Trajectory Prediction Neural Network and Model Interpretation Based on Temporal Pattern Attention

Temporal Pattern Attention 기반한 궤적 예측 신경망 및 모델 해석

Dept. of AI and Bigdata, SOON CHUN HYANG Univ.

민현식
minun001@sch.ac.kr

CONTENTS

1. Introduction

1. Introduction

1.Perception Ability of Autonomous Vehicles: Vehicles utilize sensors to perceive a broader range of information than humans.

2.Importance of Trajectory Prediction: This enables precise predictions of the intentions and future trajectories of surrounding vehicles.

3.Challenges: The diversity of driving behaviors and the complexity of contexts make trajectory prediction a difficult task.

→ **TP2Net**

TP2Net (Temporal Pattern Attention-based Trajectory Prediction Network)

- Proposes a new model based on temporal pattern attention to extract hidden driving features.

VOI Inception

- Incorporates a method based on GoogLeNet to focus on interactions with surrounding vehicles.

Interpretability Issue

- Highlights the 'black box' problem of deep learning models and suggests an interpretation module to enhance the model's reliability.

2. Related Research

2. Related Research

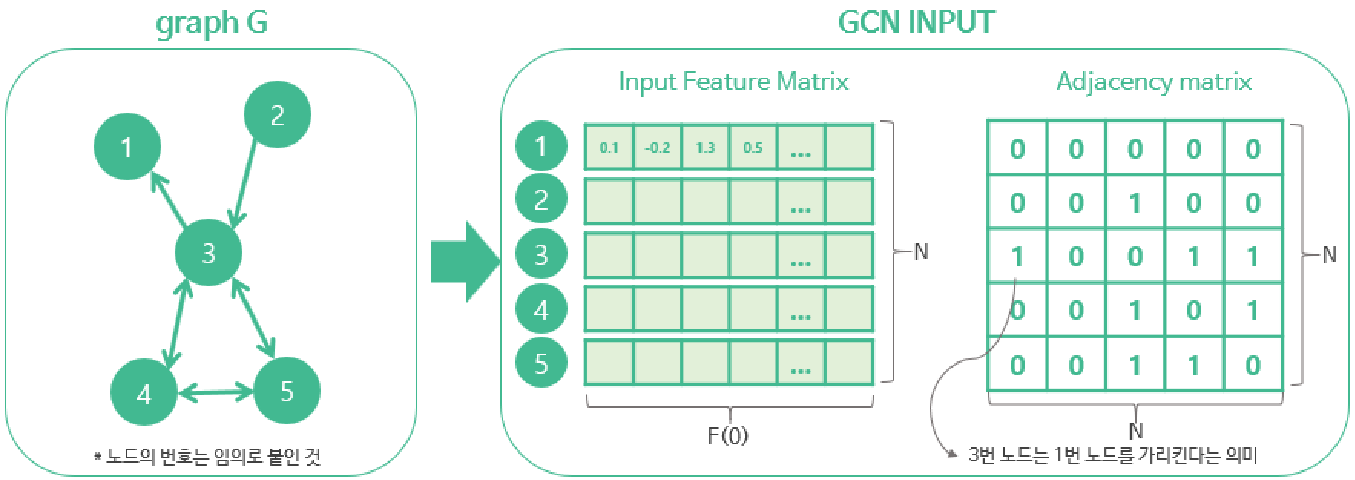
A. Trajectory Prediction Methods

1. Conventional Machine Learning Methods

- Early vehicle trajectory predictions often utilized kinematic and dynamic parameters.
- Models like the constant yaw rate and acceleration model, and Bayesian filters such as unscented and extended Kalman filters were used.
- Models incorporating driving maneuver recognition were proposed to enhance long-term prediction accuracy.

2. Deep Learning Methods

- Recent studies have leveraged network structures like RNNs and LSTMs for extracting hidden dependencies.
- Various approaches including LSTM encoder/decoder frameworks, convolutional social pooling, and conditional variational auto-encoders have been employed.
- Techniques like Multi-Agent Tensor Fusion (MATF) and Graph Convolution Networks (GCN) have been used for modeling interactions with surrounding vehicles.



2. Related Research

B. Interpretation of Neural Networks

Instance Level Interpretation

- Suitable for explaining the activation features of specific neurons causing a particular prediction.

Gradient-based Methods

- Calculate partial derivatives of each class relative to the input, evaluating global feature importance.

Perturbation Methods

- Introduce noise to the original input and observe changes in the hidden layer to measure feature importance.

3. Development of the Proposed Model

3. Development of the Proposed Model

A. Problem Formulation

Input

$$\boldsymbol{a}_T \quad t_{hst} \quad (t_{hst} = -w_h, \ldots, -2, -1, 0)$$

historical trajectory data

The coordinate system's origin is set at the current position of the target vehicle,

with the x-axis aligned with the vehicle's longitudinal direction (parallel to the lane) and

the y-axis with its lateral direction (perpendicular to the lane)

VOI Inception

focusing on key vehicles around the target vehicle such as

the preceding, left preceding, right preceding, left alongside, right alongside, left following, right following, and the following vehicle.

Additional Parameters

For each vehicle's historical trajectory

$$\boldsymbol{a} = [\boldsymbol{a}^{-w_h}, \boldsymbol{a}^{-w_h+1}, \ldots, \boldsymbol{a}^{-1}, \boldsymbol{a}^{0}]$$

$$\boldsymbol{a}^{t_{hst}} = (x^{t_{hst}}, y^{t_{hst}}, v_x^{t_{hst}}, v_y^{t_{hst}}, a_x^{t_{hst}}, a_y^{t_{hst}}, class)$$

3. Development of the Proposed Model

A. Problem Formulation

Driving Maneuver Recognition

The current driving maneuver classes M_{lon} and M_{lat} are defined to represent longitudinal (normal driving (ND), hard braking (HB), and rapid acceleration (RA)) and lateral (lane following (LF), left lane change (LLC), and right lane change (RLC)) maneuvers, respectively.

The objective of this study

$$\text{the target vehicle } T \xrightarrow{\text{predict}} t_{fut}\ (t_{fut} = 1, 2, \ldots, w_f)$$

$$\text{the predicted trajectory} \xrightarrow{\text{defined}} a^{t_{fut}} = (x^{t_{fut}}, y^{t_{fut}})$$

