

Multi-Modal Deep Learning Architecture Based on Edge-Featured Graph Attention Network for Lane Change Prediction

Petrit Rama^a and Naim Bajcinca^b

Department of Mechanical and Process Engineering, Rheinland-Pfälzische Technische Universität Kaiserslautern-Landau,
Gottlieb-Daimler-Straße 42, 67663 Kaiserslautern, Germany
{petrit.rama, naim.bajcinca}@rptu.de

Keywords: Lane Change, Maneuver Prediction, Deep Learning, Graph Neural Network, Autonomous Driving.


Abstract: Maneuver prediction, especially lane change maneuver, is of critical importance for the safe navigation of autonomous vehicles. Although benchmark datasets exist for trajectory prediction, datasets specifically tailored for maneuver prediction are rare. This is particularly true for lane change prediction. To address this gap, in the present paper, an instrumented test vehicle is used to collect, process and label lane change maneuvers across various traffic scenes. The resulting dataset, referred to as WylonSet, consists of front-facing camera images, area-view camera images, vehicle state data and lane information. Thereby, over 400 driving sessions are collected and labeled, including approximately 500 lane change maneuvers, laying the foundation for our study. The main motivation behind this work is to analyze and predict lane change maneuvers for the ego-vehicle in urban traffic scenarios using deep learning models. In this study, a novel multi-modal deep learning architecture is proposed, comprising different modules to extract important features from the collected data. The visual module is built using Convolutional Neural Networks (CNNs) to capture features from all camera images, while the interaction module utilizes Graph Neural Networks (GNNs) to capture spatial features between detected entities in the traffic scene. The state module utilizes vehicle state data, while the lane module utilizes lane features. All these features are tracked in time using the temporal module of Recurrent Neural Networks (RNNs). The proposed architecture is trained and validated on WylonSet. Finally, the proposed learning architecture is implemented, and the resulting model for lane change prediction of the ego-vehicle is evaluated in different driving scenes and traffic densities.


1 INTRODUCTION

Human driving is inherently hierarchical, aligned with discrete decision-making layers that correspond to specific maneuvers such as lane changing, overtaking and lane keeping. Incorporating this hierarchical structure into machine driving allows more manageable solutions to the complex problem of autonomous driving. Consequently, lane change prediction becomes more challenging due to the uncertainty in the control policies of individual agents. Navigating traffic environments, is inherently feedback-structured, enabling the ego-vehicle and other vehicles to reduce the likelihood of collisions by adjusting their speed and position accordingly. This demands modeling approaches that consider holistic understanding of environments and the awareness of interactions among the involved traffic agents. Thus, anticipating and recom-

mending lane change maneuvers can improve traffic safety by allowing vehicles to proactively respond to potentially dangerous situations.

Autonomous driving encompasses various tasks including object detection, semantic segmentation, scene understanding, maneuver planning, trajectory prediction and vehicle control. While benchmark datasets are available for many of these tasks (Geiger et al., 2013; Cordts et al., 2016; Yu et al., 2018; Huang et al., 2018), maneuver planning datasets are rare and often not specifically tailored for high-level maneuver prediction. This rarity extends to lane change maneuvers too, where processing trajectories and other motion cues is required to infer such driving maneuvers. Explicitly labeled maneuvers are provided for intention prediction in the BLVD dataset (Xue et al., 2019), and learn driver behaviors in the HDD dataset (Ramanishka et al., 2018). However, lane change maneuvers are limited in number in both datasets and not sufficient to comprehensively understand lane change

^a  <https://orcid.org/0000-0003-3925-7869>

^b  <https://orcid.org/0000-0002-1660-4859>

behaviors across different traffic scenes. To address this gap, in the present work, an instrumented test vehicle is utilized to collect, process and label lane change maneuvers in various traffic scenarios. The introduced dataset, named WylonSet, consists of high-resolution front-facing camera and area-view camera images, vehicle state data and lane information. It serves as a foundation for our study, to analyze and predict lane change behaviors using deep learning models in different traffic scenarios.

Generally, the research landscape of lane change prediction is dominated by deep learning models, mainly due to the availability of data and advancements in hardware capabilities. Lee et al. (Lee et al., 2017) proposes a novel framework that first builds a bird's-eye view of the traffic scene and utilizes Convolutional Neural Networks (CNNs) to perceive it, enabling lane change prediction for surrounding vehicles, including left cut-ins, right cut-ins or lane keep maneuvers. Wei et al. (Wei et al., 2019) introduces an end-to-end lane change behavior detection model using the front-facing camera images and Inertial Measurement Unit (IMU) data, leveraging Deep Residual Neural Network. The paper (Izquierdo et al., 2021) proposes an architecture based on CNNs to detect and predict lane change and lane keep maneuvers, based on vehicle motion histories, the environment context and the interaction between traffic agents.

