Vehicle Motion Prediction Based on environment Factors and Confidence Assessment

Lang Zhang[1]**, Shahabedin Sagheb**[1]
Trent Greer[1]**, Michael Shi**[1]**, Yulong Zhong**[1]
[1]Virginia Polytechnic Institute and State University, Blacksburg, VA, USA
{langzhang,shahab,michaelhs20,wtgreer,yulongzhong25}@vt.edu

Abstract

Prediction of driving behavior can help improve traffic safety in the real world. In this project we study the relationship between vehicles and driving environment to predict the future states of vehicles. Using multi-layer perceptron (MLP), Long short-term memory (LSTM), Gated recurrent units (GRUs), and Transformers we aim to predict future vehicle states from historical data as input. We test our model using the INTERACTION dataset. We provide various confidence scores that provide a metric for evaluating the accuracy of the predictions. Across multiple experimental conditions, our results show that LSTM and Transformers are better in predicting the future states of vehicles. We provide discussion on the strengths and shortcomings of our models and discuss approaches to improve performance. Our code is available at: https://github.com/langzhang2000/CS_5806_Machine_Learning_Project

Keywords
Motion Planning, Safety, Trustworthy AI

1 Introduction

Driving is a complex task in which human drivers may exhibit unexpected behaviors. These behaviors may pose danger to other drivers. Therefore, if the vehicle motion trajectory during driving can be predicted based on previous driving behaviors, road safety can be greatly improved [15]. In general, scholars mainly study the motion trajectory of a single vehicle or rely on the connection between the motion trajectory of multiple vehicles to make predictions [3]. However, driving routes are not only influenced by other vehicles, they are also highly dependent on traffic rules and road conditions, such as congestion conditions and road types [17]. Our research focuses on building a machine learning model to characterize the vehicle-vehicle relationship and vehicles-road environment. With this, we aim to achieve more accurate trajectory prediction. Ultimately, such an approach could lead to a safer driving trajectory. The ability to anticipate driving decisions and reacting accordingly can assist in reducing accidents [24]. Our intuition is to use the MLP, LSTM, GRUs, and Transformers to evaluate the human driver motion prediction problem. We note that other schors have use other Deep Learning (DL) methods such as Graph Neural Networks (GNN) to solve this problem as well [21]. Although there has been work in this domain we emphasize that the driving behavior of vehicles is multi-modal. While we consider the impact between vehicles, we also have to consider the impact of environmental factors [13]. To this end, we need to take into account the complex relationships between vehicles and their interactions with the road environment. The challenge is to deal with dynamic situations where multiple factors intersect [14].

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

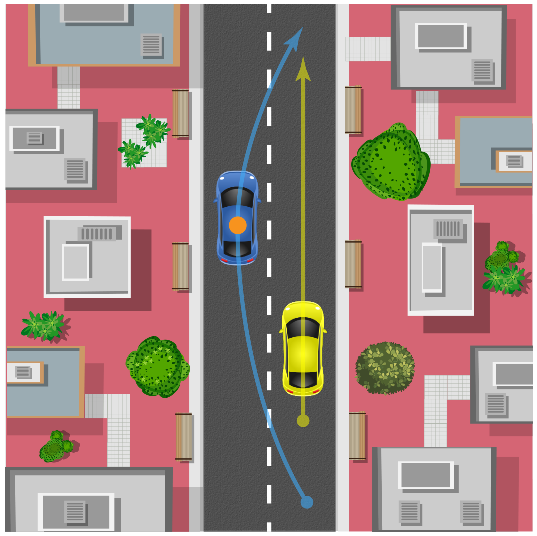


Figure 1: A human driven vehicle (blue) making a dangerous overtake maneuver. Blue line depicts the entire length of the trajectory $\xi_{\mathcal{HB}}$ from time-step $t = 0$ to $t = T$. The second human driven vehicle (yellow) follows a consistent trajectory from $t = 0$ to time-step $t = T$. We aim to predict the future states of the blue vehicle starting from $t + 1$ (shown with orange dot) until the end of the trajectory at $t = T$.

1.1 High-level problem formulation

In our project, given the temporal set of vehicle control features (e.g., acceleration, heading, position) and the environment features (e.g., road type, surrounding traffic) up to a certain point in time (i.e, $0, \cdots, t$), we aim to predict the vehicle motion in the next time steps (i.e., $t + 1, \cdots, T$) where T is the time horizon of a given trajectory ξ. In specific, we aim to explore whether a vehicle's past states will lead to unsafe behavior such as near-collision or minor accidents. Fig. 1 shows a brief use case of our problem. Similar to existing literature in this domain (e.g., [16]), we will use the INTERACTION Dataset [25] to test our algorithm. This will allow us to compare our results with the existing solutions. INTERACTION dataset provides the context dependant state information for multiple vehicles in different road conditions (roundabout, highway, intersection). It also includes data gathered from multiple countries which allows us to evaluate our hypothesis.

Our contributions are:

- Developing a forecasting model based on the features and relationships between each driver (agent) on the road.
- Using variety of confidence scores that to quantify the accuracy of the prediction.