3. Development of the Proposed Model

B. Proposed Model

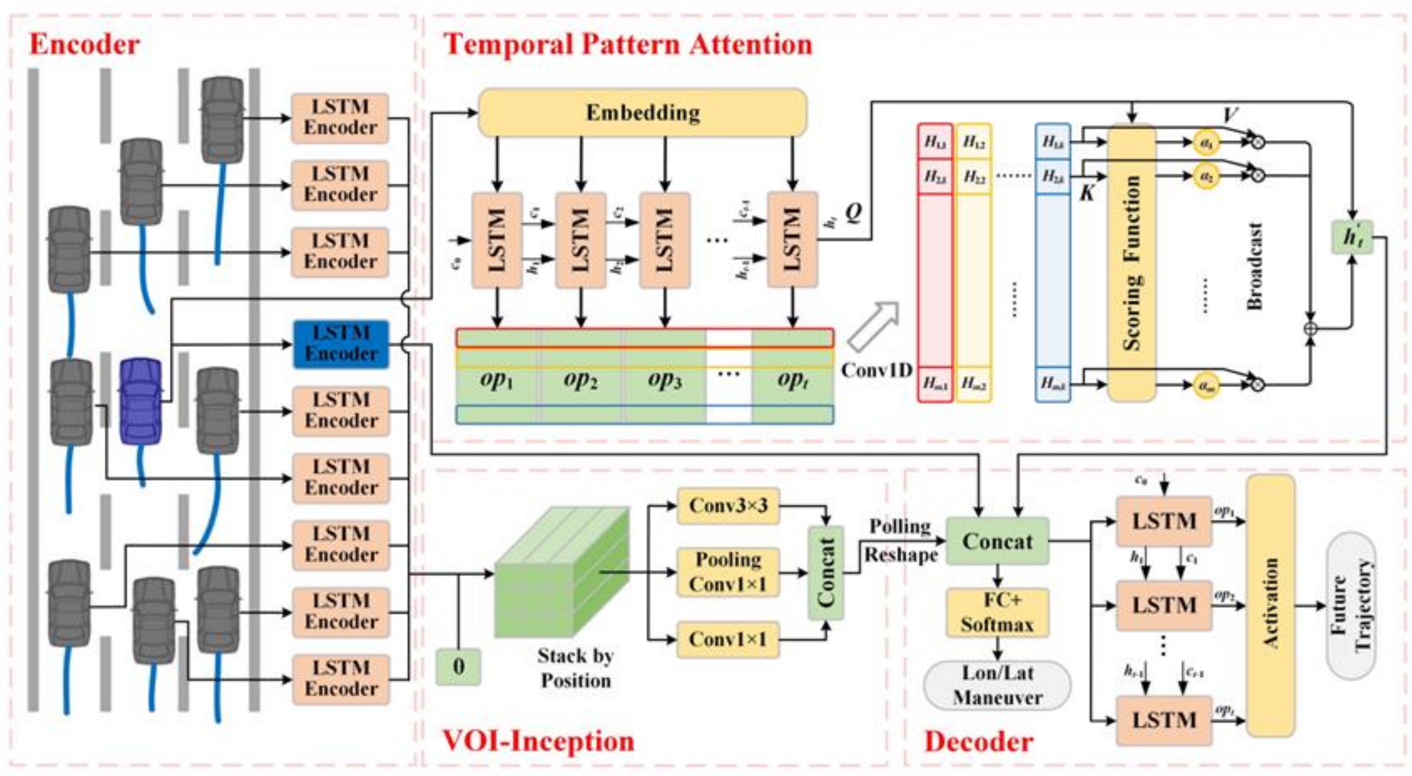


Fig. 1. Schematic diagram of the proposed TP2Net structure.

the proposed model is composed of four parts

Encoder, Decoder, TPA, and VOI inception

input coding

output decoding

target vehicle information

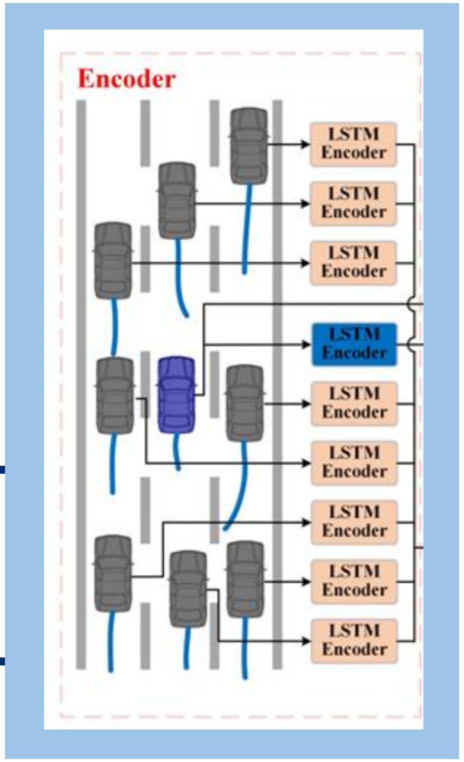
surrounding vehicle information

Senseable AI Lab

3. Development of the Proposed Model

B. Proposed Model

1) Encoder



1. Purpose Definition

The encoder is primarily used to model the vehicle trajectory, including the vehicle's past trajectory and driving maneuver patterns.

2. Processing Target and Surrounding Vehicles

Both target vehicle and surrounding vehicles (VOI) data are encoded. The VOI encoder shares parameters, whereas the target vehicle encoder uses independent parameters.

Embedding Vector Creation

Each vehicle data $\boldsymbol{a}_c \in \mathcal{A}$ and $\boldsymbol{a}_T$ is transformed into an embedding vector through a Fully Connected (FC) layer.

LSTM Processing

The embedding vector e_c is fed into the LSTM network, and the hidden state vector h_t produced by the last unit of the LSTM represents hidden driving features.

$$\boldsymbol{e}_c = \phi\left(\boldsymbol{a}_c; \boldsymbol{W}_{emb}\right) \quad (1)$$

$$\boldsymbol{h}_c^t = \phi\left(\mathrm{LSTM}\left(\boldsymbol{e}_c, \boldsymbol{h}_c^{t-1}; \boldsymbol{W}_{enc}\right); \boldsymbol{W}_{lin}\right) \quad (2)$$

Encoding Tensor Production

1. Linear Transformation

Converts the hidden state vector for the next layer's input.

2. LeakyReLU Activation

Applies non-linear transformation to address the dead neurons issue.

3. Final Encoding Tensor

The activated vector becomes the encoding tensor for further model processing.

3. Development of the Proposed Model

B. Proposed Model

2) TPA

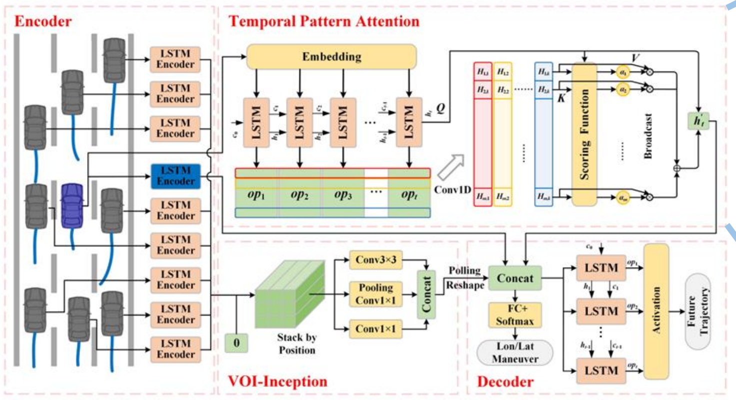
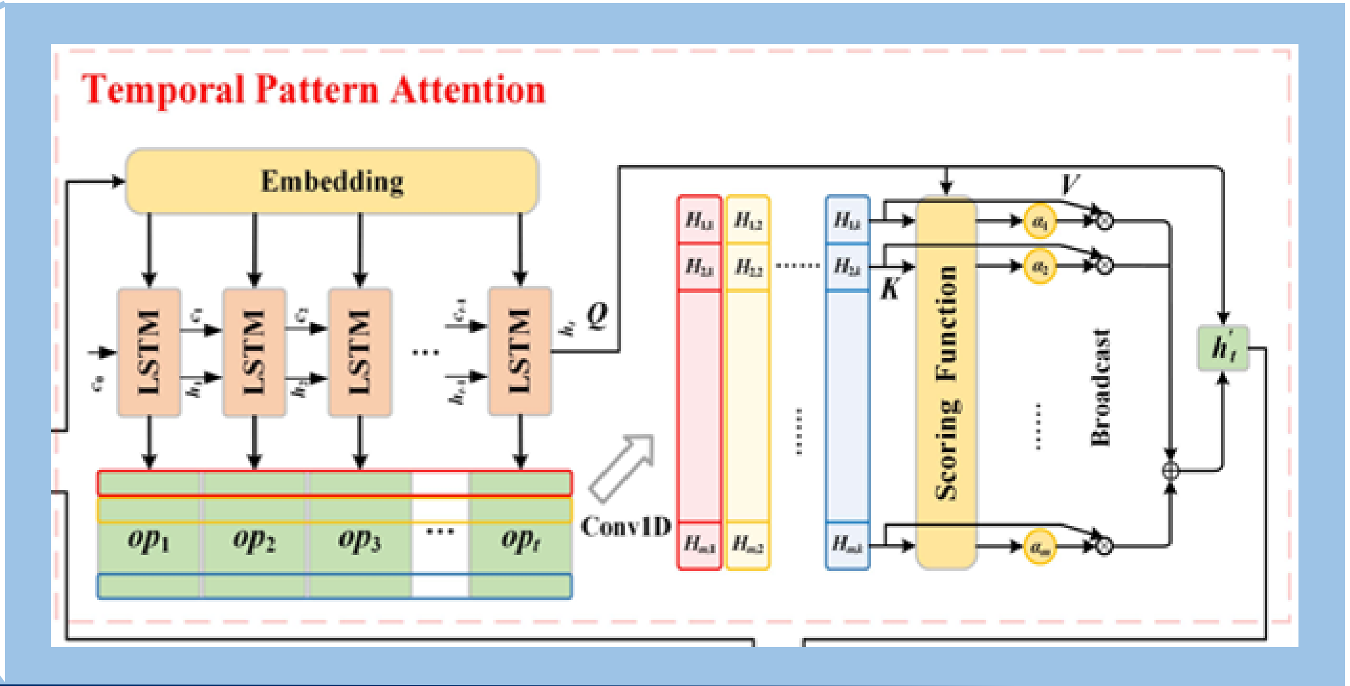


