Research on Engineering Structures & Materials
www.jresm.org
Research Group

Research Article

End-to-end decision-making algorithm for intelligent driving based on Conv-LSTM

Hui Qian [*,a]

School of New Energy Vehicles and Transportation Engineering, Nantong Institute of Technology, Nantong, China

Article Info

Article History:

Received 13 Aug 2026

Accepted 08 Oct 2026

Keywords:

Intelligent driving;
End-to-end decision-making;
Human driving behavior;
Conv-LSTM;
Self-attention memory

Abstract

To address error accumulation and limited adaptability to complex road conditions in traditional modular intelligent-driving architectures, this study proposes an end-to-end decision-making algorithm based on a convolutional LSTM network and a self-attention memory module. Incorporating human driving logic, the basic model integrates spatiotemporal feature extraction, historical state modeling, and longitudinal and lateral control. Improved residual units and depth wise separable convolution enhance feature representation efficiency, while future-sequence prediction supports task optimization. A self-attention memory module is further introduced to strengthen long-term temporal and spatial modeling, forming an optimized end-to-end decision model. Simulation results show a 100% success rate and 0% collision rate in simple scenarios, and a 94% success rate with a 4% collision rate in complex scenarios. Prediction accuracy reaches 88.7% for steering and 89.1% for speed. Driving trajectory, heading angle, and steering wheel angle closely match expert driving behavior. The proposed method also demonstrates good ride smoothness and real-time decision-making performance, confirming its effectiveness in improving the accuracy, robustness, and safety of autonomous driving and providing useful technical support for advanced end-to-end autonomous driving systems. The above experimental results are only based on the CARLA simulation platform Town4 scenario, which is constrained by the simulation scenario. The generalization ability of the model in more complex urban scenarios needs further verification.

© 2026 MIM Research Group. All rights reserved.

1. Introduction

With the rapid advancement of autonomous driving technology towards higher levels of intelligence, the traditional modular architecture that separates perception, decision-making, and control is gradually revealing obvious flaws. The architecture requires processing information step by step, with tight coupling between modules, and errors are prone to accumulate during transmission. In complex dynamic scenarios such as urban roads and multi-vehicle interaction, environmental adaptability and decision robustness are difficult to meet the requirements of safe driving. End-to-end decision-making methods, which directly map environmental inputs to vehicle control instructions, simplify system processes and reduce coupling errors, have become an important research direction in autonomous driving decision-making [1-2]. However, at present, there are still several core theoretical challenges in end-to-end autonomous driving driven by deep learning: the interpretability of the model decision-making process is poor, and it is difficult to provide complete safety constraints on the output behavior; At the same time, it is highly dependent on training data and lacks sufficient security capabilities in distributed scenarios, which hinders its practical implementation and application. However, existing end-to-end models

*Corresponding author: qianhui891150@163.com
[a]orcid.org/0009-0008-0452-0915
DOI: http://dx.doi.org/10.17515/resm2026-2184vk0813rs

generally have problems such as insufficient modeling of driving temporal features, weak ability to capture long-distance dependencies, and easy loss of spatial information. Moreover, most methods do not fully draw on the perceptual decision-making logic of human driving, and make insufficient use of vehicle historical states and driving memory, resulting in poor model interpretability, high data dependency, and limited generalization ability in complex scenarios [3-4].

To address these issues, an end-to-end decision-making algorithm for intelligent driving based on Convolutional Long Short-Term Memory (Conv-LSTM) networks and self-attention memory module (SAMM) is designed in combination with the perception-decision-making-control closed-loop mechanism of human driving behavior. This method takes sequential images and historical vehicle states as inputs, and employs an LSTM network to fuse spatiotemporal features. Improved residual units and depth wise separable convolutions are utilized to enhance feature extraction efficiency, and a SAMM is introduced to strengthen the modeling capability for long-range temporal and spatial information. An end-to-end decision-making model that balances accuracy, real-time performance, and safety is constructed, thereby providing an effective technical solution for autonomous driving decision-making in complex scenarios. The innovations of this research lie in the following aspects: a three-subnetwork collaborative architecture comprising spatiotemporal feature extraction, historical state feature extraction, and longitudinal-lateral control decision-making is adopted, which simultaneously inputs scene sequence images and vehicle historical state information to compensate for the deficiency of traditional end-to-end methods that neglect driving history states. An improved residual unit is designed, which enhances feature representation efficiency and reduces computational cost through channel splitting, depth wise separable convolution, and channel shuffling. A SAMM is proposed, which integrates the gated memory mechanism of LSTM-like networks with self-attention to achieve efficient fusion of historical memory and current features. It should be noted that this article only simulates human driving behavior using historical vehicle speed and steering angle sequences, and does not fully reproduce all physiological and psychological mechanisms of driver cognition perception decision-making. This model is an engineering approximation modeling of human driving logic. The research conclusions are only applicable to end-to-end decision-making tasks in simulation environments and do not represent a complete replica of the complete decision-making mechanism of real human drivers. The model has good real-time inference performance in simulation environments, but adaptation testing has not yet been carried out for embedded hardware such as Jetson Orin in vehicles. The deployment of real vehicles will be constrained by hardware computing power, memory bandwidth, and other conditions. The inference frame rate obtained in the simulation environment cannot be directly equivalent to the actual operating effect of the vehicle hardware. The adaptation of computing power to real vehicle hardware will be a key research focus in the future.

