive captions; however, they primarily focus on local behavioral instructions or short-horizon commands.

In parallel, recent models incorporating language or semantics [19] into autonomous driving, ranging from models processing local navigation conditioning as external signals [23] or generating them by themselves [21], to large-language model-based planners [12, 11, 7] and vision-language-action models [14, 1], suggest that language conditioning improves flexibility and generalization in complex driving scenarios. However, existing approaches do not address long-horizon routing in mapless settings, and large-scale datasets with natural language route descriptions remain rare.

In this work, we introduce a language-based routing approach for mapless driving that aligns natural language route descriptions with learned local map embeddings and integrates them into a flow-matching [8, 9] motion planner. We evaluate our method on nuPlan and provide a dataset extension containing 60-second natural language route descriptions for the nuPlan dataset for one million training scenarios and the Val14 validation split [4]. Together, our method and dataset enable long-horizon, language-conditioned routing for learned motion planning in autonomous driving settings.

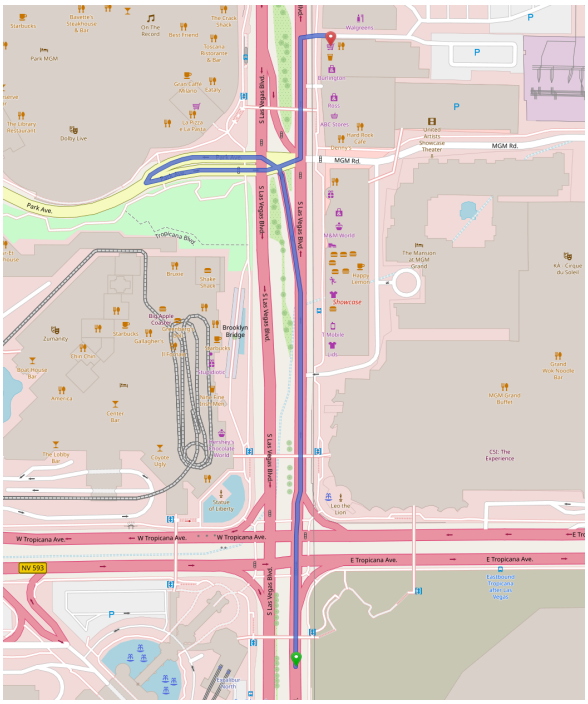
2 The Routing Dataset

This section describes our routing dataset. Compared to the doScenes [15] or LMDrive [16] datasets, our focus is more on long-term routing and not on scene dependent instructions at the behavioral level. For each chosen scenario, we generate a route from the start to the goal position and create a natural language route description.

Data extraction from nuPlan First, we extract the routing relevant information from the dataset, including the start position and the corresponding goal after 60 seconds (or less, if the scenarios are shorter). In line with recent related work, we choose 1 000 000 training scenarios [24]. Additionally, we provide routing data for the Val14 split [4]. Positions in the dataset are given in the UTM coordinate system, while most routing frameworks use lat/lon coordinates. Therefore, we transform the start and goal positions into global lat/lon (GCS) coordinates.

Route querying with routing framework For generating the route, we decided to choose the open-source framework Open Source Routing Machine (OSRM) [10], which is based on OpenStreetMap (OSM). An OSRM routing machine can be set up locally, which allows us to generate the required amount of routing data for the dataset extension. Using this setup, routing requests can be parallelized. A route is generated by querying the routing machine with the start and goal positions.

Route description generation The return of the routing machine is an information rich routing data collection. We mainly utilize four features of the data: Distance to



‘‘
Depart;
Turn slight left in 330 meters;
Turn left in 110 meters;
Continue uturn in 120 meters;
Turn left in 140 meters;
Turn right in 140 meters;
Arrive at destination in 0 meters
’’

Figure 1: Route for scenario with token `1a9aa434305b551b`. Left: Visualization of the OSRM routing. Right: Natural language-based route description, containing seven maneuvers, resp. steps with types, modifiers and distances.

next maneuver, the maneuver type and modifier, and exit (only provided if the maneuver takes place in a roundabout); see fig. 1. Maneuver types express the actions to take, while modifiers characterize the actions. The existing types in OSRM are: {`turn`, `new name`, `depart`, `arrive`, `merge`, `ramp`, `on ramp`, `off ramp`, `fork`, `end of road`, `use lane`, `continue`, `roundabout`, `rotary`, `roundabout turn`, `notification`}, while modifiers can take on the values from the following set: {`uturn`, `sharp right`, `right`, `slight right`, `straight`, `slight left`, `left`, `sharp left`}. We design a heuristic that combines the existing information for each maneuver (step) in the route and generates a sentence with it. Combining all steps of the route creates the full language-based route description as in fig. 1.