Fig. 1. Schematic diagram of the proposed TP2Net structure.



1. Traditional Attention in Time Series Forecasting

- Focuses on selecting relevant time steps.
- Suitable for tasks where each time step contains crucial pieces of information.

2. Issue with MTS Prediction

- Traditional attention might introduce noise in multivariate time series (MTS) prediction.

3. Introduction of TPA Mechanism

- Temporal Pattern Attention (TPA) uses a weighted sum of line vectors across multiple time steps.
- Captures broader temporal information, beyond individual time steps.

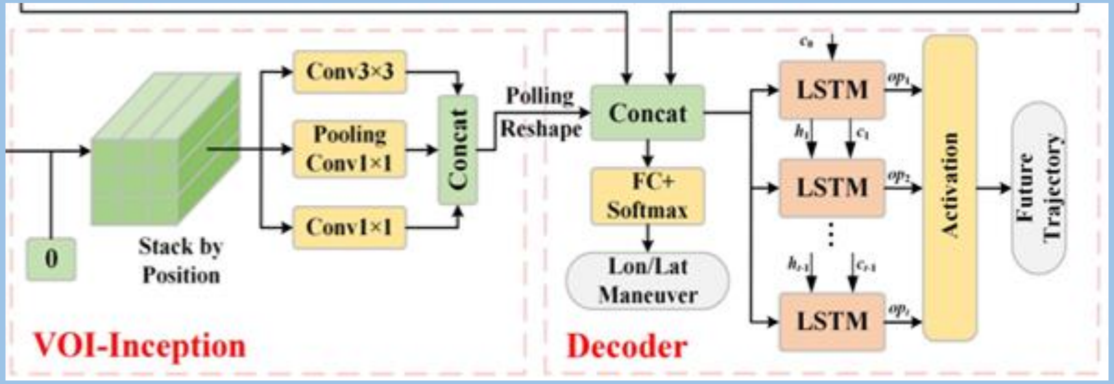
4. Advantages of TPA

- Enhances the ability to extract complex time patterns.
- Improves prediction accuracy, particularly in tasks with intricate temporal dynamics like trajectory prediction.

3. Development of the Proposed Model

B. Proposed Model

3) VOI Inception



Encoding Challenge

When encoding the motion of surrounding vehicles, it is difficult to accurately capture the spatial and positional features of the driving context.

VOI Inception Proposal

The proposal suggests stacking the encoded tensors of the target vehicle and surrounding vehicles according to their spatial positions to maintain the spatial information of all vehicles.

Inception Structure

Due to the size limitation of Estk, three branches similar to the inception v1 module are adopted, using various sizes of convolution kernels and max pooling to extract features.

Decoder Structure

The decoder concatenates the encoded tensor of the target vehicle, TPA tensor, and VOI inception tensor, predicts the distribution of driving maneuvers, and forecasts the future position values of the vehicle.

Loss Function

$$J_{MSE} = \frac{1}{N}\sum_{n=1}^{N}\sum_{j=1}^{t_{fut}}\left\|\hat{a}_{n,j} - a_{n,j}\right\|^2$$

$$J_{CE}^{lon} = -\frac{1}{N}\sum_{n=1}^{N}\sum_{j=1}^{class} P_{lon}\log(\hat{P}_{lon})$$

$$J = J_{CE}^{lon} + J_{CE}^{lat} + J_{MSE}$$

3. Development of the Proposed Model

C. Interpretation Module for TP2Net

During the training phase, noise is added to capture the changing patterns before the next training iteration.

Algorithm 1 Interpretation Module for TP2Net

Initialization: $\Psi(\cdot;\, \boldsymbol{W}_\Psi)$, $\boldsymbol{X}_{set}$, λ, $epochs$, $\boldsymbol{\sigma}_I \leftarrow \emptyset$
$\sigma_S^2 \leftarrow \mathrm{var}(\Psi(\boldsymbol{X}_{set};\, \boldsymbol{W}_\Psi))$
for $idx \leftarrow 0$ **to** $|\boldsymbol{X}_{set}|$
    Initialization: Interpreter$(\cdot;\boldsymbol{W}_\sigma)$
    for $ep \leftarrow 0$ **to** $epochs$
        $\boldsymbol{\sigma} \leftarrow Sigmoid(\boldsymbol{W}_\sigma)$
        $\boldsymbol{s} \leftarrow \Psi(\boldsymbol{X}_{idx};\, \boldsymbol{W}_\Psi)$
        $\tilde{\boldsymbol{X}}_{idx} \leftarrow \boldsymbol{X}_{idx} + \boldsymbol{\delta}_{\delta_i \sim \mathcal{N}(0, \boldsymbol{\Sigma}_i = \sigma_i^2 \boldsymbol{I})}$
        $\tilde{\boldsymbol{s}} \leftarrow \Psi(\tilde{\boldsymbol{X}}_{idx};\, \boldsymbol{W}_\Psi)$
        Update *loss* based on Equation (29)
        Train the Interpreter$(\cdot;\boldsymbol{W}_\sigma)$ (*loss* backpropagation)
    end for
    $\boldsymbol{\sigma} \leftarrow Sigmoid(\boldsymbol{W}_\sigma)$
    $\boldsymbol{\sigma}_I \leftarrow \boldsymbol{\sigma}_I \cup \boldsymbol{\sigma}$
end for
Output: $\boldsymbol{\sigma}_I$

$$J_{\sigma_i} = \lambda \sum_{j=1}^{k} \frac{\left\| \Psi(\tilde{\boldsymbol{X}})_{\delta_i \sim \mathcal{N}(0, \boldsymbol{\Sigma}_i = \sigma_i^2 \boldsymbol{I})} - s \right\|^2}{\sigma_S^2} - K \sum_{i=1}^{n} \log(\sigma_i)$$

Initialization

Variance Calculation

Dataset Iteration

Interpreter Initialization

Epoch-wise Training

Gaussian Noise Generation and Application

Hidden State Calculation with Added Noise

Loss Update

Training the Interpreter

Gaussian Noise Variance Update

Output

4. Experiments and Evaluations

4. Experiments and Evaluations

A. Experimental Setting

- **Datasets**

Utilizes HighD (captured by drone at 25fps around Cologne, Germany) and the public dataset NGSIM I-80 (captured at 10fps in 2005).