2. Literature Review

End-to-end decision-making, which can effectively alleviate problems such as the accumulation of coupling errors and poor adaptability to complex scenarios in traditional modular intelligent driving architectures, has become a research hotspot in the field of autonomous driving. Silva et al. proposed an anonymous vehicle-mounted AD hoc network topology learning method to address the issue of trajectory privacy leakage in vehicle-connected autonomous driving, based on coordinate mesh construction graph model training. Selective communication can protect vehicle privacy and match scenarios to select the optimal graph learning framework [5]. Xu et al. proposed a multimodal fusion deep reinforcement learning approach to address the training and decision-making challenges brought by high-dimensional inputs in end-to-end autonomous driving, and built a shared training framework to filter out invalid features and reduce computations and errors. Experiments show that the method converges quickly and significantly improves training efficiency and decision-making performance [6]. Chen Y et al. proposed an intent-guided heuristic double asymptotically widened partially observable Monte Carlo programming method to address the problem of inaccurate intent estimation caused by chaotic branching of belief trees and particle deprivation when partially observable Monte Carlo programming is applied to autonomous driving planning. The method improves particle resampling and trims the action and observation spaces,

enhancing the efficiency of belief node exploration and the accuracy of intention estimation. The results show that the proposed model is robust, with a decision success rate of 96.88% in ramp scenarios [7].

Combining the advantages of convolution and LSTMs, Conv-LSTM can effectively extract spatio-temporal correlation features, solve the problem of traditional LSTM losing spatial structure information, and has stronger dynamic modeling ability for sequential images. Biradar S et al. designed a detection method based on Conv-LSTM in response to the difficulty in detecting hate speech in Hindi-English code-mixed text in multilingual countries such as India and the fact that existing studies have mostly focused on monolingual data. The method fuses spatio-temporal features and combines early fusion Siamese architectures to build an integrated network, improving recognition. The results show that the method outperforms existing baseline models in hate speech recognition in low-resource code mixture scenarios [8]. Murali E et al. designed a detection method based on likelihood statistical texture feature representation and Conv-LSTM to address the problem of high difficulty in automatic detection of surveillance video anomalies due to uncertain anomaly definitions. The method uses co-occurrence and stationary appearance representations to construct spatial activity patterns, and combines Conv-LSTM to learn normal behavior patterns. The results show that the method outperforms existing anomaly detection methods in terms of equivalent error rate and area under the curve on four benchmark datasets [9]. Muiruri S M et al. proposed a hybrid detection model of Conv-LSTM combined with support vector machine to address the problems of difficult detection of violent behavior in surveillance videos and insufficient public safety assurance. The model combines the spatial features of the Convolutional Neural Network (CNN), the temporal dynamics of the LSTM and the classification capabilities of the support vector machine, and uses transfer learning to improve the feature extraction effect. The results show that the model's accuracy can reach 97.3%, and it has excellent generalization ability [10].

In summary, although existing related research has made some progress in areas such as autonomous driving decision-making and temporal feature modeling, there are still obvious deficiencies: most end-to-end decision-making methods do not fully integrate the logic of human driving behavior and ignore the vehicle's historical state and long-term temporal memory. Conv-LSTM has limited capability in modeling long-range dependencies and is prone to losing spatial information. Moreover, it is rarely combined with lightweight architectures and multi-task learning to improve efficiency. To address these deficiencies, this study proposes an end-to-end decision-making algorithm for intelligent driving based on Conv-LSTM and a SAMM.

3. Intelligent Driving Decision-Making Method Based on Conv-LSTM and Self-Attention Memory Module

3.1 Intelligent Driving Decision-Making Method Based on Human Driving Behavior and Conv-LSTM

Traditional intelligent driving uses a modular hierarchical architecture where perception, decision-making, and control are split step by step. Module coupling errors tend to accumulate, and there is insufficient adaptability in complex road conditions, making it hard to satisfy the safety driving requirements of advanced autonomous driving. End-to-end decision-making discards the step-by-step disassembly model and directly maps vehicle control instructions from the environment input [11-12]. The essence of end-to-end intelligent driving decision-making is imitation learning. Therefore, to reduce data dependence and enhance interpretability, the study proposes an end-to-end decision-making method based on human driving behavior and Conv-LSTM. The driving behavior model is shown in Figure 1. In Figure 1, in the driving behavior model, the driver perceives the road environment and vehicle status through visual, auditory, and physical sensations. Combined with driving experience, skills, driving tasks, traffic information, and map data, through brain information processing and decision-making planning, the vehicle is controlled through throttle, steering wheel, and braking, and the vehicle status and road environment are fed back to the driver, forming a closed-loop process. Driving behavior is divided into three stages: perception, decision-making, and control. Among them, perception errors account for 42% of the causes of

accidents, and visual perception plays a dominant role (80%). Human drivers predict future driving paths and speeds based on the dynamic environment and the historical state of the vehicle through short-term/long-term memory. End-to-end intelligent driving decisions imitate this logic, using CNN to extract visual signals to simulate perception, completing decisions through deep neural networks, and LSTM networks to model time series data for dynamic predictive decision-making [13-15]. The structure of the End-to-end intelligent driving decision model integrating human driving behavior (EIDIHDE) that integrates human driving behavior is shown in Figure 2.