3 Route Alignment with Local Map

In this section, we describe how we process natural-language route descriptions and fuse them with a local map representation produced by the online perception and mapping stack (approximated by ground-truth local maps in our experiments) to condition a motion planner. The goal here is to align the global route description with the local map. This enables a model to learn which lanes of a local map are part of the route to ultimately reach the destination.

Language Processing The natural language route description $r = \{s_1, s_2, ..., s_n\}$, with n representing the number of steps s in the route, is processed by a language encoder that maps a variable-length text input into a route embedding of the same length (see fig. 2).

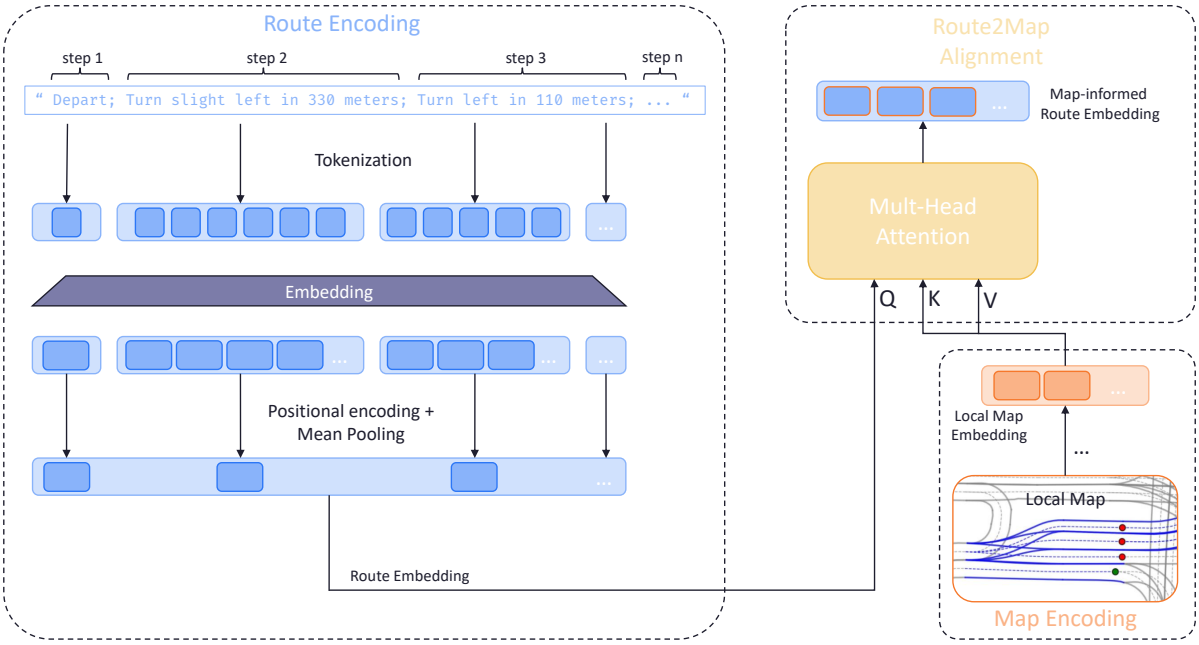


Figure 2: Route-map alignment via cross-attention. The `Route Encoding` block shows the calculation of the route embedding based on natural language route description. The route embedding is fused via cross-attention with the local map embedding generating the `map-informed route embedding`.

First, the input text is tokenized by generating a token with dimension $d_{\text{tok}} \in \mathbb{R}$ per word using a fixed-size vocabulary. Afterwards, an embedding per token is generated, either by projecting the token with a linear layer into the embedding dimension $d_{\text{emb}} \in \mathbb{R}^n$. For simplicity, we use a linear projection in our experiments. These token embeddings are aggregated per step using a mean pooling operation to obtain a single step representation. The resulting route embedding contains again n step embeddings, which can then subsequently be aligned with the learned embeddings of the local map. Experiments showed that this route embedding representation outperforms further aggregation of the step embeddings into a fixed-dimensional route representation.