- **System**

Experiments conducted on a system equipped with an Intel Core i7 CPU, NVIDIA GeForce GTX 1080Ti GPU, 16GB RAM, running on Ubuntu 16.04 LTS OS.

B. Training Setting and Evaluation Metrics

- **Model Configuration**

Employs 128 hidden layers for the LSTM encoder/decoder and 32 convolution kernels for the TPA.

- **Optimization**

Utilizes the Adam optimizer with a learning rate of 10^(-4) and a mini-batch size of 128, with learning rate scheduling for performance improvement.

- **Evaluation Metrics**

Model performance evaluated using accuracy, precision, recall, F1-score for driving behavior classification, and RMSE for trajectory prediction accuracy.

$$RMSE = \sqrt{\frac{1}{N}\sum_{n=1}^{N}\left[(\boldsymbol{x}_n - \hat{\boldsymbol{x}}_n)^2 + (\boldsymbol{y}_n - \hat{\boldsymbol{y}}_n)^2\right]}$$

4. Experiments and Evaluations

C. Results and Comparison

Model Name	Description of the model
Class variational Gaussian mixture models (C-VGMMs)	Uses variational Gaussian mixture models with a Markov random field to classify driving maneuvers and predict trajectories.
GAIL	Extends the optimization of the Gated Recurrent Unit (GRU) to improve policy fidelity in driving policy predictions.
Social LSTM (S-LSTM)	Employs a fully connected pooling LSTM structure to predict vehicle trajectories.
MATF	Encodes the historical trajectories of multiple agents and scene context into a Generative Adversarial Network (GAN) with adversarial loss.
Convolutional social pooling (CS-LSTM)	Uses an LSTM structure that incorporates convolutional social pooling to encode surrounding vehicles as social tensors and considers multi-modal driving maneuvers.
SGCN	Utilizes a Sparse Graph Convolutional Network (GCN) based on sparse directed spatial graphs to model interactions between vehicles.
Scalable Network (SCALE-Net)	Employs an edge-enhanced graph convolutional neural network to reduce sensitivity to input data and improve prediction efficiency.
Multiple futures prediction (MFP)	Uses parallel RNNs with a shared weight encoder to encode past and future interactions of an agent and predict trajectories.
Multi-head attention social pooling (MHA)	Applies multi-head attention with an encoder/decoder to extract deep features of the target vehicle and surrounding vehicles, considering various input features such as speed, acceleration, and vehicle class.
TP2Net	The model proposed in this study

4. Experiments and Evaluations

C. Results and Comparison

TABLE I

RMSE OF EACH MODEL IN THE 5-S PREDICTION HORIZON

Dataset	Prediction horizon (s)	C-VGMMs[20]	GAIL [31]	S-LSTM [44]	MATF [32]	CS-LSTM[28]	SGCN [35]	SCALE-Net [34]	MFP [45]	MHA [3]	TP2Net
NGSIM	1	0.66	0.69	0.65	0.66	0.61	0.58	0.46	0.54	0.41	**0.30**
	2	1.56	1.51	1.31	1.34	1.27	1.18	1.16	1.16	1.01	**0.86**
	3	2.75	2.55	2.16	2.08	2.09	1.95	1.97	1.90	1.74	**1.52**
	4	4.24	3.65	3.25	2.97	3.10	3.03	2.91	2.78	2.67	**2.36**
	5	5.99	4.71	4.55	4.13	4.37	4.04	-	3.83	3.83	**3.37**
HighD	1	-	-	0.22	-	0.22	0.15	-	-	0.06	**0.05**
	2	-	-	0.62	-	0.61	0.38	-	-	0.09	**0.07**
	3	-	-	1.27	-	1.24	0.72	-	-	0.24	**0.19**
	4	-	-	2.15	-	2.10	1.16	-	-	0.59	**0.49**
	5	-	-	3.41	-	3.27	1.71	-	-	1.18	**0.98**

- **RMSE Comparison**

RMSE values of various models are compared in Table I, with results from studies not using the HighD dataset not provided.

- **Data Accuracy**

The HighD dataset showed lower RMSE than NGSIM, indicating higher accuracy and fewer errors.

- **Impact Factors**

Incorrect labeling negatively impacts network predictions.

- **Prediction Accuracy**

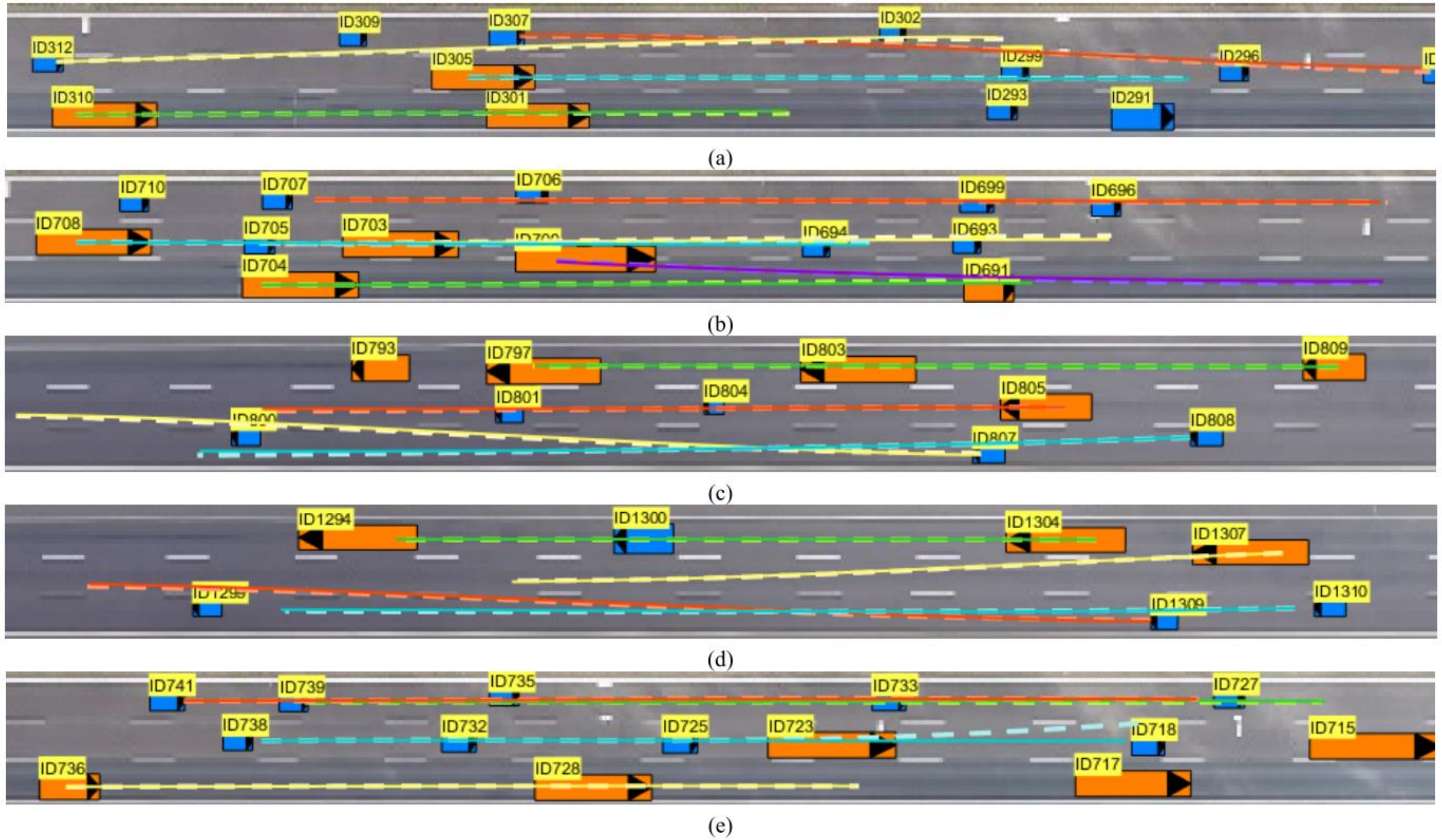
Short-term predictions show small errors, while long-term predictions highlight the importance of recognizing driving intentions due to larger errors.