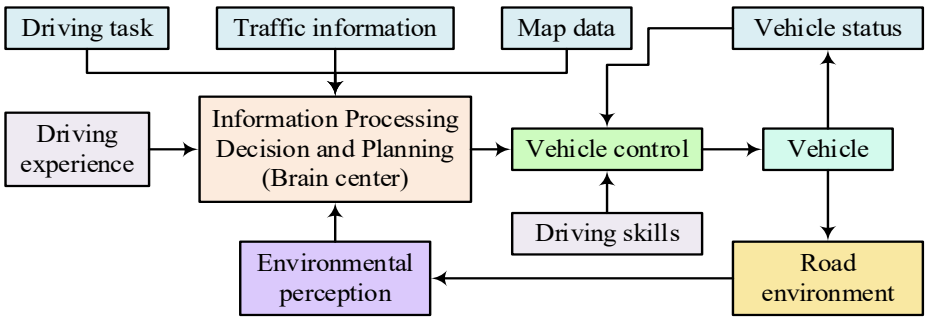


Fig. 1. Driving behavior model

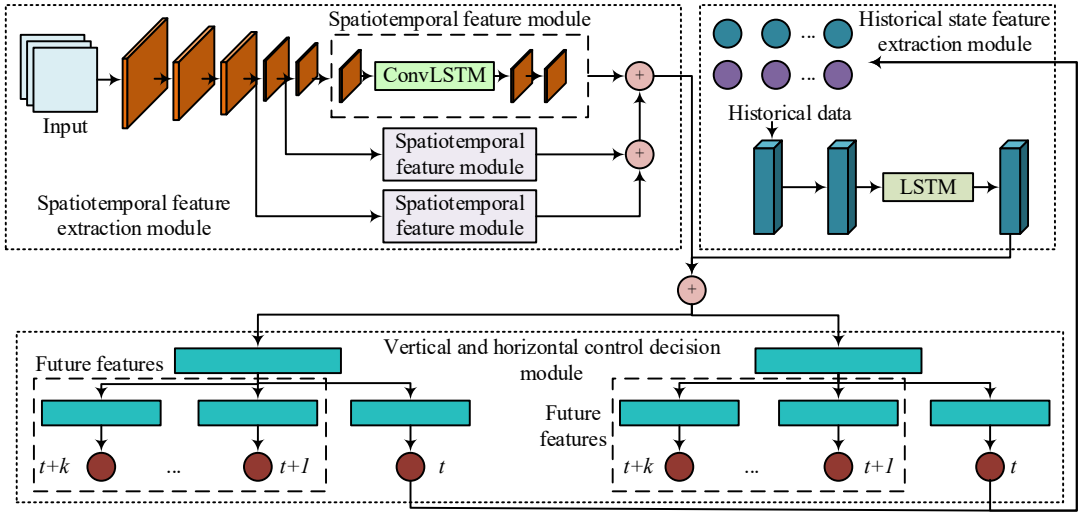


Fig. 2. The structure of EIDIHDE

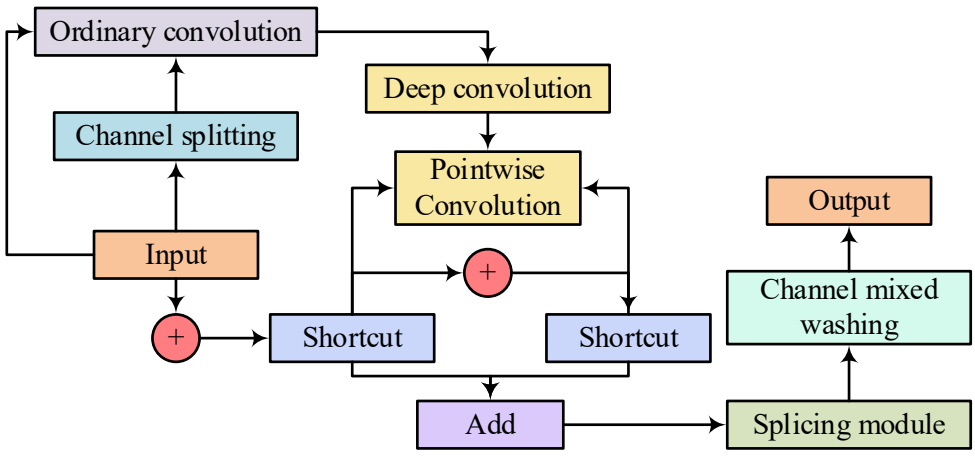


Fig. 3. Improved residual unit

From Figure 2, EIDIHDE is composed of three sub-networks: spatio-temporal feature extraction, historical state feature extraction, and longitudinal and transverse control decision-making. The

spatio-temporal feature sub-network takes sequential images as input and uses convolutional layers and Conv-LSTM to extract multi-scale spatio-temporal features. The historical state subnetwork takes in time series data of steering wheel angle and vehicle speed, and excavates time series features through the fully connected layer and LSTM. The two types of features are fused and input into the longitudinal and transverse control decision subnetwork, which simultaneously outputs the control quantity at the current moment and the auxiliary features for the future sequence prediction task, improving the overall learning efficiency and decision-making performance of the model through the supervised primary task learning. The backbone network of EIDIHDE is composed of encoders responsible for image feature extraction and compression, and consists of five sub-layers. Except for the first layer, the rest are composed of downsampling layers and improved residual units. The downsampling units perform a dual-branch parallelization of 3*3 convolution and 2*2 pooling, and then cascade the results to reduce the size of the feature map while increasing the number of channels and expanding the receptive field. The improved residuals are shown in Figure 3.