Alternatively, the route embedding can be generated by a (pre-)trained language encoder, which can be trained on large amounts of text data and therefore, may provide a more robust representation of the route description.

Route Integration Strategies We use cross-attention to fuse both modalities, i.e., the route embedding and the local map embedding. On the one hand, using the route embedding as query and the map embedding as key and value generates a `map-informed route embedding` (route-to-map attention (R2M)). This is symbolized in the `Route2Map Alignment` block in fig. 2 by the orange border around the blue route embedding. The resulting map-informed route embedding can be used as additional context for the motion planner.

As an alternative design, the map embedding can be used as query and the route embedding as key and value, obtaining a `route-informed map embedding` (map-to-route attention (M2R)). In the latter alignment strategy, the final embedding remains the same shape as the local map embedding and can therefore directly replace the original map

embedding.

Route Reconstruction Loss In addition to the main planning objective, we use an auxiliary route reconstruction head that predicts the binary route encoding, i.e., whether a lane belongs to the route. The head is trained with a binary cross-entropy loss. Models that use this auxiliary objective are denoted with an 'L' in their name.

Spatial Attention Bias We additionally evaluate a spatial bias in the cross-attention of the `Route2Map Alignment` block. The bias is computed from the distance between each lane and the next routing step and is added to guide attention toward spatially relevant lanes. Models that use this modification are denoted with a 'B' in their name.

4 Evaluation

To evaluate the language comprehension abilities, we condition a flow-matching motion planner, FlowDrive [20], on the route descriptions following the proposed alignment strategy (see section 3). Specifically, we choose the purely learned FlowDrive version without any post-processing optimization to better analyze the influence of the language conditioning on model performance. The model originally utilizes HD-map-level route information, which is a binary encoding of whether a lane belongs to the route or not. Since this is the most information rich conditioning signal a model could receive, we state its results as an upper bound for the performance of a language-conditioned model (see table 1). However, this information is not available in mapless settings; therefore, for fair comparison, we train different modified baselines with route information available from a routing system.

We compare different baselines with different language-conditioned FlowDrive modifications on the nuPlan Val14 [4] split. The results are obtained from the closed-loop reactive simulation, which is the most realistic evaluation setting for motion planners in nuPlan.

4.1 Metrics

For performance comparison, especially in the context of routing, the standard CLS metric of nuPlan does not account for the route compliance of the model. Therefore, we utilize the following three metrics to evaluate the models:

Closed-Loop Score Reactive (CLS-R) The CLS-R metric evaluates the overall planning performance of the model in a closed-loop reactive simulation, considering multiple aspects of driving quality. It is the main metric when comparing motion planners in the nuPlan benchmark and is calculated as follows:

$$\text{CLS} = \left(\prod_{m \in M} \text{score}_m \right) \times \frac{\sum_{w \in W} \text{weight}_w \text{score}_w}{\sum_{w \in W} \text{weight}_w} \tag{1}$$

where $M = \{\text{collision, drivable area, direction, progress}\}$ is the set of all multiplicative metrics, $W = \{\text{time-to-collision, speed, comfort}\}$ is the set of weighted metrics, and

score_m and score_w are the scores for each multiplicative and weighted metric, respectively. However, CLS-R does not explicitly evaluate the route compliance of the model, which is crucial in real driving. Therefore, we also utilize the metrics from [23] to account for the model's route compliance.

Navigation Compliance (NAVI) The NAVI metric evaluates the route compliance of the model by checking if the last trajectory point T_{last} lies within a lane that belongs to the route. Since this metric was originally designed for a trajectory forecasting setting, we adapt it to the closed-loop reactive simulation by checking the final position of the simulated trajectory T after the simulation horizon:

$$\text{NAVI}(T) = \mathbf{1}_{\mathbf{L}_{\text{route}}}(T_{\text{last}}) \tag{2}$$

with L_{route} being the set of lanes belonging to the route.

Navigation based CLS (NAVI-CLS) Since the CLS metric does not account for route compliance, NAVI-CLS is proposed to evaluate the overall performance of the model while accounting for route compliance by simply adding the NAVI metric as a multiplicative metric to the set M. For the reactive simulation, the metric results in NAVI-CLS-R. The metric is a simplification of the Controllability Measure [23] that neglects the influence of multiple navigation options.