- **Model Performance**

The proposed model demonstrates superior performance in both short-term and long-term predictions, offering a 15% accuracy improvement over the MHA model.

4. Experiments and Evaluations

C. Results and Comparison



Prediction instances of HighD dataset

- **Vehicle Representation**

Blue boxes represent cars, and orange boxes represent trucks.

- **Driving Direction**

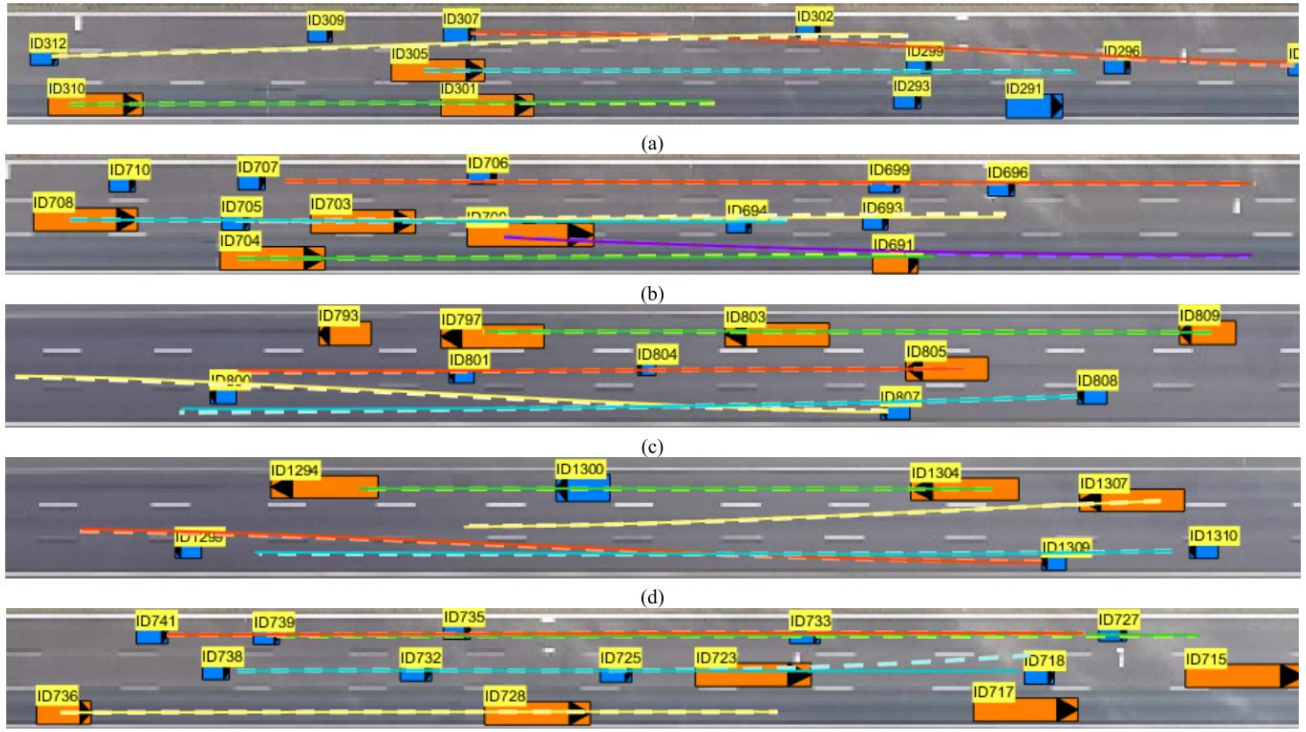
The direction of the vehicle is indicated by a triangle inside the box.

- **Trajectory Information**
  - Light-colored dotted line: Actual vehicle trajectory (Ground Truth)
  - Dark-colored solid line: Trajectory predicted by the model

- **Driving Pattern**
  - (a), (b), (e): Vehicles drive from left to right
  - (c), (d): Vehicles drive from right to left

4. Experiments and Evaluations

C. Results and Comparison



(a) (b) (c) (d) (e)

Model Prediction Instances
Various vehicle driving maneuvers and their prediction instances are presented.

Prediction Accuracy
The model shows good prediction performance for most driving maneuvers in the validation set.

Case of Prediction Error
There's an instance where a significant prediction error occurred during a lane-changing maneuver.

Specific Vehicle Prediction Failure
The network failed to predict the lane-changing maneuver of vehicle ID 738.

Cause of Prediction Failure
The presence of a vehicle (ID 739) on the left-front led the network to predict straight driving instead of a lane change.

Decrease in Prediction Accuracy
The prediction accuracy of TP2Net decreased when the actual lane change occurred.

Future Research Direction
Strategies for performance improvement in such scenarios are suggested as topics for future research.

4. Experiments and Evaluations

D. Prediction Error of Each Driving Maneuver

TABLE II

CLASSIFICATION PERFORMANCE OF EACH DRIVING MANEUVER

Lateral accuracy: 99.22% Longitudinal accuracy: 99.10%	Precision (%)	Recall (%)	F1-score (%)
LF	99.60	99.58	99.59
LLC	92.05	91.61	91.83
RLC	95.09	95.73	95.41
ND	99.40	99.68	99.54
HB	85.64	80.16	82.81
RA	84.55	71.90	77.71

- lane following (LF)
- left lane change (LLC)
- right lane change (RLC)

- normal driving (ND)
- hard braking (HB)
- rapid acceleration (RA)

- **Lateral Maneuvers**

The classification results for lateral driving maneuvers (e.g., lane changes) were generally satisfactory. However, the F1-scores for left lane change (LLC) and right lane change (RLC) were lower compared to left turn (LF)

- **Classification Errors**

In certain scenarios, such as the driving maneuver of the target vehicle in the next 3-4 seconds, there were often incorrect classifications. Solutions to this issue will be explored in future work.

- **Longitudinal Maneuvers**

The prediction accuracy for longitudinal driving maneuvers (e.g., acceleration or deceleration) was relatively low, possibly due to the unclear classification boundaries between different longitudinal maneuvers.