In Figure 3, the improved residual unit first splits the input feature map into channels. The two branches extract features through normal and depth-separable convolution respectively, concatenate and connect to the original input residuals, and then output through channel shuffle to enhance feature representation and improve information interaction efficiency. Depth wise separable convolution consists of depth wise convolution and point-by-point convolution. The computational cost of depth wise convolution is shown in Equation (1).

$$FLOP_D = H \times W \times K^2 \times C_{in} \tag{1}$$

In Equation (1), $FLOP_D$ represents the amount of computation for depth-separable convolution; $H \times W$ denotes the size of the input feature map; K is the size of the convolution kernel; C_{in} means the amount of input channels. The amount of convolutional computation at each point is shown in Equation (2).

$$FLOP_P = H \times W \times C_{in} \times C_{out} \tag{2}$$

In Equation (2), $FLOP_P$ represents the amount of computation for point wire convolution; C_{out} denotes the amount of output channels. The total computation of depth-separable convolution is shown in Equation (3).

$$FLOP_T = H \times W \times C_{in} \times (K^2 + C_{out}) \tag{3}$$

In Equation (3), $FLOP_T$ represents the total amount of computation. Each operation item is calculated based on the same set of spatial dimensions to ensure that the dimensions of the front and back matrices match. For the spatio-temporal feature extraction module, the structure of "1*1 convolution dimensionality reduction +Conv-LSTM+3*3 convolution +1*1 convolution dimensionality reduction" is used to reduce the number of parameters and accelerate training. Meanwhile, since the end-to-end approach of mapping control instructions directly from images ignores the consideration of vehicle historical states in human driving, the study supplements the extraction of vehicle historical state features. The study uses two-layer one-dimensional convolution to extract angular and speed temporal features, combines with LSTM to mine temporal patterns, and then fuses historical and time-domain features. Meanwhile, EIDIHDE adds an auxiliary task of future sequence prediction to enhance the network's feature extraction ability through loss constraints. The loss function of EIDIHDE is denoted in Equation (4). In Equation (4), J represents the total loss function; ω_i represents weights; J_i represent the loss function of the subtask; $L_i(t)$ represents the predicted loss of the subtask at time t, that is, the mean square error function; k means the total amount of time steps; γ_j represents the weight coefficient.

$$\begin{cases} J = \sum_{i=1}^{2} \omega_i \times J_i \\ J_i = Loss_i(t) + \sum_{j=1}^{k} \gamma_j \times Loss_i(t+j), \gamma_j = 1/j \end{cases} \tag{4}$$

The study uses fixed heuristic weights to gradually reduce the weight of long-term prediction loss as the prediction step size increases, which is consistent with the prior experience that short-term prediction has a greater impact on decision-making in driving tasks. The Kendall uncertainty automatic weighting method was not adopted due to the limited size of the simulation dataset, which makes automatic weighting prone to unstable training and weight oscillations. Therefore, prior heuristic weights were selected to achieve multi task loss balancing. By using the above method, an end-to-end decision algorithm for intelligent driving is constructed.

3.2 An Improved Intelligent Driving Decision-Making Method Integrating Self-Attention Memory Modules

Although end-to-end decision-making for intelligent driving can be achieved through the above-mentioned methods, the Conv-LSTM has limited ability to model long-distance temporal dependencies, and there are problems such as easy loss of spatial information and low efficiency of receptive field expansion [16-17]. Therefore, to enhance decision-making performance, a SAMM is designed to improve the Conv-LSTM. The SAMM introduces memory units similar to the LSTM gated structure to store long-distance spatial memory, and then draws on the self-attention mode to fuse the memory features with the features of the current time step to solve the problem of temporal information loss. The structure of the SAMM is denoted in Figure 4.

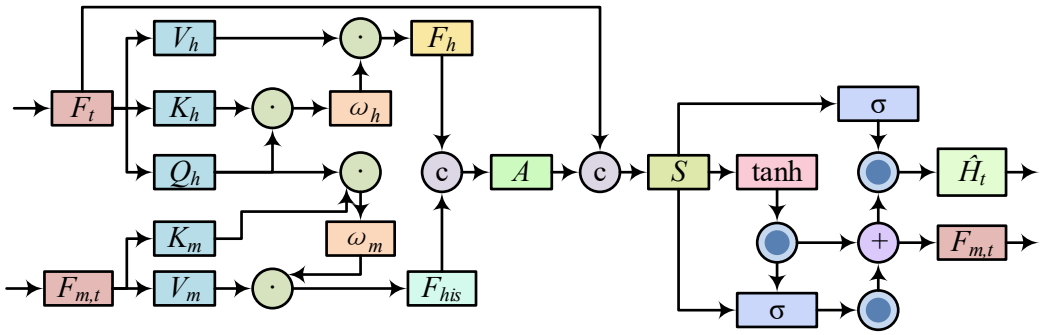


Fig. 4. Structure of SAMM

In Figure 4, the SAMM consists of two parts: feature aggregation and memory update. The input is the current time step feature and the memory unit of the previous moment. The feature aggregation part performs linear transformations on the current time step feature and the memory unit at the previous moment respectively to generate queries, keys, and values. The attention weights of the current feature and the memory feature are calculated through self-attention, and then multiplied with the corresponding value matrix to obtain weighted features. The fused features are obtained by channel concatenation and linear transformation of the weighted features. The memory update part uses a gated structure, controlling the information flow through the update gate and input gate, combining the fused feature with the memory unit of the previous moment to obtain the updated memory unit and the current output feature, ultimately achieving the effective fusion and transfer of the current feature with historical memory. The weighted feature calculation formula is shown in Equation (5).

$$\begin{cases} Q_h = W_{hq}F_t \\ K_h = W_{hk}F_t \\ V_h = W_{hv}F_t \\ \omega_h = softmax\left(Q_h^T K_h/\sqrt{d_k}\right) \\ F_h = \omega_h V_h \end{cases} \tag{5}$$

In Equation (5), Q_h represents the query vector generated by the current feature; W_{hq}, W_{hk}, and W_{hv} are learnable weight matrices; K_h represents the key vector generated by the current feature; V_h represents the value vector generated by the current feature; F_t represent the current feature; $\omega_h$represents the weight of the current feature; $1/\sqrt{d_k}$X represents the scaling factor suppressing dot product amplitude; F_h represents weighted features. The formula for calculating weighted historical memory features is shown in Equation (6).