4.2 Experiments

We conduct experiments with different language-conditioned FlowDrive (FD-LG) modifications and compare them to different baselines (FD-B) on the nuPlan Val14 split. The map-informed route embedding is fused into the FlowDrive model by aggregating through pooling and adding it to the original map embedding used as map context. The baselines are different versions of FlowDrive, which do not utilize the binary route encoding but receive intermediate positions where routing steps take place instead. As already introduced in section 3, models using the route reconstruction loss and/or the spatial attention bias are denoted by 'L' or 'B', respectively. We evaluate models giving different numbers of routing steps (2 or 3) as condition. Here, we ignore the first step (see fig. 1), since it remains constant in each route description ('depart').

In table 1 we show the results for different route-conditioned baselines and language-conditioned models with the R2M alignment strategy. Overall, no model reaches the performance of the upper bound model (FlowDrive epoch 100) with HD-map-level binary route encoding, which is expected since this is the most information rich conditioning signal given the model architecture. However, most language-conditioned models show faster learning progress regarding driving performance (CLS-R) in the early epochs (see epoch 50), but then seem to reach a performance plateau or even decrease in performance. This trend is especially present in the modified baseline models. Here, the models decrease in performance with respect to the CLS scores, while the NAVI score increases, indicating that the models learn to better comply with the route, but at the cost of worse overall driving performance. In contrast, the language-conditioned models seem to maintain a better balance between driving performance and route compliance.

Table 1: Experiments on the nuPlan Val14 [4] evaluation split: Evaluation of language instructed models (R2M) and modified route-informed baselines with different configurations. Best results per metric and per epoch are in **bold**, second-best are underlined (orignal model is not included). We report the results for 50 and 100 training epochs, respectively.

Type	**Planner**	**Configuration**			**Val14** (50 / 100 epochs)		
		#steps	Aux. loss	Bias	CLS-R↑	NAVI-CLS-R↑	NAVI↑
Expert	Log-replay	-	-	-	80.32	-	-
Baseline	FlowDrive	-	-	-	82.44 / 85.84	80.33 / 84.07	97.05 / 97.67
Modified Baselines	FD-B-G	1 (goal)	✗	-	80.23 / 79.99	77.22 / 78.0	94.28 / 95.44
	FD-B-2	2	✗	-	81.73 / 79.82	79.15 / 77.31	94.1 / 94.01
	FD-B-L2	2	✓	-	82.03 / 81.25	78.9 / 78.87	94.54 / 94.72
	FD-B-3	3	✗	-	82.42 / 79.37	80.03 / 77.83	95.08 / **96.24**
	FD-B-L3	3	✓	-	83.04 / 81.1	80.27 / 79.22	95.17 / **96.24**
Language instructed models (R2M)	FD-LG-2	2	✗	✗	82.03 / 82.14	80.16 / 80.21	**95.97** / 95.71
	FD-LG-L2	2	✓	✗	82.47 / 82.46	79.84 / 80.17	95.17 / 95.26
	FD-LG-B2	2	✗	✓	**83.98** / 83.40	**81.37** / 81.16	95.53 / 95.44
	FD-LG-LB2	2	✓	✓	82.66 / 80.18	80.01 / 78.22	95.62 / 95.62
	FD-LG-3	3	✗	✗	82.79 / 83.05	80.19 / 80.88	95.53 / 95.62
	FD-LG-L3	3	✓	✗	82.65 / **84.39**	79.77 / **81.94**	94.81 / 95.97
	FD-LG-B3	3	✗	✓	83.02 / 82.55	80.35 / 80.51	95.17 / 95.89
	FD-LG-LB3	3	✓	✓	82.50 / 82.14	80.11 / 79.70	**95.97** / 95.61