Another class of models that is relevant to the present paper are Graph Neural Networks (GNN). Graph models are used widely in maneuver prediction, recently including lane change prediction. The main reasons that graph structures gained attention are because of their versatility to accommodate diverse sets of detected entities and their adaptable structures within various deep learning architectures. But, mainly, graphs inherently capture interactive features, making them well-suited for interaction-aware approaches. These approaches are crucial for motion prediction systems, where the movement of each participant significantly impacts the movement of others. GNNs have emerged as powerful models to capture the spatial interaction within such graphs. Adopting GNNs for modeling traffic scenes as graphs has been empirically proved to increase the accuracy of trajectory prediction by Diehl et al. in (Diehl et al., 2019). GRIP (Li et al., 2019) represents the interaction between traffic agents in the form of a graph, using GNNs to capture spatial features to predict trajectories for observed agents, not lane change maneuvers. Similarly, Pan et al. (Pan et al., 2020) proposes an architecture based on GNNs, Long Short-Term Memory networks and attention mechanism to model the problem as a spatio-temporal graph and predict lane

change trajectories. Liang et al. (Liang et al., 2020) encodes the map as a graph, and uses graph convolutions to capture complex topological dependencies, to predict multi-modal trajectories.

The present paper extends the research conducted in (Rama and Bajcinca, 2022; Rama and Bajcinca, 2023) by introducing a dataset and a multi-modal deep learning architecture designed for analyzing and predicting lane changes of ego-vehicles. The architecture incorporates visual, interaction, state, lane and temporal features. Visual features and detected entities from the traffic scene are extracted using CNNs from camera images. Adopting an interaction-aware approach, the architecture models spatial interactions among these detected traffic entities as scene graphs, whereby nodes represent detected traffic entities, while edges represent the relative interaction among them. Such graphs serve as inputs of GNNs for learning spatial features. Finally, vehicles state data and lane information, with features extracted from the aforementioned modules, are tracked in time using Recurrent Neural Networks (RNNs), enabling the capture of temporal dynamics relevant to lane change maneuver classification. The main contributions of the paper include:

- Utilizing surround area-view cameras to extract visual features and detect diverse traffic entities, modeling the surrounding view of the ego-vehicle and the interaction as one large scene graph;
- Proposal of a novel multi-modal deep learning architecture based on CNNs, GNNs and RNNs, and conducting an ablation study to analyze the impact of each module on lane change predictions;
- Optimizing the utilization of sparse visual features of interaction graphs for scrutinizing and enhancing the accuracy of the lane change maneuver prediction in different traffic scenarios.

2 METHODOLOGY

The decision-making process in urban environments is highly interactive, influenced by surrounding traffic agents, vehicle dynamics and lane information. Given these complexities, the methodology proposed in this work follows a multi-modal feature extraction approach from various inputs. These modalities include the visual, interaction, state, lane and temporal modules. Assuming the goal maneuver from the global path planning module is to drive straight, the aim is to predict lane keep, left lane change or right lane change maneuvers based on the aforementioned input features and traffic constraints.

The system detects visual information and features of traffic entities using a state-of-the-art computer vision algorithm, deployed on the front-facing camera and area-view cameras. The visual features captured by all cameras provide a visual perspective of the nearby detected entities. YOLOv7 (Wang et al., 2022), pre-trained on the COCO (Lin et al., 2014) is employed to detect entities and extract visual features.

This work adopts an interaction-aware motion model by representing the problem as an interaction scene graph. A graph is built for every image frame captured by the cameras, whereby nodes represent detected entities, while the edges represent the relative spatial distance in the image space between entities. Separate graphs generated from each camera are merged, with the ego-node serving as the common node. Graph modeling offers the flexibility to treat the problem as a dynamic system by adding or removing nodes, edges and features, to reflect the varying numbers of detected agents during driving.

The vehicle's state data and lane information obtained from the CAN bus are also used as input, providing the insights of the internal dynamics of the ego-vehicle and the road structure. They include signal data for the steering, acceleration, braking, yaw, as well as lane markings, their type and color, curvature, offset, etc. These inputs, combined with visual and interaction features, are tracked in time over a specific time-window, providing the navigation module of the ego-vehicle with comprehensive data to make the final decision, which includes Left Lane Change (*LLC*), Lane Keep (*LK*) and Right Lane Change (*RLC*).

2.1 Problem Formulation

The model receives historical sequences from front-facing and area-view cameras, interaction graphs, vehicle state data and lane features as inputs. These sequences are observed for $t = [-T_w : 0]$, where T_w is the observation time. The objective is to predict the output probability distribution y at $t + T_p$ for maneuvers $\{LLC, LK, RLC\}$ of the ego-vehicle, where T_p represents the prediction time step in the future.