4. Experiments and Evaluations

D. Prediction Error of Each Driving Maneuver

TABLE III
RMSE OF EACH DRIVING MANEUVER IN THE 5-S PREDICTION HORIZON

	Prediction Horizon (s)	LF		LLC		RLC	
		Lon	Lat	Lon	Lat	Lon	Lat
MHA [3]	1	0.05	**0.01**	0.20	**0.03**	0.07	**0.03**
	2	0.07	**0.02**	0.32	**0.07**	0.12	**0.06**
	3	0.22	**0.06**	0.42	0.19	0.34	0.18
	4	0.54	**0.14**	0.88	0.45	0.79	0.43
	5	1.10	**0.22**	1.74	0.78	1.43	0.76
TP2Net	1	**0.04**	0.02	**0.08**	0.07	**0.06**	0.05
	2	**0.06**	0.03	**0.12**	0.10	**0.09**	0.08
	3	**0.17**	0.08	**0.27**	**0.17**	**0.21**	**0.14**
	4	**0.44**	0.18	**0.66**	**0.41**	**0.52**	**0.34**
	5	**0.88**	0.30	**1.30**	**0.63**	**1.04**	**0.54**

- **Prediction Error Increase**

Lateral driving maneuvers increased both longitudinal and lateral prediction errors, with a higher proportion of longitudinal prediction errors.

- **Comparative Analysis**

The proposed model was not as accurate as MHA in terms of longitudinal errors (especially for LF) and short-term lateral maneuver errors, but showed significantly reduced errors otherwise.

- **Long-term Prediction Performance**

TP2Net is indicated to better predict the trajectory in the long-term prediction horizon based on the extracted hidden features.

- **Specific Error Comparison**

The prediction error for LLC in longitudinal maneuvers was significantly larger than that for RLC in lateral maneuvers, likely due to more active lane changes and greater speed variance associated with LLC.

- **Lateral Error Analysis**

The lateral error associated with LF was smaller than that associated with either LLC or RLC.

4. Experiments and Evaluations

E. Comparison and Ablation Experiment

Experiment Condition	*Description*
Less feature	Only the x and y coordinates of the target and surrounding vehicles were used as input.
Only hT	Predictions were based solely on the target vehicle's encoded tensor hT.
No hT	The target vehicle's encoded tensor hT was removed from the model.
No ht	The TPA output tensor ht was removed from the model.
No hA	The VOI inception tensor hA was removed from the model.
Simple VOI	Replaced VOI inception using 1 × 1 and 3 × 3 convolution kernels to convolve the stacking tensor Estk.
MHA	Replaced the TPA mechanism with the MHA mechanism.
GCN	Used a graph convolutional neural network to model the interactions among surrounding vehicles.
TP2Net	The model proposed in this study

TABLE IV

RMSE of Each Comparison and Ablation Model in the 5-Second Prediction Horizon

	Prediction Horizon (s)	Less feature	Only h_T	No h_T	No h_t'	No h_A	Simple VOI	MHA	GCN	TP2Net (proposed)
Longitudinal	1	0.181	0.093	0.071	0.051	0.063	0.049	0.045	0.044	**0.043**
	2	0.511	0.133	0.100	0.072	0.090	0.070	0.065	0.067	**0.063**
	3	1.055	0.264	0.209	0.199	0.218	**0.177**	0.185	0.184	**0.177**
	4	1.760	0.682	0.510	0.506	0.563	0.454	0.476	0.486	**0.453**
	5	2.645	1.377	1.028	1.022	1.164	0.943	0.973	0.987	**0.930**
Lateral	1	0.133	0.075	0.049	0.052	0.027	0.029	0.053	0.034	**0.023**
	2	0.255	0.097	0.059	0.057	0.043	0.044	0.090	0.043	**0.037**
	3	0.367	0.159	0.100	0.099	0.092	0.087	0.144	**0.081**	0.082
	4	0.480	0.284	0.193	0.206	0.196	0.189	0.240	**0.177**	0.185
	5	0.562	0.435	0.310	0.335	0.326	0.317	0.345	**0.298**	0.310

- **Importance of Basic Inputs**

Using only the position information of the target vehicle significantly reduced prediction accuracy, suggesting the need for additional kinematic features.

- **Interplay between TPA and hT**

Removing hT reduced lateral errors, while removing ht reduced longitudinal errors, indicating that the two tensors complement each other.

- **Interaction with Surrounding Vehicles**

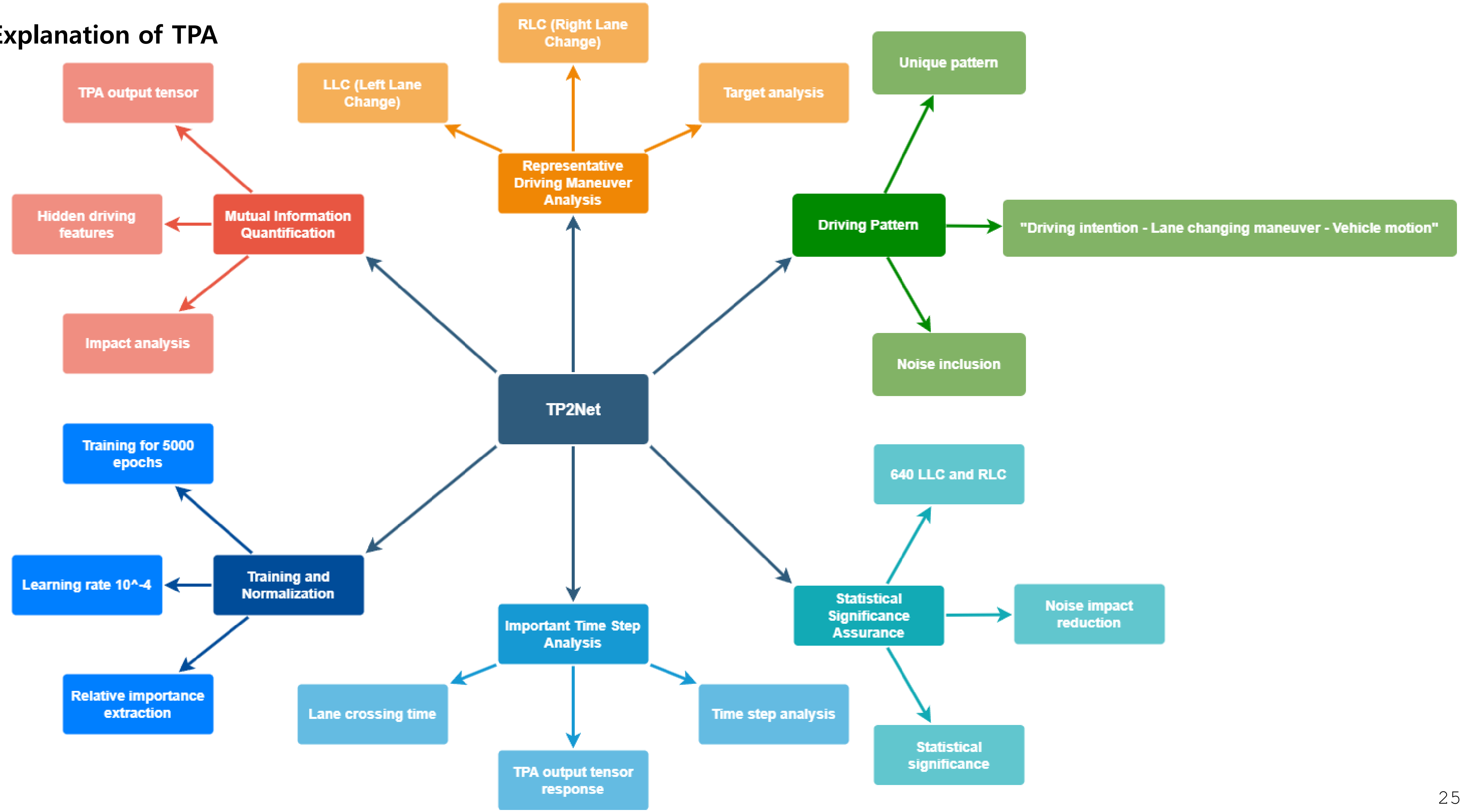
Lack of interaction information with surrounding vehicles increased errors, with VOI inception improving accuracy.