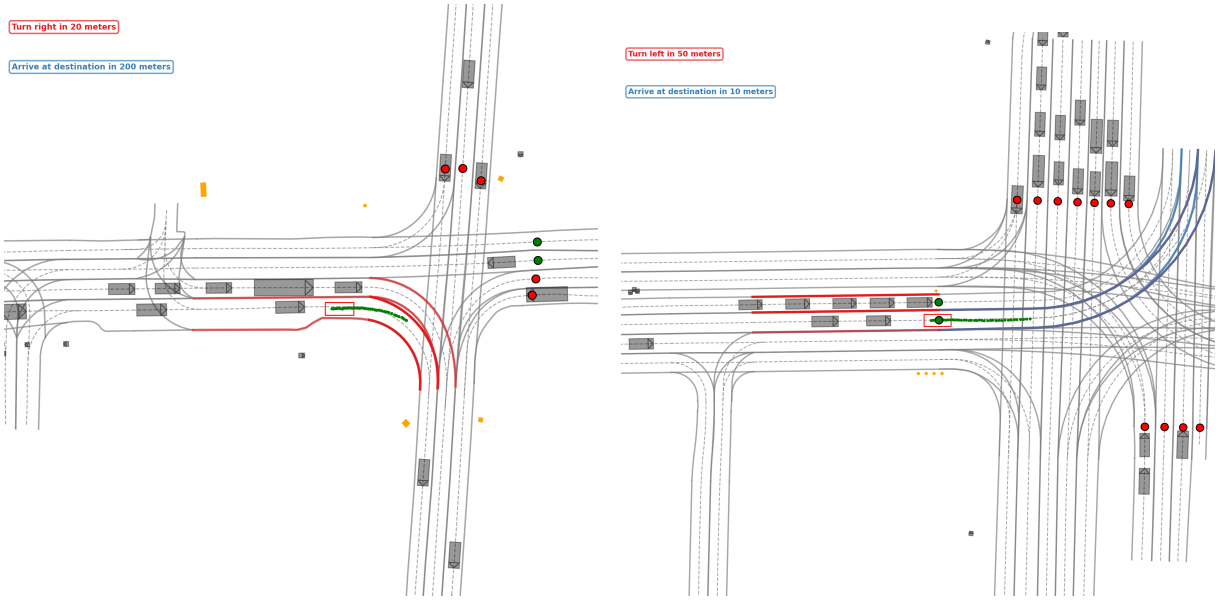
The baseline models with 3 routing steps outperform those with 2 steps, while both configurations benefit from the route reconstruction loss, indicating improved representation learning from the intermediate route positions. The FD-B-(L)3 models even achieve the highest NAVI score among all adapted models, which comes with a trade-off since their NAVI-CLS-R scores are lower than those of most of the language-conditioned models.

Given the language-conditioned models, there is no clear trend regarding the influence of the route reconstruction loss or the spatial attention bias on the performance. The model with 2 routing steps seem to benefit from the attention bias, while the model with 3 steps seems to benefit from the route reconstruction loss. In both step configurations the models do not generally gain from their combination, which indicates that the two modifications do not necessarily complement each other. Overall, the best language-conditioned models (2 and 3 steps) show a better performance compared to the baselines, especially in terms of the CLS-R and the NAVI-CLS-R score, which indicates that the language conditioning helps to better balance the driving performance and the route compliance of the model.

4.3 Attention Visualization

To analyze the route-map alignment of the model, we visualize the attention weights of the cross-attention in the `Route2Map Alignment` block. In fig. 3 we show the attention visualization for two example scenes. The figures highlight the 3 most attended lanes (above a certain threshold) for each route step, with the color intensity indicating the

attention weight.



(a) Right turn scenario: Token `8f72d629`

(b) Left turn scenario: Token `653f122d`

Figure 3: Route-to-lane attention visualization for two example scenes obtained from model `FD-LG-B2`. The model correctly attends to lanes consistent with the instructed route.

In the right turn scenario (see fig. 3a), the model correctly attends to the lanes corresponding to the right turn following the first route step. The attention of the second step is not above the threshold yet; thus it is not visualized. Analogously, in the left turn scenario (see fig. 3b), the model attends to the lanes corresponding to the two route steps, which are a left turn at the intersection and the arrival at the destination right after the turn. The attended lanes highlight the correct route to take by the model, indicating that the model has learned to align the route description with the local map representation in these scenarios.

4.4 Longer Routing

In addition to the standard 15-second evaluation horizon, we also evaluate the models on a longer 60-second horizon to analyze long-term route compliance and planning performance. We argue that long-term simulation is especially crucial in routing evaluation since complex routing decisions may not be fully captured in a short horizon. The results show that the language-conditioned model maintains a higher NAVI score compared to the baselines, indicating better route compliance over longer horizons. Especially compared to the 15-second evaluation, the models with language conditioning show superior performance compared to the baselines. However, all scores decrease compared to the shorter horizon due to the increased difficulty of long-term planning, but the language-conditioned model still outperforms the baselines, demonstrating the benefits of route instructions for long-term planning. In the long-term planning setting, the models seem to benefit from the route reconstruction loss, guiding the model to learn a better route

Table 2: Experiments on the nuPlan Val14_long evaluation split: Evaluation of language instructed models (M2R) and modified route-informed baselines with different configurations. We report the results for 50 and 100 training epochs, respectively. Especially in long-term planning, the language-instructed models show superior performance compared to the baselines.