$$\begin{cases} Q_m = W_{mq}F_{m,t-1} \\ K_m = W_{mk}F_{m,t-1} \\ V_m = W_{mv}F_{m,t-1} \\ \omega_m = softmax\left(Q_m^T K_m/\sqrt{d_k}\right) \\ F_{his} = \omega_m V_m \end{cases} \tag{6}$$

In Equation (6), Q_m represents the query vector generated by the historical memory feature; W_{mq}, W_{mk}, and W_{mv} are learnable weight matrices; K_m represents the key vector generated by historical memory features; V_m represents the value vector generated by historical memory features; $F_{m,t-1}$ represents the historical memory features of a moment $t-1$; ω_m is the weight of historical memory features; F_{his} represents weighted historical memory features. The aggregated feature calculation formula is shown in Equation (7).

$$A = W_A[F_h; F_{his}] \tag{7}$$

In Equation (7), A represents the aggregation feature; W_A represents weights. The memory update part is designed to adaptively update historical memory states, allowing the module to better capture long-distance spatial dependencies. This part first concatenates the current time-step feature with the aggregated feature to obtain the new feature, and then updates it using a gated structure similar to LSTM [18-20]. The fusion feature calculation formula is shown in Equation (8).

$$\begin{cases} S = [F_t; A] \\ i_t' = \sigma(W_{mi} * S + b_{mi}) \\ U_t = tanh\left(W_{mg} * S + b_{mg}\right) \\ F_{m,t} = (1 - i_t') \odot F_{m,t-1} + i_t' \odot U_t \end{cases} \tag{8}$$

In Equation (8), S represents the feature after concatenating the single signature feature with the aggregated feature; i_t' is the input gate; σ indicates the Sigmoid function; W_{mi} and W_{mg} both represent the weight matrices; b_{mi} and b_{mg} both represent the bias matrices; U_t represents fused features. The output module also mimics the output gate structure of the force LSTM, whose calculation formula is shown in Equation (9).

$$\begin{cases} o_t' = \sigma(W_{mo} * S + b_{mo}) \\ \widehat{H}_t = o_t' \odot F_{m,t} \end{cases} \tag{9}$$

In Equation (9), o_t' represents the output gate; W_{mo} indicates the weight matrix of the output gate; b_{mo} means the bias matrix of the output gate; $\widehat{H}_t$ means the hidden state. The structure of the improved Conv-LSTM with the introduction of the SAMM is shown in Figure 5.

As shown in Figure 5, the improved Conv-LSTM unit integrates a SAMM on top of the traditional Conv-LSTM. The input feature is convolved with the hidden state at the previous moment, and the cell state is updated through the forget gate and input gate, and then the current hidden state is obtained through the output gate. The state and the memory feature of the previous moment are input into the SAMM to complete feature aggregation and memory update, and the new hidden

state and memory feature are output, thereby achieving the fusion of spatiotemporal information and long-distance memory. The calculation formulas for the forgetting gate, input gate, and output gate of the improved Conv-LSTM are shown in Equation (10).

$$\begin{cases} f_t = \sigma\left(W_f * X_t + W_{hf} * \hat{H}_{t-1} + b_f\right) \\ i_t = \sigma\left(W_i * X_t + W_{hi} * \hat{H}_{t-1} + b_i\right) \\ o_t = \sigma\left(W_o * X_t + W_{ho} * \hat{H}_{t-1} + b_o\right) \end{cases} \tag{10}$$

In Equation (10), f_t is the forgetting gate; i_t denotes an input gate; o_t denotes an output gate; W_f, W_i, W_o, W_{hf}, W_{hi}, and W_{ho} are weight matrices; b_i, b_i, and b_o are biased.

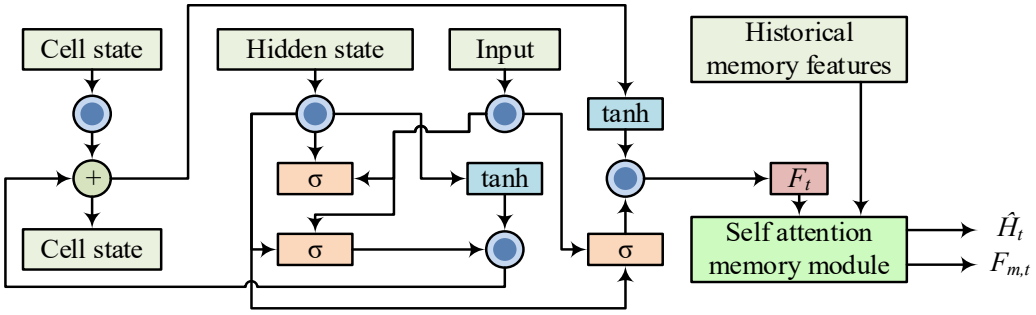


Fig. 5. Structure of improved Conv-LSTM

The cell state calculation formula is shown in Equation (11).In Equation (11), C_t is the state of the cell; $\hat{c}_t$ denotes candidate cell states; W_c and W_{hc} both represent weights; X_t is input data; b_c indicates bias. The formula for calculating the hidden state is shown in Equation (12).

$$\begin{cases} C_t = f_t \odot C_{t-1} + i_t \odot \hat{c}_t \\ \hat{c}_t = tanh\left(W_c * X_t + W_{hc} * \hat{H}_{t-1} + b_c\right) \end{cases} \tag{11}$$

$$\hat{H}_t = o_t \odot tanh(C_t) \tag{12}$$

Once the improved Conv-LSTM has been constructed, it is utilized to replace the original Conv-LSTM structure within EIDIHDE, yielding the improved end-to-end intelligent driving decision model integrating human driving behavior (IEIDIHDE). The module fusion process is as follows: the hidden state tensor output by the standard Conv-LSTM retains the original spatial and channel dimensions unchanged, and is input into the SAMM module along with the memory tensor output by the SAMM at the previous moment; First, complete feature aggregation to obtain fused features, and then update the memory tensor and output new hidden states through gating units. The tensor dimension remains consistent throughout the entire process, without dimension transformation or clipping, achieving the fusion of Conv-LSTM temporal gating and SAMM long-term memory. The overall pipeline of IEIDIHDE is depicted in Figure 6.