- **Performance of TPA and GCN**

TPA outperformed MHA, showing effectiveness in extracting hidden driving features. GCN was useful for lateral predictions but didn't match TP2Net's performance in longitudinal predictions.

4. Experiments and Evaluations

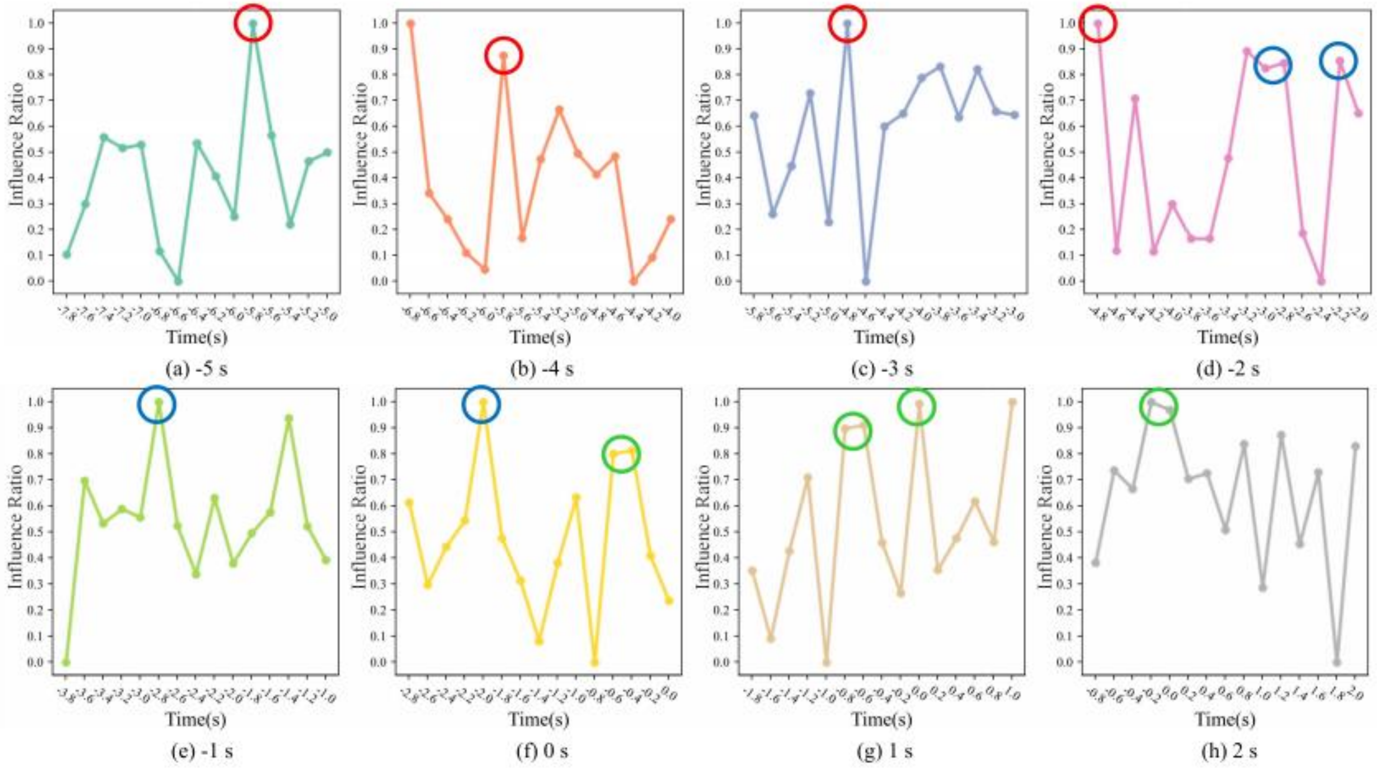
F. Output Explanation of TPA



4. Experiments and Evaluations

F. Output Explanation of TPA

the normalized relative influence between
the TPA output tensor and each time step in the LLC condition

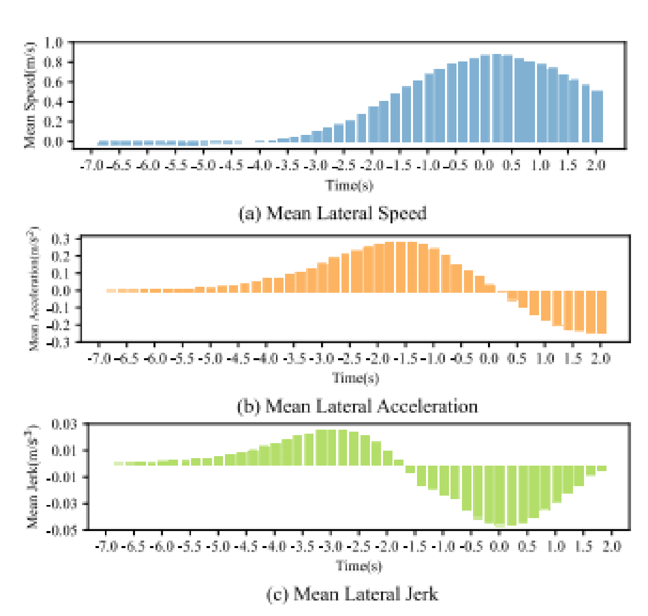


- **Response Peaks**
The analysis identified moments with clear response peaks in at least two subfigures.

- **First Stage (Red Circle)**
Between -4.8 s and -5.8 s, this period is interpreted as the potential intention to change lanes by the driver.

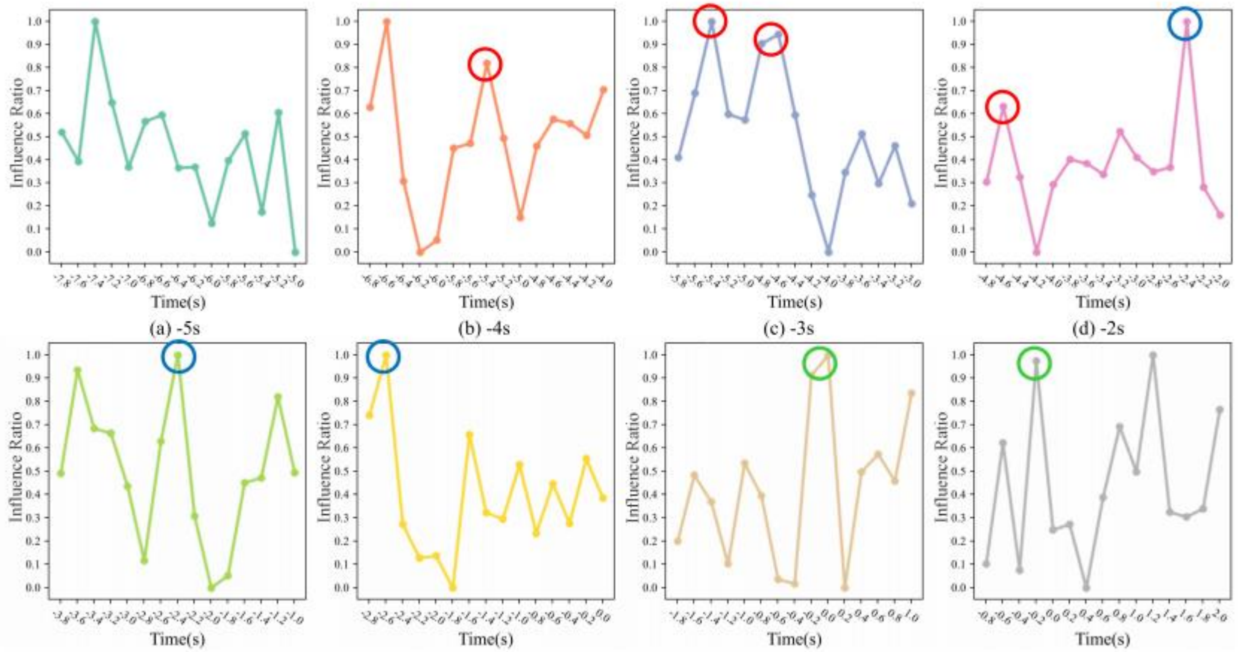
- **Second Stage (Blue Circle)**
Between -2.2 s to -2.0 s and -3.0 s to -2.8 s, this period is interpreted as the stage where the driver begins to turn the steering wheel.