Type	**Planner**	**Val14_long** (50 / 100 epochs) CLS-R↑	NAVI-CLS-R↑	NAVI↑
Modified Baselines	FD-B-2	62.12 / 54.46	55.91 / 47.9	83.01 / 82.29
	FD-B-L2	63.69 / 56.78	56.59 / 50.91	82.02 / 84.08
	FD-B-3	62.49 / 51.74	55.79 / 46.43	85.06 / 87.12
	FD-B-L3	64.57 / 56.71	57.82 / 51.06	83.99 / 86.67
Language instructed models (R2M)	FD-LG-2	65.29 / 61.74	59.23 / 56.46	85.6 / 86.94
	FD-LG-L2	65.96 / 60.93	**61.16** / 55.96	**88.01** / 87.66
	FD-LG-B2	**66.90** / 64.51	60.36 / 59.23	85.42 / 86.85
	FD-LG-LB2	66.06 / 54.8	60.69 / 49.73	87.12 / 87.75
	FD-LG-3	64.93 / 66.36	58.73 / 60.53	86.23 / 87.39
	FD-LG-L3	62.76 / **66.61**	56.46 / **61.07**	84.7 / 86.76
	FD-LG-B3	62.74 / 61.49	56.51 / 55.20	83.99 / 85.06
	FD-LG-LB3	65.52 / 59.06	59.19 / 53.94	85.87 / **88.19**

representation. Notably, the models with both route reconstruction loss and spatial attention bias (after 100 epochs) achieve the highest NAVI scores in the long-term planning evaluation. This indicates that the two modifications may complement each other in terms of route understanding.

4.5 Ablations

Following the main experiments, we conduct ablation studies to analyze the influence of route information and different language integration strategies on the performance.

4.5.1 No and Wrong Routing

To analyze the influence of route information on model performance, we first compare the original FlowDrive model with a version without route information and, second, compare a language-instructed model with correct versus incorrect route information (see table 3).

The version without route information is implemented by removing the binary route encoding.

The results show that the model without route information performs significantly worse than the original model, which indicates that the route information is crucial for the planning performance as well as the route compliance of the model. This is especially highlighted by the difference in the scores from epoch 50 to epoch 100, where the model without route information is generally decreasing in performance, while the original model is further improving by leveraging the route information.

The language-instructed model with incorrect route information is implemented by instructing the model with only u-turn maneuvers during evaluation. The results show that the model with incorrect route information performs significantly worse than the

Table 3: Ablation on the nuPlan Val14 evaluation split: First, comparison of the original FlowDrive model with a version without route information and second, the language-instructed FD-LG-L3 compared to itself with wrong route instructions. The performance changes in compared to the corresponding baseline are expressed as percentages, with red indicating a decline and blue representing improvement.

Planner	**Val14** (50 / 100 epochs) CLS-R↑	NAVI-CLS-R↑	NAVI↑
FlowDrive [20]	82.44 / 85.84	80.33 / 84.07	97.05 / 97.67
FlowDrive without route	82.65 (+0.3%) / 81.77 (-4.7%)	78.84 (-1.9%) / 78.64 (-6.5%)	93.2 (-4.0%) / 93.47 (-4.3%)
FD-LG-L3 (see table 1)	82.65 / 84.39	79.77 / 81.94	94.81 / 95.97
FD-LG-L3 wrong route	81.33 (-1.6%) / 82.09 (-2.7%)	77.72 (-2.6%) / 79.13 (-3.4%)	94.1 (-0.7%) / 94.72 (-1.3%)

same model with correct route information, which indicates that the model relies on the route description. The performance decrease is especially evident in the NAVI-CLS-R score, underlining its relevance not only for route compliance but also for the overall performance of the model.

4.5.2 Map-informed Route Embedding vs. Route-informed Map Embedding

As introduced in the route integration strategies in section 3, there are two strategies to align the route embedding with the local map embedding. The first strategy results in a map-informed route embedding by applying route-to-map attention (R2M), while the second strategy results in a route-informed map embedding by applying map-to-route attention (M2R). So far, we have only evaluated the first strategy. For completeness, we also evaluate the M2R strategy (see table 4) and compare it to the R2M strategy (see table 1). The route-informed map embedding is, in this case, fused into the FlowDrive model by simply replacing the original map embedding.