The traffic scene is modeled as an interaction graph $\mathcal{G}^{(t)} = (\mathcal{V}^{(t)}, \mathcal{E}^{(t)}, \mathcal{X}_{\mathcal{V}}^{(t)}, \mathcal{X}_{\mathcal{E}}^{(t)})$, for timestep t . The node set \mathcal{V} represents n detected traffic entities, with $n + 1$ total nodes in the traffic graph, where the additional node represents the ego-vehicle. The edge set \mathcal{E} encodes the inverse Euclidean distance between centerpoints of detected entities in image space, so that entities closer to the ego-vehicle have a stronger influence on lane change prediction. The feature vector of the entire node set is denoted as $\mathcal{X}_{\mathcal{V}}$, while the feature vector of the entire edge set as $\mathcal{X}_{\mathcal{E}}$.

2.2 Model Architecture

The proposed deep learning architecture is shown in Fig. 1. The architecture is multi-modal and considers current inputs of visual, spatial, vehicle state data and lane information, together with past observations of these inputs for predicting lane change maneuvers.

First, image frames $I^{(t)}$ of timestep t from all cameras pass through multiple *CNN* layers for feature extraction, providing the architecture with visual information of the surrounding traffic. Visual features are tracked in time-window T_w using the module *RNN_v*:

$$v^{(t)} = CNN(I^{(t)}), \quad \mathcal{H}_v^{(t)} = RNN_v(v^{(t)}, \mathcal{H}_v^{(t-1)}), \quad (1)$$

where $\mathcal{H}_v^{(t)}$ is the output of the *RNN_v* at timestep t .

The same image frames $I^{(t)}$ are passed through YOLOv7 (Wang et al., 2022) algorithm to detect traffic entities and extract sparse visual features from the traffic environment. Two different interaction graphs are constructed, as shown in bottom-right part of Fig. 1.

BBox graph $\mathcal{G}_b^{(t)}$ is created using bounding box information as node features, wherein each node i in the graph $\mathcal{G}_b^{(t)}$ corresponds to only one detected objects and its features f_i , as described below:

$$f_i = [x_i, y_i, w_i, h_i, a_i, c_i], \quad (2)$$

where x, y are the centerpoint coordinates, w, h are width and height of the bounding box in pixels, a is the detection confidence and c is the class of the detected entity. Contrarily, patch graph $\mathcal{G}_p^{(t)}$ is built with the same structure as the BBox graph, but it uses the extracted visual features from the last layer of YOLOv7 as node features. These graphs are then passed through separate GNN modules to capture interactive features from separate graphs:

$$g_b^{(t)} = GNN_b(\mathcal{G}_b^{(t)}), \quad g_p^{(t)} = GNN_p(\mathcal{G}_p^{(t)}). \quad (3)$$

The ego-vehicle is represented in the interaction graphs as the ego-node, capturing the spatial interactive features from all nodes in the graph at timestep t . The output embedding of the ego-node $g_{b_0}^{(t)}$ is extracted from the transformed graph $g_b^{(t)}$, while the embedding of ego-node $g_{p_0}^{(t)}$ is extracted from the transformed graph $g_p^{(t)}$. These two vector embeddings and observed over time-window T_w using the interaction module *RNN_g*, similar to Eq. (1).

Vehicle state data $s^{(t)}$ and lane features $l^{(t)}$ at timestep t are also tracked over observed time-window T_w , using the respective *RNN* modules:

$$\mathcal{H}_s^{(t)} = RNN_s(s^{(t)}, \mathcal{H}_s^{(t-1)}), \quad (4)$$

$$\mathcal{H}_l^{(t)} = RNN_l(l^{(t)}, \mathcal{H}_l^{(t-1)}). \quad (5)$$

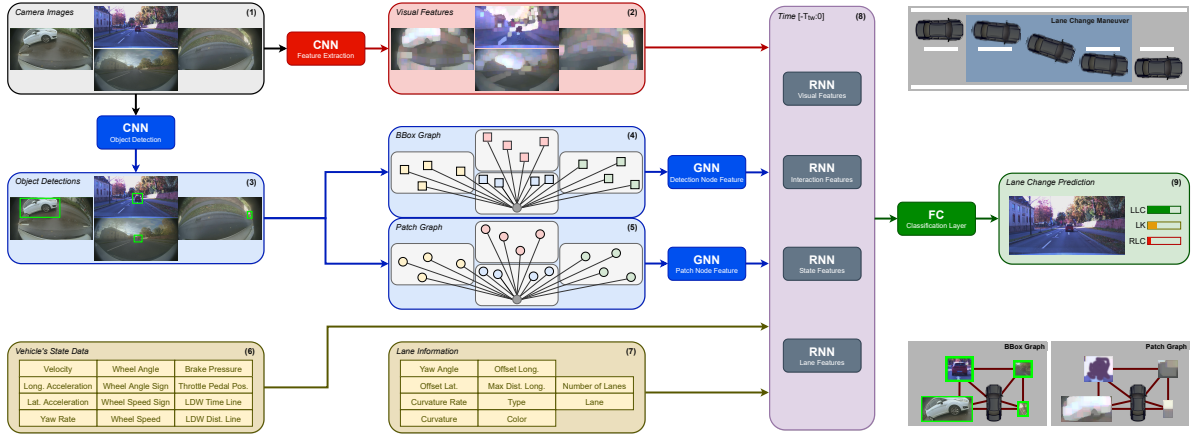


Figure 1: Network architecture of the proposed model for lane change prediction, (top-right) labeled lane change maneuver, and (bottom-right) interaction graphs of the detected bounding boxes and extracted visual features.

The hidden latent representation from all aforementioned modules, namely $\mathcal{H}_v^{(t)}$, $\mathcal{H}_g^{(t)}$, $\mathcal{H}_s^{(t)}$ and $\mathcal{H}_l^{(t)}$ at timestep t are concatenated and passed through a fully-connected classification layers $FC(\cdot)$:

$$\mathfrak{H}_{fc}^{(t)} = FC(\mathcal{H}_v^{(t)} \parallel \mathcal{H}_g^{(t)} \parallel \mathcal{H}_s^{(t)} \parallel \mathcal{H}_l^{(t)}). \quad (6)$$

The output representation $\mathfrak{H}_{fc}^{(t)}$ of the network is used to train the model in a supervised fashion to predict $\{LLC, LK, RLC\}$ of the ego-vehicle at $t + T_p$.

3 EXPERIMENTS

Experiments were conducted on a desktop: Ubuntu 18.04 with 2.2GHz Intel(R) Xeon(R) CPU E5-2698 v4, 256 GB RAM, Tesla V100-DGXS-32GB.

3.1 Dataset

For the present work, the in-house dataset WylonSet, which is specifically tailored for lane change maneuvers, has been collected utilizing an instrumented test vehicle. Drivers were instructed to drive normally, while adhering to traffic rules, signs, and speed limits. They were primarily directed to drive straight in all scenarios and avoid executing other turning maneuvers. Driving straight was also assumed as the goal to train the proposed deep learning model.

The dataset has been collected in various parts of the city of Kaiserslautern, in Germany, between October 2023 and February 2024, featuring diverse lane information, traffic densities and weather conditions. The dataset includes high-resolution front-facing camera images (30Hz with a resolution of 2048×864 pixels), area-view camera images (15Hz, 1280×800), vehicle state data and lane information

obtained from the CAN bus. The motion of the ego-vehicle is measured using the IMU, which records brake pressure, velocity, acceleration, yaw rate and steering wheel, among others. In addition, the CAN provides information about the lane markings, including yaw angle, latitudinal/longitudinal offset, curvature, and the type and color of the markings.

WylonSet has initially been preprocessed to ensure proper structuring for lane change behavior analysis. Sessions that contained missing or corrupt information were removed to ensure data integrity. Timestamps from front-facing camera images serve as the master clock for synchronizing area-view cameras, vehicle state data and lane features. Timestamping is performed using the RTMaps¹.

After processing and labeling, more than 400 driving sessions were obtained, with nearly 3 hours of driving videos and around 250,000 front-facing image frames. The dataset includes 315 right lane change maneuvers and 175 left lane change maneuvers. The density distribution and histogram of the main input data are shown in Fig. 2. The velocity distribution centers around 50 – 60km/h, reflecting typical driving speeds for urban and rural areas. This is also evident in Fig. 2b, showing a dominance of roads with two lanes. The Fig. 2c shows the specific lane driven by the ego-vehicle, with lane numbering from right to left. Lastly, in Fig. 2d, the histogram of the left lane marking type is shown, where: 0 is “no line”, 1 is “solid”, 2 is “dashed”, 3 is “sidewalk”, 4 is “grass”, 5 is “bot-dots”, 6 is “unknown” and 7 is “error”. The histogram for the types of right lane markings are very similar to those for the left lane markings.