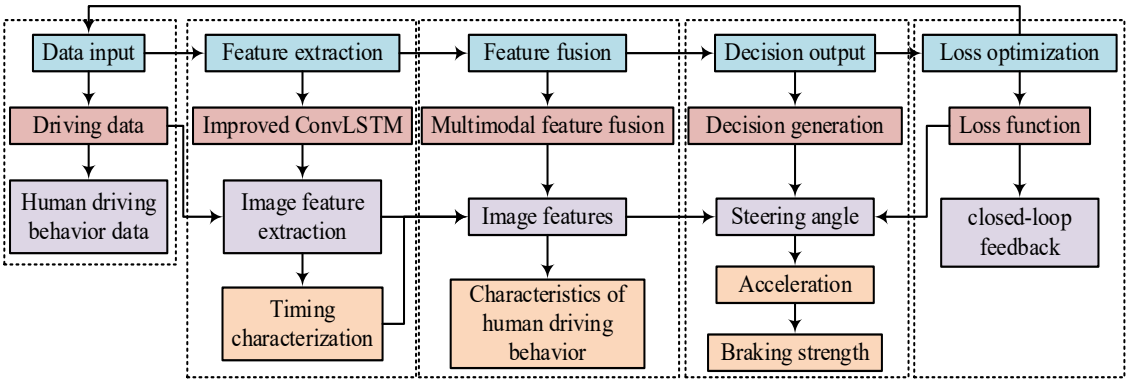


Fig. 6. Process of IEIDIHDE

As shown in Figure 6, the IEIDIHDE model takes continuous driving scene images and vehicle historical state sequences as inputs, extracts image features through the backbone encoder and the improved residual unit, and performs spatio-temporal feature fusion through the improved Conv-LSTM. Meanwhile, the temporal features of the vehicle's historical state are extracted through one-dimensional convolution and LSTM, and the two types of features are fused and sent into the decision network to complete the multi-task decision-making with the current control quantity output as the main task and the future sequence prediction as the auxiliary task. Finally, the network parameters are optimized based on the weighted loss backpropagation of the primary and secondary tasks, and the vehicle control instructions are finally output and combined with vehicle and environmental feedback to form an end-to-end closed-loop decision.

4. Test results of the Intelligent Driving End-to-End Decision-Making Method

4.1 Simulation Analysis Results

To assess the intelligent decision-making performance of the IEIDIHDE proposed in the study, simulation tests were conducted using the CARLA platform, and the test section was Town4. Research on collecting expert driver driving data based on the CARLA simulation platform, reproducing human driving operations through a built-in expert controller, with a total driving time of approximately 7.2 hours in the dataset; The training set, validation set, and test set are divided in a ratio of 7:2:1. The model is trained using the Adam optimizer, with a learning rate of 0.0001 and a batch size of 16.

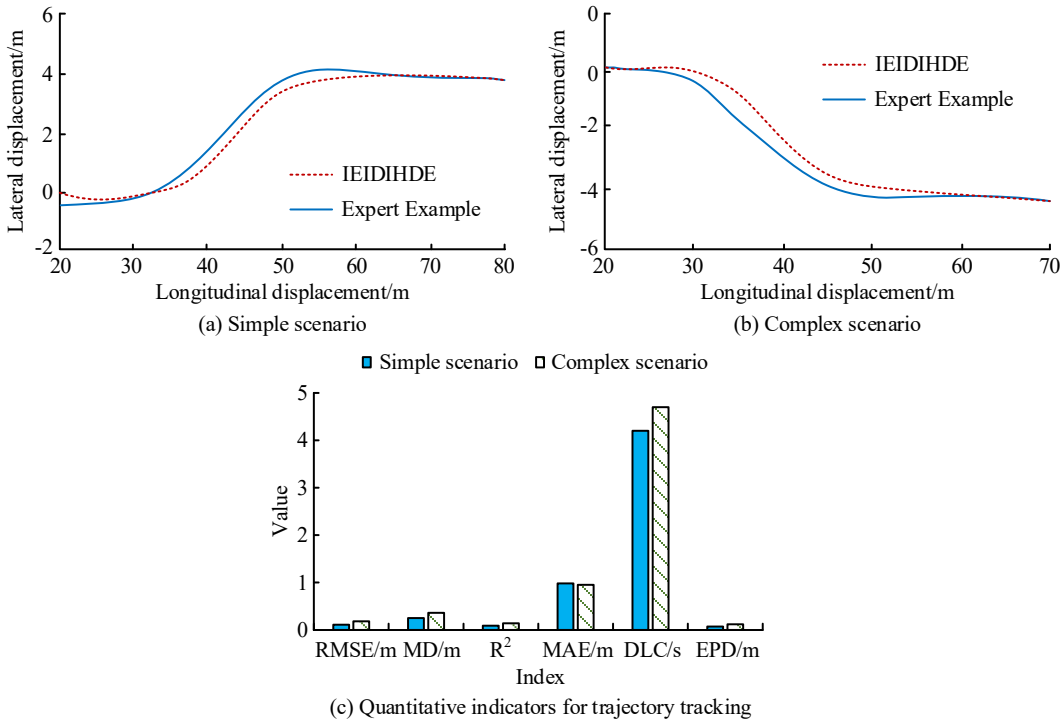


Fig. 7. Quantitative indicators of driving trajectory and trajectory tracking in different testing scenarios

Note: RMSE and MAE represent root mean square error and mean absolute error, MD represents maximum deviation, DLC and EPD represent lane changing duration and endpoint deviation, respectively.