Table 4: Experiments on the nuPlan Val14 evaluation split. Ablation study of using route-informed map embeddings (M2R) contrary to map-informed route embeddings using R2M (see table 1).

Type	**Planner**	**Configuration** #steps	Aux. loss	Bias	**Val14** (50 / 100 epochs) CLS-R↑	NAVI-CLS-R↑	NAVI↑
M2R	FD-LG-2	2	✗	✗	81.95 / 80.84	79.44 / 78.96	95.08 / 95.8
	FD-LG-L2	2	✓	✗	82.04 / 81.81	78.73 / **79.81**	94.45 / **96.06**
	FD-LG-B2	2	✗	✓	82.29 / 81.15	79.6 / 78.75	**95.44** / 95.17
	FD-LG-LB2	2	✓	✓	**83.4** / **81.7**	**80.59** / 79.63	95.08 / 95.89

The table shows the results for 2 routing steps. In general, the R2M models outperform the M2R models with respect to the driving scores CLS-R and NAVI-CLS-R, while the M2R models show slightly better route compliance with respect to the NAVI score. However, the most relevant metric for the overall performance of the model is the NAVI-CLS-R score, which includes the NAVI score as a multiplicative factor. While the FD-LG-LB2 (M2R) model achieves overall better results than the FD-LG-LB2 (R2M) model, the best R2M model (FD-LG-B2) outperforms the best M2R model (FD-LG-LB2) by a significant margin. Comparing the metrics after 50 epochs suggests a faster learning

rate for the R2M models.

5 Conclusion and Future Work

In this work, we introduced a language-based routing framework for mapless motion planning and a corresponding nuPlan dataset extension with 60-second natural language route descriptions for 1 000 000 training scenarios and the Val14 validation split. Our method aligns route text with local map embeddings via cross-attention and conditions a flow-matching planner on this fused representation. Across closed-loop reactive simulation, language-conditioned models achieve a better trade-off between driving quality and route compliance than route-conditioned baselines that rely on intermediate positions only. In particular, the results indicate faster early learning and stronger long-horizon navigation compliance, while ablations confirm that route information is essential and that the route-to-map alignment strategy is generally more effective for overall performance than map-to-route alignment.

Future work should focus on stronger language representations and richer instruction settings. While we used a simple linear projection for route text in this study, replacing it with a pretrained language encoder could improve semantic robustness, especially for longer, noisier, or more diverse route descriptions. In addition, applying the proposed language-based routing interface to end-to-end driving models is a promising direction since these approaches naturally perform map-less planning.

Acknowledgments

The authors gratefully acknowledge the computing time provided on the high-performance computer HoreKa by the National High-Performance Computing Center at KIT (NHR@KIT). This center is jointly supported by the Federal Ministry of Education and Research and the Ministry of Science, Research and the Arts of Baden-Württemberg, as part of the National High-Performance Computing (NHR) joint funding program (`https://www.nhr-verein.de/en/our-partners`). HoreKa is partly funded by the German Research Foundation (DFG).

References

[1] Hidehisa Arai, Keita Miwa, Kento Sasaki, Kohei Watanabe, Yu Yamaguchi, Shunsuke Aoki, and Issei Yamamoto. CoVLA: Comprehensive Vision-Language-Action Dataset for Autonomous Driving. In *WACV*, 2025.

[2] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, 2020.

[3] Holger Caesar, Juraj Kabzan, Kok Seang Tan, Whye Kit Fong, Eric Wolff, Alex Lang, Luke Fletcher, Oscar Beijbom, and Sammy Omari. NuPlan: A closed-loop ML-based planning benchmark for autonomous vehicles. *arXiv preprint arXiv:2106.11810*, 2022.

[4] Daniel Dauner, Marcel Hallgarten, Andreas Geiger, and Kashyap Chitta. Parting with Misconceptions about Learning-based Vehicle Motion Planning. In *CoRL*, 2023.