Lane change maneuvers are labeled using turn indicators that mark the start and end of the lane change. As shown in the top-right part of Fig. 1, the turn indi-

¹<https://intempora.com/products/rmaps/>

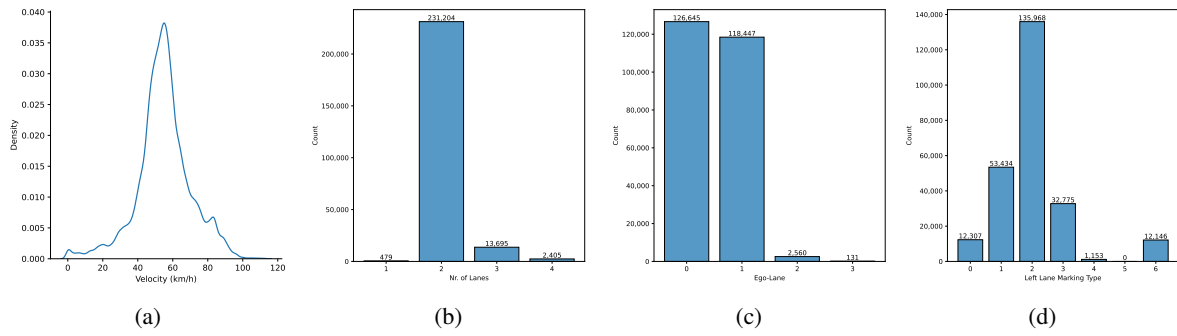


Figure 2: Density distribution of ego-vehicle data from CAN bus for (a) velocity. Histogram of (b) number of lanes in driven roads, (c) lane where ego-vehicle was driving (0 is the right-most lane), and (d) left lane marking type.

ator signal is initiated when the vehicle is in the lateral middle of the ego-lane and starts the movement towards the next lane, coinciding with the rotation of the steering wheel. The turn indicator is stopped, once the vehicle reaches the lateral middle of the next lane, completing of the lane change.

3.2 Model Implementation

The image frames are processed by the visual module V of the proposed architecture, implemented using $CNNs$. Frames from the front-facing camera and area-view cameras are resized to 256×128 pixels before feeding them into a two-layer 2D convolution to extract visual features from each frame of raw images. Each convolution layer uses a kernel size of 5 and is followed by a 2D max pooling layer with a kernel size of 2 and a stride of 2. The network employs the $ReLU$ activation function, a 0.2 dropout value and 32 hidden units per layer. The hidden visual representation is then flattened and passed through a linear layer to obtain the output visual representation for each frame.

The same image frames from all four cameras are also utilized for sparse visual extraction. YOLOv7, pre-trained on the COCO dataset (Lin et al., 2014), is employed to detect entities in a traffic scene. The detected sparse features are employed to construct the respective scene graphs. The BBox graph is constructed using the detected bounding box information inferred from YOLOv7, illustrated as the first graph in the bottom-right part of Fig. 1. Such graphs are enriched with node and edge features as described in Sec. 2.2, and are processed by the first interaction module I_p . Contrarily, the patch graph is constructed using the last feature layer of YOLOv7, by extracting the inner parts of detected entities based on the bounding box coordinates, illustrated as the second graph in the bottom-right part of Fig. 1. Patches of hidden visual representation are first resized to 16×16 pixels, passed through a CNN layer with a kernel size

of 4, followed by a max pooling layer with a kernel size of 2 and stride 2. The hidden representations are flattened and passed through a linear layer to produce node features. The edge features in this graph remain the same as in the BBox graph, encoding the inverse Euclidean distance. The patch graph is processed by the second interaction module I_p . Both graphs model the spatial interaction among entities, but with different visual features as node features. Each interaction module is implemented as a two-layer Edge-Featured Graph Attention Network (Wang et al., 2021).

The vehicle state values are processed by the state module S , while lane features are processed by the lane module L . Continuous values from vehicle state data and lane features are scaled between 0 and 1. Categorical values from vehicle state data and lane features are one-hot vector encoded. This processing step enables the integration of these features, feeding them directly into the temporal module.

The temporal module T tracks the hidden representations from each module, implemented using Gated Recurrent Units (GRUs). Separate GRUs are implemented for each module: GRU_v for visual features, GRU_g for interaction graphs, GRU_s for vehicle state data and GRU_l for lane features. Each GRU consists of a single layer with 32 hidden units and no dropout regularization. Such modules allow the architecture to capture temporal dependencies in the input.

All output features from the temporal modules are concatenated. Subsequently, they are passed through a classification layer, which is implemented as a two-layer Fully Connected (FC) network with 32 hidden units per layer. The first FC layer applies a $ReLU$ activation function, and the second FC layer produces the final classification, generating predictions for lane change maneuvers based on fused input features.

All input data are synchronized based on the front-facing camera images, which are captured at a rate of 30 frames per second. A single frame was used as a sampling step, resulting in a sampling rate of 1/30 of a second. For the experiments, the input data are pre-

Table 1: Main results show the impact of number of sessions in the performance of the trained model.