The experiment set up multiple randomly controlled autonomous vehicles. After the vehicles started, all control instructions were output from the end-to-end network, and the perceptual data were obtained from the simulation platform. The scenarios were divided into two groups: simple (a single other vehicle in front) and complex (multiple other vehicles around). The driving trajectories and quantified trajectory tracking metrics for different test scenarios are shown in Figure 7. As shown in Figure 7 (a), in the simple scenario, the overall trend of the lane-changing trajectories generated by the IEIDIHDE model was highly consistent with that of the expert

trajectories. Both started to shift towards the target lane at about 30m of longitudinal displacement, completed the lane-changing at about 55m and tended to stabilize. The maximum lateral displacements of the two were approximately 3.9m and 4.1m respectively, with deviations less than 0.2m, and the lane-changing process was smooth without abrupt changes. The model could accurately reproduce the expert's lane-changing behavior in scenarios without complex disturbances. As shown in Figure 7 (b), in complex scenarios, despite the presence of multiple vehicle interferences around, the IEIDIHDE model could still generate lane change paths that are highly consistent with the expert trajectories. The trajectories-initiated lane changes at a longitudinal displacement of approximately 28m and completed around 50m. The maximum lateral deviation between the model trajectories and the expert trajectories was approximately 0.3m, with only a slight lag in the middle of the lane change. Overall, the timing, speed and lateral displacement of the lane change are close to the expert level. As shown in Figure 7 (c), the error indicators in complex scenarios were higher than those in simple scenarios. The maximum lateral RMSE of the IEIDIHDE model was only 0.18 m, the maximum deviation was 0.36 m, the R^2 still reached 0.95, the increase in lane change time was small, and the trajectory tracking accuracy remained stable. The above results indicate that the IEIDIHDE model still has strong track-following ability and generalization in dynamic interaction scenarios, and can maintain safe and reasonable lane-changing behavior under different working conditions.

Heading angles in different test scenarios are shown in Figure 8.

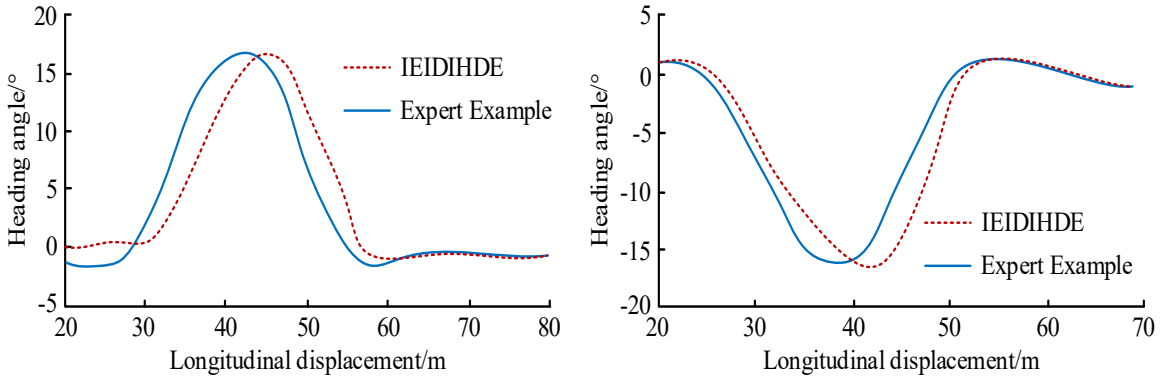


Fig. 8. Heading angles under different testing scenarios

As shown in Figure 8 (a), the trend of the heading angle variation of the IEIDIHDE model in the simple scenario was highly consistent with the expert trajectory. Both started to rise at about 30m of longitudinal displacement, reached the peak in the 40-45m range, the model peak was about 16.5°, the expert peak was about 17°, with a deviation of less than 0.5°, then fell synchronously and stabilized after 60m. The course angle changes were smooth and unaltered throughout the course, which was highly consistent with the steering rhythm of the expert driver. As shown in Figure 8 (b), the model 's heading angle still maintained a high degree of consistency with the expert' s trajectory in complex scenarios. Both rapidly dropped to negative peaks in the 20-40m range, with the model 's lowest value at about -16° and the expert' s lowest value at about -16.5°, with a deviation of about 0.5°, and then recovered synchronously and returned to positive in the 50-60m range. The overall turning timing and variation were close to the expert level. The results suggest that the model can still maintain reasonable and smooth steering control in dynamic interaction scenarios and effectively reproduce the course angle variation pattern of expert driving.

The steering wheel angles in different test scenarios are shown in Figure 9. In Figure 9 (a), the changes in the steering wheel angle of the IEIDIHDE model in simple scenarios were highly synchronized with the expert trajectory, both starting to rise at about 30m of longitudinal displacement, with peaks of 0.085 and 0.09 respectively, and deviations of less than 0.005. Then, it dropped synchronously and reached a trough of around -0.095 at 50m. The overall change was smooth without abrupt changes and highly consistent with the expert's steering rhythm. As shown in Figure 9 (b), the model could still reproduce the expert's steering pattern well in complex

scenarios, both reaching a trough of -0.085 at 30m and a peak of about 0.09 at 50m. The turning angle deviation between the model and the expert was controlled within 0.01, and the steering timing and amplitude were close to the expert's level. The results suggest that the model can still maintain stable and reasonable steering control in dynamic interaction scenarios.