1.2 Paper structure

We organize our paper using the following sections. In Section 2 we discuss the state-of-the-art for vehicle motions prediction. We provide a high-level overview of each group of work and discuss the strengh and shortcomings of the existing works. In Section 3 we provide details of the DL models that we use in our project. For each architecture we discuss the suitability and shortcomings related to the data objects that we use in our evaluations. In Section 4 we evaluate each proposed architecture using the INTERACTION dataset. We provide results that evaluate the accuracy of the model and prediction. Finally, in Section 5 we discuss our results and provide analysis of the strengths and shortcomings of our methods.

2 Literature Review

In recent years, the pursuit of autonomous vehicles have accelerated, with a wide variety of different machine learning techniques at the forefront of the evolution of the idea of self-driving vehicles. To better understand our own problem of incorporating an uncertainty quantification with the motion prediction of a self-driving vehicle, we need to explore the work that has already been completed when it comes to autonomous vehicles motion prediction, and how others have incorporated uncertainty quantification to their motion predictions.

2.1 Motion Prediction for Autonomous Vehicles

One of the main issues with Autonomous Vehicles is being able to understand the environment around them. While humans can recognize different situations and act accordingly, this task is different for a machine learning model that doesn't have the same situational awareness as a human. Motion prediction doesn't just pertain to the vehicle itself but to its surroundings as well, including pedestrians and other vehicles. .

State-of-the-art motion planning Techniques include Graph Search Based Planners, Sampling Based Planners, and Interpolating Curve Planners. According to [5] the primary Motion Planning Techniques used by researchers are the Interpolating Curve Planners and Graph Search Planners. Interpolating Curve Planners are used because of the accessibility of detailed Global Positioning System (GPS) Data that can be used to generate optimal curves. On the other hand in Graph Search Planners, primarily state lattices are used due to their speed that allow them to be used in real-time.

Another Motion Planning Technique is a Deep Reinforcement Learning Approach. While a single vehicle can learn through Reinforcement Learning, when it comes to increasingly complex situations involving multiple vehicles – such as following another vehicle lane keeping, merging, and driving in traffic – Multi-Agent Reinforcement Learning is utilized, but is significantly more complex than Reinforcement Learning. Authors in [1] discuss the Modeling of such conditions. For Reinforcement Learning this would mean Rewarding the agent when it does something correctly, and then modeling the Observation space, which pertains more to the vehicle state (e.g., position, speed), the environment around them (e.g., lanes, signs, rules), and other participants (e.g., vehicles/pedestrians) .

Authors in [22] discuss the Final Motion Planning Technique, a Multi layer Perceptron Approach. This algorithm predicts the likelihood of surrounding vehicles following specific lanes and trajectories, and achieves high accuracy and early detection of lane-change events. Utilizing real-world traffic data and simulation results, the proposed Multi-layer Perceptron model outperforms existing methods by detecting lane changes up to one to one and a half seconds earlier and predicting lane crossings three seconds in advance with about 90% accuracy. The algorithm's structure is adaptable to various prediction horizons without requiring numerous parameters and can assist with collision avoidance through risk assessment and autonomous control strategies on highways.

In the context of autonomous vehicle motion prediction involving pedestrians, leveraging modern machine learning-based models proves paramount. These models, unlike traditional physics-based approaches, adeptly handle complex environmental dependencies, enabling more accurate trajectory estimations. A comprehensive understanding of the surrounding environment, including pedestrian behaviors, road semantics, and interactions, is essential for building robust navigation systems. By employing multi-modal prediction techniques that consider various output types and situational awareness factors, autonomous vehicles can anticipate and respond effectively to diverse pedestrian movements, ensuring safer interactions on the road [7].

Authors in [11] provide a comprehensive review of various motion prediction algorithms. They identify three ways of making a motion prediction. These are based on Physics, Patterns, or Planning. For Physics Motion Prediction, models make use of motion equations to make predictions about the future motion of the desired object. This leads to some methods such as State Estimation, to estimate the mean and covariance of the desired object, and Reachable Sets, which are all the possible options that a vehicle could reach. Pattern Motion Prediction utilizes observed states to identify motion patterns for trajectory prediction. Clustering is the most simple, based on observed features and its closest match. Classification is very adaptable, categorizing observations into pre-defined groups and leveraging features to determine future actions. Finally, encoding-decoding utilizes deep neural networks to encode past trajectories and decode future predictions making it very flexible. In planning based prediction, there are two primary methods, Learning from Demonstration, and Planning from Uncertainty. Learning from Demonstration aims to infer motion predictions by estimating underlying cost functions or directly determining optimal policies. On the other hand, Planning from Uncertainty integrates prediction into ego-motion planning by treating predictions implicitly within the planning module, typically through methods like Partially Observable Markov Decision Processes (POMDPs) or game theory. Although very robust POMDP and game theory approaches are computationally intractable in continuous spaces [10].

Being able to be aware of social behaviors is one of the main reasons we utilize machine learning models over physics based trajectory prediction approaches nowadays. For example in one experiment authors in [8] classified drivers as one of three types: Aggressive, Normal, and Cautious, with aggressive being more concerned with travel efficiency and cautious more concerned with safety. However, in their experiment, they were able to account for this type of behavior and create different weighting coefficients based on safety, comfort and efficiency for each of the different driving styles.