Sess.	Seq.	Acc	F1-S	ROC-S
75	≈ 21.000	76.32%	76.30%	78.24%
120	≈ 44.000	84.86%	84.88%	86.90%
175	≈ 64.000	88.15%	88.13%	90.61%
240	≈ 82.000	91.35%	91.37%	93.52%

pared as ordered sequences, with a sampling step of 4 and an observation time-window of $T_{tw} = 15$. This means that every fourth front frame, along with synchronized camera frames, state data and lane values, are taken to build a 15-timestep sequence of input.

The dataset is randomly split in 70% for training and 30% for validation. The architecture is implemented in Python, PyTorch (Paszke et al., 2019) and Deep Graph Library (DGL) (Wang et al., 2019). Adam optimizer is used with a learning rate of 0.001 and L2 regularization of 0.0001. The model is trained as a supervised classification task, minimizing the cross-entropy loss between the predicted outputs and labels. The model is evaluated using accuracy (Acc), F1 score (F1-S), ROC score (ROC-S). Moreover, the model is evaluated using per-class precision (P), recall (R), F1 score (F1-S), for three maneuver classes.

4 RESULTS

4.1 Main Results

The performance of the proposed model was evaluated on the WylonSet dataset. To facilitate hyperparameter tuning and examine the impact of dataset size on model performance, the dataset was incrementally enlarged, and the model was retrained and validated at each step. The results are summarized in Table 1, showing that increasing the dataset size led to a steady improvement in performance metrics. With 240 sessions (approx. 82,000 sequences), the model achieved an accuracy of 91.35%, an F1-score of 91.37% and a ROC score of 93.52%. While increasing the dataset size improved performance, the marginal gain diminished as the dataset grew, at the cost of longer processing, training, and evaluation times.

4.2 Ablation Study

To better understand and interpret the contribution of each module to the model's performance and analyze the problem of lane change maneuver prediction from different modalities, an ablation study was conducted. Modules are denoted as in the previous sections: V is the visual module, I_b the interaction module with

Table 2: Ablation results show the impact and contribution of each module in the performance of the trained model.

	Average		
	Acc	F1-S	ROC-S
$V \cdot I_b I_p \cdot S \cdot L \cdot T$	<i>91.35%</i>	<i>91.37%</i>	<i>93.52%</i>
$I_b I_p \cdot S \cdot L \cdot T$	92.39%	92.40%	94.29%
$V \cdot I_b \cdot S \cdot L \cdot T$	<u>91.62%</u>	<u>91.63%</u>	<u>93.72%</u>
$V \cdot I_p \cdot S \cdot L \cdot T$	<u>91.67%</u>	<u>91.67%</u>	<u>93.75%</u>
$V \cdot S \cdot L \cdot T$	<u>89.63%</u>	<u>89.65%</u>	<u>92.22%</u>
$V \cdot I_b I_p \cdot L \cdot T$	87.00%	87.00%	90.25%
$V \cdot I_b I_p \cdot S \cdot T$	85.13%	85.15%	88.85%

bounding boxes, I_p the interaction module with detection patches, S is the state module, L is the lane module and T is the temporal module. The results of this study, shown in Table 2, are based on experiments carried out on more than 100,000 sequences ($\approx 55\%$ for the *LK*, $\approx 25\%$ for *LLC*, and $\approx 20\%$ for *RLC*).

The architecture with all modules [$V \cdot I_b I_p \cdot S \cdot L \cdot T$] is used as a baseline model, which based on the validation results of 91.35% accuracy (*in italic*), is surprisingly not the best performing model. The visual module is a crucial module for decision-making, yet the model without the visual module outperformed the baseline model, with an accuracy of 92.39% (**in bold**). This improvement can be attributed to the fact that enough visual features are effectively encapsulated in the traffic graphs, which also integrate spatial interactions through their edges. Removing one of the interaction graphs slightly improved the results to 91.62% and 91.67%, respectively. However, removing both interaction graphs led to a more significant drop in performance, reducing accuracy to 89.63% (in underline). The largest decrease in performance occurred when the state or lane modules were removed, resulting in accuracy of 87.00% and 85.13%, respectively, highlighting their crucial role in accurate lane change prediction for the ego-vehicle.