- **Third Stage (Green Circle)**
At 0 s and between -0.6 s to -0.4 s, this period is interpreted as the stage where the vehicle crosses the lane line.



4. Experiments and Evaluations

F. Output Explanation of TPA



(a) -5s (b) -4s (c) -3s (d) -2s
(e) -1s (f) 0s (g) 1s (h) 2s

- **First Stage (Red Circle)**

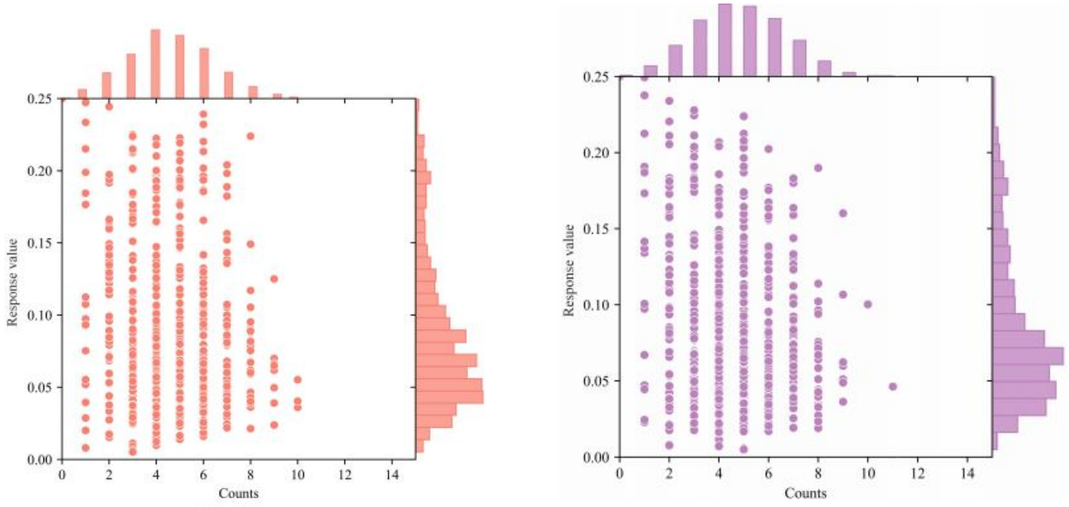
-4.6 s and -5.4 s represent moments that may indicate an intention to change lanes, suggesting the preparation phase for lane changing.

- **Second Stage (Blue Circle)**

The period between -2.6 s and -2.4 s marks the beginning of the steering wheel turn by the driver, indicating the execution phase of the lane-changing maneuver.

- **Third Stage (Green Circle)**

The moments between -0.2 s and 0 s represent the vehicle actually crossing the lane line, indicating the completion phase of the lane change.



(a) 0s LLC (b) 0s RLC

- Analyzes high response values for 320 trajectories in LLC and RLC maneuvers.
- Values closer to 0 have a greater impact on prediction.
- Mean values are determined by a sliding window and recorded at peak response counts.
- LLC and RLC show similar distributions of high responses, suggesting around 4 key time steps within 3 seconds before lane changes.
- TPA demonstrates its ability to stably extract important time steps related to hidden driving features, even after using Gaussian noise.

5. Conclusion

5. Conclusion

- **TP2Net and TPA**

TP2Net integrates TPA, which uses Gaussian noise to quantitatively assess changes in input and hidden tensors, highlighting the significance of specific dimensions.

- **Identification of Key Stages**

Analyzing 10 seconds around a lane change at 1-second intervals revealed three critical stages. The model detects the driver's lane-change intent approximately between -4.5 to -6.0 seconds.

- **Efficiency of TPA**

TPA adeptly captures the pattern of driving intention, maneuver, and vehicle motion, extracting hidden driving features for trajectory prediction.

- **Ablation Study Outcomes**

The results confirm that TPA precisely extracts lateral maneuvers of the target vehicle, and encoding of the target vehicle compensates for significant longitudinal errors, showcasing their complementary nature.

- **Interaction and Accuracy**

Extracting interactions between the target and surrounding vehicles on a multi-scale significantly enhances prediction accuracy.

- **Prediction Performance**

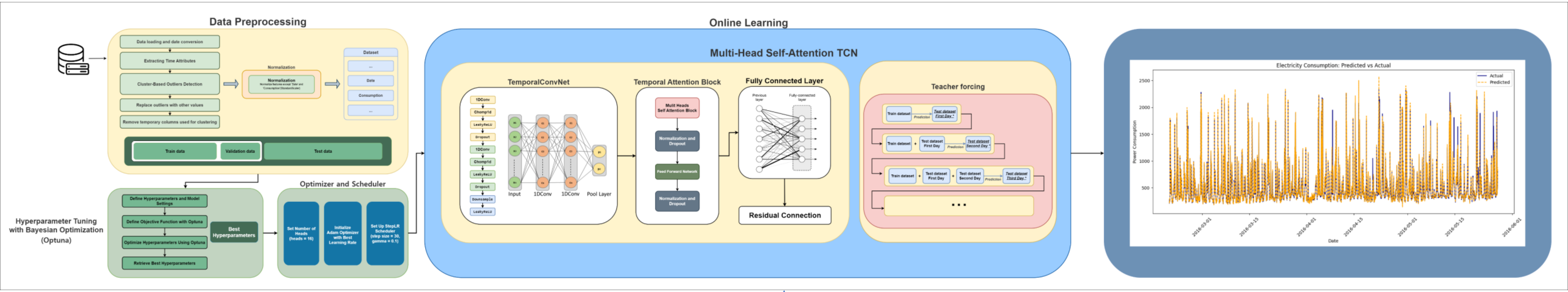
The proposed model boosts prediction accuracy in real-world scenarios, benefiting from faster inference and reduced computational demands.

- **Future Research Directions**

Addressing the limitation of predicting only one vehicle at a time and the decline in accuracy for certain maneuvers in the next 4 to 5 seconds will be the focus of future research. It will also involve considering more surrounding vehicles and enabling the network to select vehicles of interest through end-to-end learning.

6. How to apply

6. How to apply



Gaussian noise

Thank You