2.2 Incorporating Uncertainty Quantification

Imagine your car being able to signal to you that another driver on the road may engage in a risky maneuver. For our project, we wish to utilize motion prediction capabilities and combine them with an uncertainty quantification (i.e., confidence score) to have a trustworthiness score for a motion prediction that can help a driver determine whether or not they should trust the signal from an their vehicle. In this subsection we will review the works that achieve uncertainty quantification.

Exploring the idea of uncertainty quantification due to the imperative nature of enhancing system reliability and safety in autonomous vehicle applications is discussed in [9]. The authors discuss the challenges posed by deterministic motion planning algorithms when it comes to handling uncertainties in a real-world scenario. Their proposed approach aims to quantify uncertainties arising from perception and environment modeling, and provide a more robust foundation for decision-making. They also discuss methodologies such as probabilistic modeling and Bayesian inference to capture and propagate uncertainties effectively.

Utilizing Deep Learning Networks to achieve motion prediction for vehicles is explores in [12]. This article discusses ways that can assist in incorporating uncertainty quantification for the motion prediction. They propose a framework to accomplish this, utilizing an uncertainty-aware 3D object detector – a chance-constrained safe motion planning approach – and evaluations based on both real-world and simulated driving scenarios. They also identify two sources of uncertainty in Deep Neural Network approaches, epistemic uncertainty, and aleatoric uncertainty. Epistemic Uncertainty is due to model parameters being mismatched between the training and inference input data, and aleatoric is more about randomness and incompleteness of the observations (i.e., noise in a sensor). The authors also identify three main approaches to uncertainty quantification: Monte Carlo dropout, deep ensembles, and direct modeling .

Authors in [18] implement a deep ensemble technique to estimate the uncertainty of motion prediction models and test them on two public datasets, Next Generation Simulation (NGSIM), and the INTERACTION Dataset. Their model is able to estimate both the epistemic and aleatoric uncertainty, and incorporate them into their motion prediction model. In order to accomplish this, they create a motion prediction model based on long short-term memory (LSTM) and created an uncertainty-aware potential field to help process the prediction uncertainty. Finally, they create a decision-making framework that takes into account the different uncertainty-aware potential fields, the road boundaries, and various vehicle dynamic constraints.

A novel approach to a physics constrained motion prediction model that incorporates uncertainty quantification is investigated in [19]. Authors utilize a two step integration consisting of an intent and trajectory prediction that's subject to dynamic constraints. By integrating physical constraints and probabilistic modeling techniques, the proposed framework offers more accurate and reliable predictions of object trajectories. The methodology involves leveraging physics-based models alongside data-driven approaches to capture uncertainties effectively. One of the tools they utilized is conformal prediction, which is a machine learning framework for uncertainty quantification that provides valid measures of confidence for individual predictions.

3 Methods

We implement various Deep Learning models and evaluate their performance in predicting the future states of a target vehicle. In this section we provide implementation details for each model and discuss an improved model based on Transformer to address the impact of environmental factors on computer systems. Our model has been extended and optimized on the basis of traditional Transformer to better capture the impact of environmental factors and improve resilience to complex environments.

3.1 MLP

Multi-Layer Perceptrons (MLP) consists of multiple layers of neurons. These neurons use linear activation functions (e.g., ReLU) that allow the network to learn complex patters in data [6]. MLPs are organized into layers with an input layer, followed by a hidden layer(s) and the output layer. Neurons in layers are connected by weighted connections. The weight in this context refers to the amount of influence one neuron's output has on another neuron's input. MLPs are very powerful and have shown success in learning non-linear relationships in data for tasks such as classification, regression, and pattern recognition. However, MLPs are not well suited for time series data such as our driving dataset. Additionally, MLPs do not have a temporal dimension therefore the can not understand the order of inputs. We use MLP with 100 epochs in our experiments and compare its performance against our other models. Our MLP model use a feed-forward network with three fully connected (dense) layers. The input data has a shape of 10×8 which is 10 frames or 1 seconds of historical data with 8 features and the input layer has 256 output features. The hidden layer takes the 256 output from the input layer and transforms them into 128 output features. The activation function used is Rectified Linear Unity (ReLU). The output layer takes the 128 output features from the previous layer and transforms them into 30×8 which is 30 frames or 3 seconds of predicted future states. The features that we chose for implementing the MLP are: case-id, track-id, frame-id, timestamp-ms, agent-type, x, y, v_x, v_y, psi-rad, length, and width. Case-id is the ID of the case under this driving scenario, track-id is the agent's ID which we use for choosing the target vehicle, frame-id is the ID of the current frame, timestamp-ms is the time instant of the corresponding frame in ms, agent-type is the type of the agent, x is the x position of the agent with unit of meters, y is the x position of the agent with unit of meters, v_x is the velocity in the x-direction of the agent with units m/s, v_y is the velocity in the y-direction of the agent with units m/s, and psi-rad is the yaw angle of the agent if the agent is a vehicle.