To evaluate the model's performance across different lane change maneuvers, precision (*P*), recall (*R*) and F1-score (*F1-S*) were calculated for each class: *LLC*, *LK* and *RLC*. The detailed results are provided in Table 3. Generally, the *LK* maneuver achieves a higher F1 score compared to *RLC* and *LLC*, with an F1-score of 92.08%. The differences are not substantial, suggesting relatively balanced performance across all maneuvers. For *RLC* and *LLC*, the model achieved F1-scores of 90.93% and 90.19%, respectively. Conversely, recall values generally tend to be higher for *RLC*, and was particularly high (95.28%) in the model without the visual module, showing that the model is highly sensitive to *RLC* despite fewer data points. Lastly, *RLC* precision varied significantly depending on the inclusion of interaction modules.

Table 3: Ablation results show the impact and contribution of each module in the per-class performance of the trained model.

	Left Lane Change			Lane Keep			Right Lane Change		
	P	R	F1-S	P	R	F1-S	P	R	F1-S
$V \cdot I_b I_p \cdot S \cdot L \cdot T$	87.97%	92.51%	90.19%	93.22%	90.97%	92.08%	91.03%	90.82%	90.93%
$I_b I_p \cdot S \cdot L \cdot T$	92.73%	90.22%	91.46%	93.38%	91.94%	92.66%	88.32%	95.28%	91.67%
$V \cdot I_b \cdot S \cdot L \cdot T$	88.28%	92.43%	90.31%	92.12%	92.14%	92.13%	94.11%	87.91%	90.90%
$V \cdot I_p \cdot S \cdot L \cdot T$	90.11%	91.18%	90.64%	93.30%	91.21%	92.24%	89.42%	93.66%	91.49%
$V \cdot S \cdot L \cdot T$	87.31%	88.49%	87.89%	92.25%	88.86%	90.52%	85.79%	93.32%	89.40%
$V \cdot I_b I_p \cdot L \cdot T$	85.50%	83.97%	84.73%	88.11%	87.70%	87.90%	86.04%	89.25%	87.61%
$V \cdot I_b I_p \cdot S \cdot T$	87.61%	75.01%	80.82%	87.13%	89.09%	88.10%	75.32%	85.14%	79.93%



Figure 3: Visualization of model inference for lane change prediction.

4.3 Scenario Visualization

A qualitative evaluation of the model’s predictions was performed using visualizations from selected traffic scenarios, as shown in Fig. 3. The top-left section provides information about the session, model and lanes. The left-middle part and right-middle part shows the type and color of the detected lane markings. The tables on the bottom-left and bottom-right shows the CAN bus data for the lane marking. The upper-right table shows the main state data for the ego-vehicle. The middle part displays image frames from the front-facing and area-view cameras. The top part shows the prediction probability distribution for *LLC*, *LK*, *RLC* maneuver classes in the form of bars, which are color-coded based on their probability.

In the urban traffic scene depicted in Fig. 3, the model extracts visual features from the scene to detect vehicles, the bus and traffic signs, while also reading the ego-vehicle’s state values and lane features from CAN. For the timestep $t + T_p$, the model predicts an *LLC* with a probability of nearly 90%. This decision can be interpreted considering that the ego-vehicle’s goal is to continue straight, the ego-vehicle is moving

faster than the bus, and that the left lane is free.

5 CONCLUSION

This work introduces WylonSet, a lane change dataset and proposes a novel multi-modal deep learning architecture for analyzing and predicting lane change maneuvers for the ego-vehicle. The dataset comprises front-facing camera and area-view cameras, vehicle state data and lane information, with around 500 lane change maneuvers labeled across diverse urban scenes. The proposed architecture is based on CNNs for extracting visual features, GNNs for capturing spatial features from interaction graphs of traffic scenes, and RNNs for tracking over time these features, along with vehicle state values and lane information. The ablation study highlights the substantial impact of the interaction module on the model’s performance, demonstrating improved results even without the visual module. The primary limitation is the difficulty in directly comparing and assessing the proposed architecture against existing approaches.

ACKNOWLEDGEMENTS

This work is supported by the Federal Ministry for Digital and Transport (BMDV) of Germany in the scope of project AORTA (FKZ: 01MM20002).