[5] Daniel Dauner, Marcel Hallgarten, Tianyu Li, Xinshuo Weng, Zhiyu Huang, Zetong Yang, Hongyang Li, Igor Gilitschenski, Boris Ivanovic, Marco Pavone, Andreas Geiger, and Kashyap Chitta. NAVSIM: Data-Driven Non-Reactive Autonomous Vehicle Simulation and Benchmarking. In *NeurIPS*, 2024.

[6] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *CoRL*, 2017.

[7] Kanishkha Jaisankar and Sunidhi Tandel. Align2Act: Instruction-Tuned Models for Human-Aligned Autonomous Driving. *arXiv preprint arXiv:2510.10503*, 2025.

[8] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

[9] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

[10] Dennis Luxen and Christian Vetter. Real-time routing with OpenStreetMap data. In *ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2011.

[11] Yunsheng Ma, Can Cui, Xu Cao, Wenqian Ye, Peiran Liu, Juanwu Lu, Amr Abdelraouf, Rohit Gupta, Kyungtae Han, Aniket Bera, et al. Lampilot: An open benchmark dataset for autonomous driving with language model programs. In *CVPR*, 2024.

[12] Jiageng Mao, Yuxi Qian, Junjie Ye, Hang Zhao, and Yue Wang. Gpt-driver: Learning to drive with gpt. *arXiv preprint arXiv:2310.01415*, 2023.

[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*, 2016.

[14] Katrin Renz, Long Chen, Elahe Arani, and Oleg Sinavski. Simlingo: Vision-only closed-loop autonomous driving with language-action alignment. In *CVPR*, 2025.

[15] Parthib Roy, Srinivasa Perisetla, Shashank Shriram, Harsha Krishnaswamy, Aryan Keskar, and Ross Greer. doScenes: An Autonomous Driving Dataset with Natural Language Instruction for Human Interaction and Vision-Language Navigation. In *International Conference on Intelligent Transportation Systems (ITSC)*, 2025.

[16] Hao Shao, Yuxuan Hu, Letian Wang, Guanglu Song, Steven L. Waslander, Yu Liu, and Hongsheng Li. LMDrive: Closed-Loop End-to-End Driving with Large Language Models. In *CVPR*, 2024.

[17] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *CVPR*, 2020.

[18] Shuhan Tan, Boris Ivanovic, Yuxiao Chen, Boyi Li, Xinshuo Weng, Yulong Cao, Philipp Krähenbühl, and Marco Pavone. Promptable Closed-loop Traffic Simulation. In *CoRL*, 2024.

[19] Ömer Şahin Taş and Royden Wagner. Words in Motion: Extracting Interpretable Control Vectors for Motion Transformers. In *ICLR*, 2025.

[20] Lingguang Wang, Ömer Şahin Taş, Marlon Steiner, and Christoph Stiller. FlowDrive: moderated flow matching with data balancing for trajectory planning. *arXiv preprint arXiv:2509.21961*, 2025.

[21] Yiru Wang, Zichong Gu, Yu Gao, Anqing Jiang, Zhigang Sun, Shuo Wang, Yuwen Heng, and Hao Sun. HiST-VLA: A Hierarchical Spatio-Temporal Vision-Language-Action Model for End-to-End Autonomous Driving. *arXiv preprint arXiv:2602.13329*, 2026.

[22] Dongming Wu, Wencheng Han, Yingfei Liu, Tiancai Wang, Cheng-zhong Xu, Xiangyu Zhang, and Jianbing Shen. Language prompt for autonomous driving. In *AAAI*, 2025.

[23] Hanfeng Wu, Marlon Steiner, Michael Schmidt, Alvaro Marcos-Ramiro, and Christoph Stiller. Closing the Navigation Compliance Gap in End-to-end Autonomous Driving. *arXiv preprint arXiv:2512.10660*, 2026.

[24] Yinan Zheng, Ruiming Liang, Kexin Zheng, Jinliang Zheng, Liyuan Mao, Jianxiong Li, Weihao Gu, Rui Ai, Shengbo Eben Li, Xianyuan Zhan, and Jingjing Liu. Diffusion-Based Planning for Autonomous Driving with Flexible Guidance. *arXiv preprint arXiv:2501.15564*, 2025.

[25] Julius Ziegler, Philipp Bender, Markus Schreiber, Henning Lategahn, Tobias Strauss, Christoph Stiller, Thao Dang, Uwe Franke, Nils Appenrodt, Christoph G Keller, et al. Making bertha drive-an autonomous journey on a historic route. *IEEE Intelligent Transportation Systems Magazine*, 2014.