3.2 LSTM

Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) that is suitable for handing sequential data with long-term dependencies [6] [23]. LSTM networks consist of a series of LSTM cells in which there are a set of gates (input, output, and forget gates) that control the flow of information into and out of the specific cell. Note that the cell remembers the values over arbitrary time interval sand the gates regulate the flow of information into and out of the cell. Given that LSTMs are designed to handle sequential data they are particularly suitable for our time series data. This enables the network to remember past information (in our case historical vehicle state) and make future predictions. Another key advantage of LSTMs is their ability to capture long-term dependencies in a data sequence. In times series data such as ours this is beneficial because a drivers pattern of driving may be repeated in several time steps. We implement our LSTM model with an LSTM layer that takes the input data. The input data has size of 8. Th LSTM layer also has a hidden size of 256. The fully connected layer takes the output from the output of the LSTM layer and transforms it into the output size of 8 which is the number of features. The features in our LSTM implementation are the same as in MLP and we train the model for 5000 epochs.

3.3 GRU

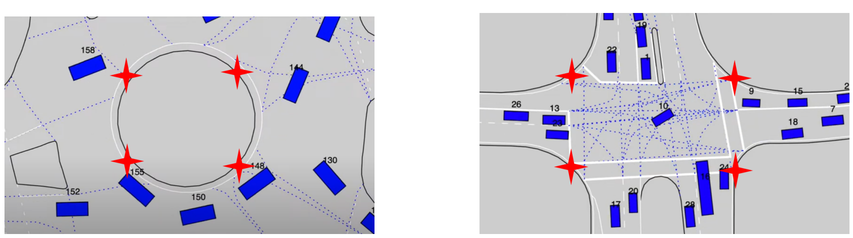
Gated Recurrent Unit (GRU) is a type of model which stems from recurrent neural networks (RNN) and are designed to handle sequence prediction problems more effictevely than RNNs. GRUs also address the vanishing gradient problem, enabling them to maintain performance even when learning from data with long-duration dependencies. GRUs streamline the complex structures of traditional RNNs by merging the forget and input gates into a singular "update gate," which facilitates faster training without significantly compromising the model's effectiveness[4]. The features we chose for implementing our GRU model are 1 seconds (10 datapoints) of x, y, v_x, v_y, psi-rad, length, and width values. Our GRU model consists of several GRU layers stacked together, which enhances the model's ability to learn trajectory projection from the input sequences. Each of the GRU layers in our model uses the ReLU activation function to help with this task. After the GRU layers, two dense layers follow. Both dense layers output 60 values, corresponding to 30 x coordinates and 30 y coordinates, which is equivalent to predicting 3 seconds of trajectory.

3.4 Road Information Based Transformer

Transformer model design is the core of the dependence on the attention mechanism, the mechanism by different positions of the input data related to the sequence of said [20]. The encoder consists of a series of identical layers, each containing two sub-layers. The first sub-layer is the self-concerned mechanism, and the second sub-layer is the position fully connected feedforward network. Self-focus in the encoder allows the model to weigh the importance of different words in the input data regardless of their position in the sequence, capturing context more efficiently. The decoder is able to focus on the relevant parts of the input sequence, thereby enhancing its predictive power. The decoder layer also integrates a self-focused and feedforward network to facilitate the generation of an output sequence of elements one at a time. Each element of the output sequence is generated by considering the elements already produced, ensuring that the output is coherent and context-relevant.

In this section, we will delve into the integration of environment variables into the Transformer model to enhance trajectory prediction in autonomous driving scenarios. The combination of these variables is critical because it allows the model to more effectively predict and respond to dynamic road conditions and traffic interactions.

As mentioned earlier, the modified Transformer architecture is used to process location data sequences from the vehicle, which are represented by the blue rectangles in Fig. 2 These represent the real-time locations of various cars in traffic scenarios, such as roundabouts. The environmental background is enriched by custom road markings, which are represented by a red star on the chart. These markers are strategically placed at points where the trajectory could change, such as near the exit of a roundabout or at an intersection. The rationale behind placing these markers stems from the observation that vehicles are more likely to change their trajectory at these key points. This mechanism allows the model to weigh the importance of different locations in the input data based on the environmental context provided by the road markings. Each marker is a signal that changes the focus of the attention mechanism, leading it to consider the trajectory change more likely at these marker points.



(a) Roundabout road type.

(b) Intersection road type.

Figure 2: Difference road information.

4 Experiments

4.1 Evaluation Metrics

In this section, we evaluate the performance of our vehicle motion prediction models using three key metrics: Minimum Average Displacement Error (minADE), Minimum Final Displacement Error (minFDE), and Miss Rate (MR). These metrics help quantify the accuracy and reliability of our predictions under various environmental conditions and operational scenarios.