REFERENCES

- Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., and Schiele, B. (2016). The cityscapes dataset for semantic urban scene understanding. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, Las Vegas, NV, USA, June 27-30, 2016*, pages 3213–3223.
- Diehl, F., Brunner, T., Truong-Le, M., and Knoll, A. C. (2019). Graph neural networks for modelling traffic participant interaction. In *2019 IEEE Intelligent Vehicles Symposium, IV 2019, Paris, France, June 9-12, 2019*, pages 695–701. IEEE.
- Geiger, A., Lenz, P., Stiller, C., and Urtasun, R. (2013). Vision meets robotics: The KITTI dataset. *Intl. Journal of Robotics Research (IJRR)*, 32(11):1231–1237.
- Huang, X., Cheng, X., Geng, Q., Cao, B., Zhou, D., Wang, P., Lin, Y., and Yang, R. (2018). The apollo-scape dataset for autonomous driving. In *2018 IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops, Salt Lake City, USA, June 18-22, 2018*, pages 954–960. IEEE.
- Izquierdo, R., Quintanar, Á., Lorenzo, J., Daza, I. G., Parra, I., Llorca, D. F., and Sotelo, M. Á. (2021). Vehicle lane change prediction on highways using efficient environment representation and deep learning. *IEEE Access*, 9:119454–119465.
- Lee, D., Kwon, Y. P., McMains, S., and Hedrick, J. K. (2017). Convolution neural network-based lane change intention prediction of surrounding vehicles for ACC. In *20th IEEE Intl. Conf. on Intelligent Transportation Systems, ITSC 2017*, pages 1–6. IEEE.
- Li, X., Ying, X., and Chuah, M. C. (2019). GRIP: graph-based interaction-aware trajectory prediction. In *2019 IEEE Intelligent Transportation Systems Conference, ITSC 2019, Auckland, New Zealand, October 27-30, 2019*, pages 3960–3966. IEEE.
- Liang, M., Yang, B., Hu, R., Chen, Y., Liao, R., Feng, S., and Urtasun, R. (2020). Learning lane graph representations for motion forecasting. In *ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part II*, volume 12347 of *Lecture Notes in Computer Science*, pages 541–556. Springer.
- Lin, T., Maire, M., Belongie, S., Bourdev, L. D., Girshick, R. B., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312.
- Pan, J., Sun, H., Xu, K., Jiang, Y., Xiao, X., Hu, J., and Miao, J. (2020). Lane-attention: Predicting vehicles' moving trajectories by learning their attention over lanes. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems, IROS 2020, Las Vegas, NV, USA, Oct. 24, 2020 - Jan. 24, 2021*, pages 7949–7956. IEEE.
- Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E. Z., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32: Annual Conf. on Neural Information Processing Systems 2019, NeurIPS 2019, Vancouver, BC, Canada*, pages 8024–8035.
- Rama, P. and Bajcinca, N. (2022). NIAR: interaction-aware maneuver prediction using graph neural networks and recurrent neural networks for autonomous driving. In *Sixth IEEE Intl. Conf. on Robotic Computing, IRC 2022, Italy, Dec. 5-7, 2022*, pages 368–375. IEEE.
- Rama, P. and Bajcinca, N. (2023). MALE-A: stimuli and cause prediction for maneuver planning via graph neural networks in autonomous driving. In *2023 IEEE 26th International Conference on Intelligent Transportation Systems, ITSC 2023, Bilbao, Spain, September 24-28, 2023*, pages 3545–3550. IEEE.
- Ramanishka, V., Chen, Y., Misu, T., and Saenko, K. (2018). Toward driving scene understanding: A dataset for learning driver behavior and causal reasoning. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, Salt Lake City, USA, June 18-22, 2018*, pages 7699–7707. IEEE Computer Society.
- Wang, C., Bochkovskiy, A., and Liao, H. M. (2022). Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In *IEEE/CVF Conf. on Computer Vision and Pattern Recognition, CVPR 2023*, pages 7464–7475. IEEE.
- Wang, M., Yu, L., Zheng, D., Gan, Q., Gai, Y., Ye, Z., Li, M., Zhou, J., Huang, Q., Ma, C., Huang, Z., Guo, Q., Zhang, H., Lin, H., Zhao, J., Li, J., Smola, A. J., and Zhang, Z. (2019). Deep graph library: Towards efficient and scalable deep learning on graphs. *CoRR*, abs/1909.01315.
- Wang, Z., Chen, J., and Chen, H. (2021). EGAT: edge-featured graph attention network. In *Artificial Neural Networks and Machine Learning - ICANN 2021 - 30th Intl. Conf. on Artificial Neural Networks, 2021*, volume 12891, pages 253–264. Springer.
- Wei, Z., Wang, C., Hao, P., and Barth, M. J. (2019). Vision-based lane-changing behavior detection using deep residual neural network. In *2019 IEEE Intelligent Transportation Systems Conference, ITSC 2019*, pages 3108–3113. IEEE.
- Xue, J., Fang, J., Li, T., Zhang, B., Zhang, P., Ye, Z., and Dou, J. (2019). BLVD: building A large-scale 5d semantics benchmark for autonomous driving. In *Intl. Conf. on Robotics and Automation, ICRA, Montreal, QC, Canada, May 20-24, 2019*, pages 6685–6691.
- Yu, F., Xian, W., Chen, Y., Liu, F., Liao, M., Madhavan, V., and Darrell, T. (2018). BDD100K: A diverse driving video database with scalable annotation tooling. *CoRR*, abs/1805.04687.