4.1.1 Minimum Average Displacement Error (minADE)

The minADE metric measures the minimum Euclidean distance, averaged over time, between each predicted modality and the ground truth trajectory. This metric provides a robust assessment of the model's performance over the entire prediction horizon. Given that the number of predicted timestamps T is 30 and the number of modalities K varies, the minADE for a single case is computed as follows:

$$\text{minADE}_K = \min_{k \in 1,\dots,K} \frac{1}{T} \sum_{t=1}^{T} \sqrt{(\hat{x}_t - x_t^k)^2 + (\hat{y}_t - y_t^k)^2}$$

where $\hat{x}$ and $\hat{y}$ is the ground truth position at the same timestamp.

4.1.2 Minimum Final Displacement Error (minFDE)

The minFDE metric focuses on the accuracy of the model at the critical final predicted timestamp, which is crucial for immediate navigational decisions. This is calculated as:

$$\text{minFDE} = \min_{k \in 1,\dots,K} \sqrt{(\hat{x}_T - x_T^k)^2 + (\hat{y}_T - y_T^k)^2}$$

4.1.3 Miss Rate (MR)

MR quantifies the percentage of cases where the vehicle's predicted position deviates beyond acceptable lateral or longitudinal thresholds at the final timestamp. These thresholds are adapted to the vehicle's speed with the following piecewise function for the longitudinal threshold:

$$\text{Threshold}_{\text{lon}} = \begin{cases} 1 & v < 1.4m/s \\ 1 + \frac{v-1.4}{11-1.4} & 1.4m/s \leq v \leq 11m/s \\ 2 & v \geq 11m/s \end{cases}$$

4.2 RMSE (Root Mean Squared Error)

RMSE measures the square root of the average squared differences between predicted values ($\hat{y}_i$) and actual values (y_i)

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2}$$

4.3 MAE (Mean Absolute Error)

MAE calculates the average of the absolute difference between predicted values ($\hat{y}_i$) and actual values (y_i)

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i|$$

4.4 MRE (Mean Relative Error)

MRE computes the average of the absolute differences between predicted values ($\hat{y}_i$) and actual values (y_i), normalized by the magnitude of the actual values.

$$\text{MRE} = \frac{1}{n} \sum_{i=1}^{n} \frac{|y_i - \hat{y}_i|}{|y_i|}$$

4.5 Coefficient of Determination (R2 Score)

R2 score is a measure of how well the model fits the data. It measures the proportion of variance in the dependent variable that is predictable from the independent variables. It ranges from 0 to 1, with 1 being a perfect prediction.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

4.6 Explained Variance Score (EVS)

EVS quantifies the amount of variance in a target variable (y) that is explained by the model predictions ($\hat{y}$).

$$\text{EVS} = 1 - \frac{Var(y - \hat{y})}{Var(y)}$$

4.7 Model Performance

4.7.1 MLP

We present are MLP results in Figure 3. Although MLP shows correct progression of loss for training and validation datasets we note that as expected the predicted trajectory is not smooth and does not reflect a realistic motion of a vehicle. Given the lack of temporal dimension that we discussed in Subsection 3.1 our results correctly show the weaknesses of MLP architecture. However, we note that MLP shows the correct trend of motion that is close to the ground truth data.

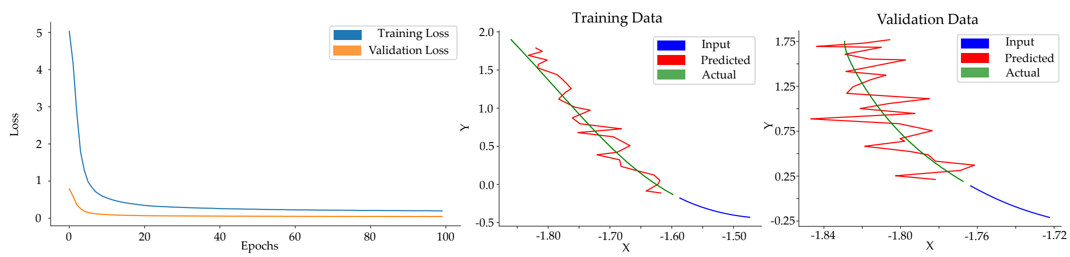


Figure 3: MLP Loss (MSE) vs Epochs & Trajectory Prediction. Left: plots of loss for training and validation vs epochs. Middle: Comparison of predicted trajectory (red) vs actual/ground truth trajectory (green) given the historical input (blue) for training dataset. Right: Comparison of predicted trajectory (red) vs actual/ground truth trajectory (green) given the historical input (blue) for validation dataset.

We present the accuracy metrics of implementation and confidence score for MLP in Table 2. We used Root Mean Squared Error (RSME), Mean Absolute Error (MAE), Mean Relative Error (MRE), Coefficient of Determination (R2 score), Explained Variance Score (EVS) to evaluate the accuracy of the model and Minimum Average Displacement Error (minADE), Minimum Final Displacement Error (minFDE), and Miss Rate (MR) to evaluate the confidence on the predicated trajectory. We note that the results for MLP in Table 2 are for one highway environment in one country and have not been averaged across all environments.

4.7.2 LSTM

We present are LSTM results in Figure 4. Although LSTM predicts smooth trajectory that is close to the ground truth it learns over longer epochs. The higher time complexity of using LSTM is a drawback that limits its application in real-life settings.

We present the accuracy metrics of implementation and confidence score for LSTM in Table 3. We used the same metric as discussed before and evaluate the results for one highway environment in one country. The results have not been averaged across all environments.

Metric	Training Score	Validation Score
RMSE	0.126	0.150
MAE	0.087	0.102
MRE	18307.998	19467.482
R2	0.841	0.828
EVS	0.841	0.829
minADE	0.166	0.176
minFDE	0.552	0.674
MR	0.068	0.087

Table 1: Accuracy and confidence score results for MLP architecture. The results are for evaluation on one highway environment.

minADE	minFDE	MR	Dataset
0.1872	0.1060	0.4031	DEU Merging
0.1536	0.2949	0.4554	CHN Merging
0.3726	0.7506	0.6188	DEU Roundabout
0.2875	0.5310	0.5603	CHN Roundabout
0.3479	0.4414	0.5340	USA Roundabout
0.2138	0.4036	0.3351	USA Intersection

Table 2: Accuracy and confidence score results for MLP architecture. The results are for evaluation on different environments.

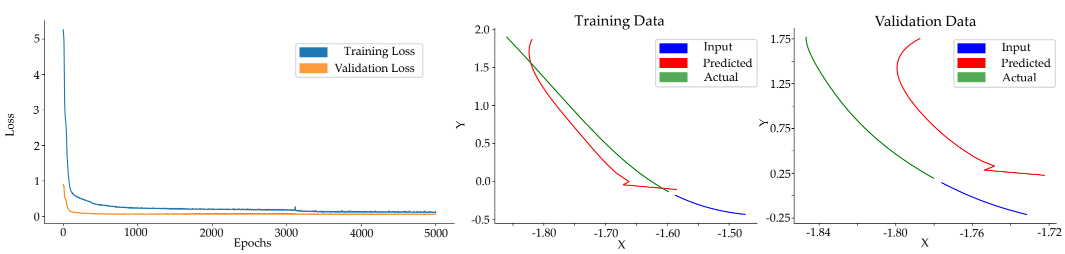


Figure 4: LSTM Loss (MSE) vs Epochs & Trajectory Prediction. Left: plots of loss for training and validation vs epochs. Middle: Comparison of predicted trajectory (red) vs actual/ground truth trajectory (green) given the historical input (blue) for training dataset. Right: Comparison of predicted trajectory (red) vs actual/ground truth trajectory (green) given the historical input (blue) for validation dataset.

Metric	Training Score	Validation Score
RMSE	0.088	0.201
MAE	0.056	0.124
MRE	26941.446	26197.692
R2	0.863	0.807
EVS	0.864	0.808
minADE	0.184	0.366
minFDE	0.328	0.756
MR	0.0315	0.124

Table 3: Accuracy and confidence score results for LSTM architecture. The results are for evaluation on one highway environment.

4.7.3 Transformer

Experimental results show that the predictive accuracy of the Transformer model is significantly improved when using larger data sets and introducing environmental variables. As shown in the Table 4, when using only a single dataset, the minADE and minFDE values of the model are high, and the MR Values are relatively high. When using a dataset of both U.S. intersections and roundabout roads, all metrics improved, with minADE dropping from 0.6712 to 0.5035, minFDE from 0.7194 to 0.6195, and MR From 0.2291 to 0.2062. The most significant improvement occurred when environment variables were introduced at the same time, when minADE dropped to 0.4207, minFDE dropped to 0.4629, and MR Dropped significantly to 0.1534. This result shows that the expansion of data set and the addition of environment variables are crucial to improve the prediction ability of the model in complex traffic scenarios.

minADE	minFDE	MR	Dataset	Environment factor
0.6712	0.7194	0.2291	USA Intersection	Not including
0.6476	0.6834	0.2146	USA Roundabout	Not including
0.5035	0.6195	0.2062	USA Intersection, USA Roundabout	Not including
0.4207	0.4629	0.1534	USA Intersection, USA Roundabout	Including

Table 4: Accuracy and confidence score results for Transformer model. The results are for evaluation on environments factor.

(a) (b) (c)

Figure 5: Trajectory Prediction with single dataset without environment factor (left), multiple dataset without environment factor (middle), and multiple datasets with environment factor (right).

From the trajectory prediction graph provided, it is obvious that with the increase of data sets and the introduction of environmental variables, the prediction effect of the model is significantly improved. In the training of a single data set (Fig. 5a), there is a large deviation between the predicted trajectory and the real trajectory, and the predicted path is chaotic. When the training data set increases (Fig. 5b), the accuracy of the predicted trajectory is improved, and the predicted path is more consistent with the real path, indicating that the model can better learn and adapt to different driving scenarios. Finally, in the case of multiple data sets and the introduction of environmental variables (Fig. 5c), the predicted trajectories almost coincide with the real trajectories, showing high accuracy and sensitivity to environmental factors. This shows that the addition of environment variables is very effective for improving the adaptability and prediction ability of the model in complex environments.

4.7.4 GRU

We present are GRU results in Figure 3. Like some of the other models, GRU does not offer smooth enough trajectory prediction for both training and validation data, although the validation data does appear to be smoother than some of the other models.

We present the accuracy metrics of implementation and confidence score for GRU in Table 5. We use the same measurements as described in the previous sections.

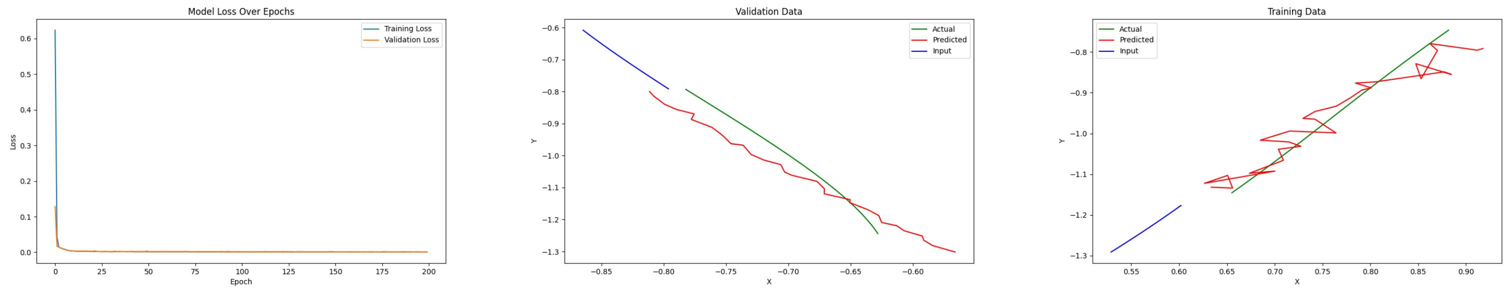


Figure 6: Loss vs Epochs & Trajectory Prediction.Left: plots of loss for training and validation vs epochs. Middle: Comparison of predicted trajectory (red) vs actual/ground truth trajectory (green) given the historical input (blue) for validation dataset. Right: Comparison of predicted trajectory (red) vs actual/ground truth trajectory (green) given the historical input (blue) for training dataset

Metric	Training Score	Validation Score
RMSE	0.0207	0.0238
MAE	0.0159	0.018
MRE	0.0937	0.0763
R2	0.976	0.828
EVS	0.977	0.973
minADE	0.1557	0.179
minFDE	0.1445	0.139
MR	0.0503	0.0794

Table 5: Accuracy and confidence score results for GRU architecture. The results are for evaluation on one highway environment.

minADE	minFDE	MR	Dataset
0.4367	0.1719	0.0877	DEU Merging
0.1905	0.1621	0.1178	CHN Merging
6.418	5.163	0.9999	DEU Roundabout
0.2604	0.2125	0.1784	CHN Roundabout
6.737	10.660	0.9967	USA Roundabout
7.144	4.6701	0.9953	USA Intersection

Table 6: Accuracy and confidence score results for GRU architecture. The results are for evaluation on different environments.

5 Discussion & Future Work

Our results across the four different deep learning architecture show that Transformer and LSTMs are more successful in predicting the future state of target vehicle given the historical trajectory of multiple vehicles as the input. Although, our accuracy metrics and confidence scores demonstrate the effectiveness of our models we note that there are potential sources of error that need to be addressed in future implementations. In this section we describe each potential source and provide a way to address them.

5.1 Missing data

Given that some cars were not always visible in the dataset we made choices about handling the missing data. The two strategies that we implemented are: using 0 for the empty rows and using the last known state. But these introduce bias to the model. To address this we propose using a consistent set of strategies to handle missing data tested across multiple environments. With this, we will be able to assess which strategy is more successful than the others. Also, the strategy can be adaptive and based on specific cars and their previous trajectories.

5.2 Realism of the predicted trajectory

Although the models accurately predict the future states, they do not consider dynamics of the vehicle. That is, some of the predicted states may not be possible given the physics of the environment and the cars. We propose to implement the model as a high-level planner that provides the general trend of the motion of a vehicle. Then, implement a low-level planner that considers the car dynamics to plan a feasible trajectory.

5.3 Assumptions

Our models assume that drivers are rational agents, and their previous actions are an indication of their future states. This is a strong assumption and may not hold in real-life scenarios. To overcome this limitation, we propose using the historical data as a prior for a human model (such as Boltzmann nosily rational model explored in cognitive science [2]) and tune the hyperparameters of the human model to be able to better predict drivers' behavior.

Conclusion

In conclusion, our study presents an analysis of vehicle motion prediction by integrating multi-layer perceptron (MLP), long short-term memory (LSTM), gated reccurent units (GRU) and transformer models. Our models' performance on the INTERACTION dataset present the ability to predict vehicle trajectories accurately.

We observed the LSTM and Transformer models in particular showed superior capability in handling the nature of vehicle motion data and the complex interactions with the traffic environments. The LSTM model was great at predicting a smooth trajectory, which is necessary for a safe and comfortable driving experience Moreover, the the inclusion of the Transformer model highlighted the importance of incorporating external environmental factors that influence vehicle behavior.

Despite our success, our study also acknowledges the limitations associated with data availability and our assumptions of driver behavior. There is still work to be done addressing these challenges by integrating more advance models that better capture the realities of driving patterns.

References

[1] Szilárd Aradi. Survey of deep reinforcement learning for motion planning of autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 23(2):740–759, 2022.

[2] Chris L Baker, Joshua B Tenenbaum, and Rebecca R Saxe. Goal inference as inverse planning. In *Proceedings of the annual meeting of the cognitive science society*, volume 29, 2007.

[3] Chiho Choi, Joon Hee Choi, Jiachen Li, and Srikanth Malla. Shared cross-modal trajectory prediction for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 244–253, 2021.

[4] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling, 2014.

[5] David González, Joshué Pérez, Vicente Milanés, and Fawzi Nashashibi. A review of motion planning techniques for automated vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 17(4):1135–1145, 2016.

[6] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[7] Mahir Gulzar, Yar Muhammad, and Naveed Muhammad. A survey on motion prediction of pedestrians and vehicles for autonomous driving. *IEEE Access*, 9:137957–137969, 2021.

[8] Peng Hang, Chen Lv, Chao Huang, Jiacheng Cai, Zhongxu Hu, and Yang Xing. An integrated framework of decision making and motion planning for autonomous vehicles considering social behaviors. *IEEE Transactions on Vehicular Technology*, 69(12):14458–14469, 2020.

[9] Michael Hartmann, Daniel Watzenig, Michael Stolz, Edoardo Patelli, and Marco De Angelis. Uncertainty quantification for motion planning. In *Proceedings of the NAFEMS World Congress 2017*, 2018.

[10] Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1-2):99–134, 1998.

[11] Phillip Karle, Maximilian Geisslinger, Johannes Betz, and Markus Lienkamp. Scenario understanding and motion prediction for autonomous vehicles—review and comparison. *IEEE Transactions on Intelligent Transportation Systems*, 23(10):16962–16982, 2022.

[12] Dachuan Li, Bowen Liu, Zijian Huang, Qi Hao, Dezong Zhao, and Bin Tian. Safe motion planning for autonomous vehicles by quantifying uncertainties of deep learning-enabled environment perception. *IEEE Transactions on Intelligent Vehicles*, 9(1):2318–2332, 2024.

[13] Jiale Li, Hang Dai, Hao Han, and Yong Ding. Mseg3d: Multi-modal 3d semantic segmentation for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21694–21704, 2023.

[14] Jintang Li, Zhouxin Yu, Zulun Zhu, Liang Chen, Qi Yu, Zibin Zheng, Sheng Tian, Ruofan Wu, and Changhua Meng. Scaling up dynamic graph representation learning via spiking neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 8588–8596, 2023.

[15] Ruibo Li, Hanyu Shi, Ziang Fu, Zhe Wang, and Guosheng Lin. Weakly supervised class-agnostic motion prediction for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17599–17608, 2023.

[16] Xiaoyu Mo and Chen Lv. Predictive neural motion planner for autonomous driving using graph networks. *IEEE Transactions on Intelligent Vehicles*, 8(2):1983–1993, 2023.

[17] Ransalu Senanayake, Lionel Ott, Simon O'Callaghan, and Fabio T Ramos. Spatio-temporal hilbert maps for continuous occupancy representation in dynamic environments. *Advances in Neural Information Processing Systems*, 29, 2016.

[18] Xiaolin Tang, Kai Yang, Hong Wang, Jiahang Wu, Yechen Qin, Wenhao Yu, and Dongpu Cao. Prediction-uncertainty-aware decision-making for autonomous vehicles. *IEEE Transactions on Intelligent Vehicles*, 7(4):849–862, 2022.

[19] Renukanandan Tumu, Lars Lindemann, Truong Nghiem, and Rahul Mangharam. Physics constrained motion prediction with uncertainty quantification. In *2023 IEEE Intelligent Vehicles Symposium (IV)*, pages 1–8, 2023.

[20] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[21] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1):4–24, 2021.

[22] Seungje Yoon and Dongsuk Kum. The multilayer perceptron approach to lateral motion prediction of surrounding vehicles for autonomous vehicles. In *2016 IEEE Intelligent Vehicles Symposium (IV)*, pages 1307–1312, 2016.

[23] Yong Yu, Xiaosheng Si, Changhua Hu, and Jianxun Zhang. A review of recurrent neural networks: Lstm cells and network architectures. *Neural computation*, 31(7):1235–1270, 2019.

[24] Ekim Yurtsever, Jacob Lambert, Alexander Carballo, and Kazuya Takeda. A survey of autonomous driving: Common practices and emerging technologies. *IEEE Access*, 8:58443–58469, 2020.

[25] Wei Zhan, Liting Sun, Di Wang, Haojie Shi, Aubrey Clausse, Maximilian Naumann, Julius Kümmerle, Hendrik Königshof, Christoph Stiller, Arnaud de La Fortelle, and Masayoshi Tomizuka. INTERACTION Dataset: An INTERnational, Adversarial and Cooperative moTION Dataset in Interactive Driving Scenarios with Semantic Maps. *arXiv:1910.03088 [cs, eess]*, September 2019.