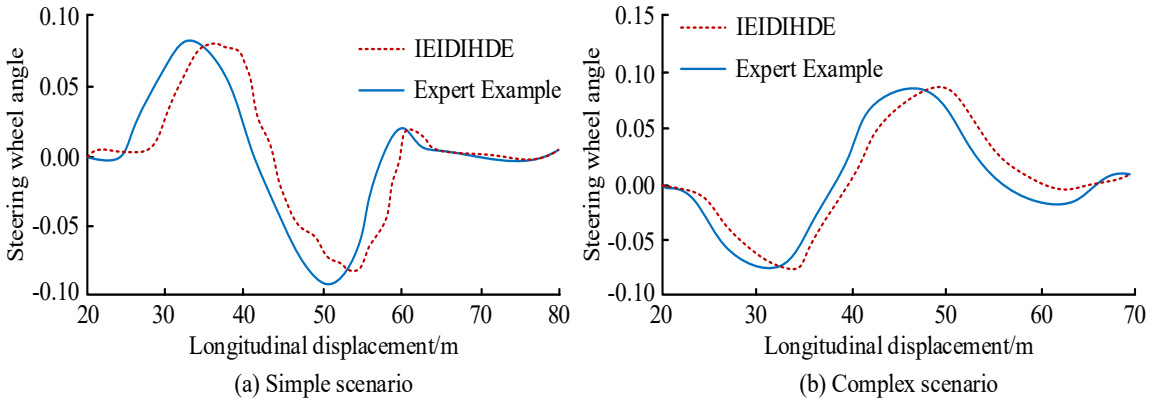


Fig. 9. Steering wheel angle under different test scenarios

Lateral acceleration in different test scenarios is shown in Figure 10

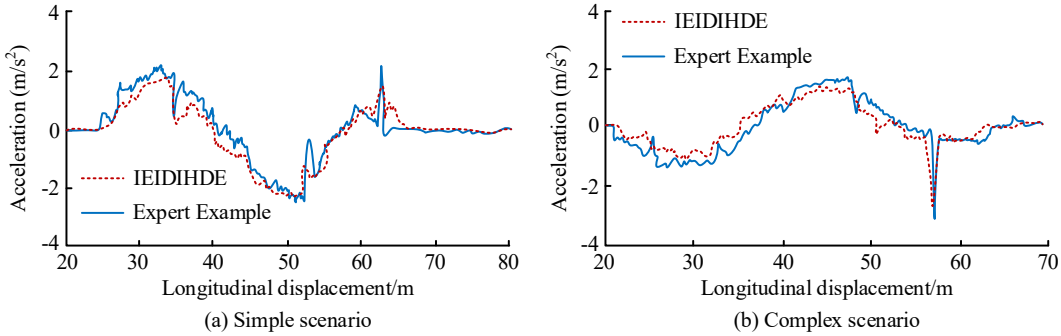


Fig. 10. Lateral acceleration under different testing scenarios

As shown in Figure 10 (a), the lateral acceleration trend of the IEIDIHDE model in the simple scenario was highly consistent with the expert trajectory. Both peaks occurred at a longitudinal displacement of approximately 30-35m, the model peak was about 1.8m/s^2, the expert peak was about 2.1m/s^2, and the deviation was controlled within 0.3m/s^2. Then, it dropped synchronously and reached a trough of about -2.3m/s^2 at 50-55m.

Table 1. Test results of the model on Town03 urban map

Evaluation index	Value
Task success rate	84 %
Collision rate	9 %
Steering prediction accuracy	82.3 %
Speed prediction accuracy	83.5 %
Driving Score	72.6
Route Completion	86.2%
Infraction Penalty	118

The overall change was smooth and there was no sharp fluctuation, indicating that the model can stably reproduce the lateral acceleration characteristics during the expert lane change process. In Figure 10 (b), the model still maintained an acceleration variation pattern similar to that of the expert trajectory in complex scenarios. Both slowly rose to a peak of approximately 1.6m/s^2 in the 30-45m range. The deviation between the model and the expert was less than 0.2m/s^2, with only slight differences in the braking stage at the end of the lane change, and no abnormal fluctuations

exceeding ±2m/s^2 occur overall. The results suggest that the model can still maintain ride stability in complex interaction scenarios. To further evaluate the performance of the model under complex urban conditions, this paper conducted supplementary validation on the CARLA Town03 city map. Town03 includes multiple intersections, traffic signals, pedestrians, and narrow streets and alleys, which are significantly different from the Town4 highway loop scene. The experimental statistical results are shown in Table 1. From the results in Table 1, it can be seen that in the complex urban scene of Town03, the performance of the model is somewhat reduced compared to the high-speed loop scene of Town4 due to complex interactions such as intersection passing, pedestrian crossing, and signal start stop. However, it can still complete most driving tasks. The results indicate that the model has certain adaptability to urban intersection scenarios; At the same time, it also reflects that the model still has shortcomings when facing dense pedestrian and multi vehicle interactions, and further optimization is needed for complex interaction scenarios in the future. To verify the robustness of the model to harsh environments, this paper further conducted multiple sets of harsh environment simulation tests under the Town03 map, setting four working conditions: rainy, foggy, nighttime, and nighttime superimposed image sensor Gaussian noise. The test results are shown in Table 2.

Table 2. Model test results under different adverse environmental conditions

Evaluation index	Rainy	Foggy	Night	Night with sensor noise
Task success rate	78 %	71 %	75 %	64 %
Collision rate	12 %	16 %	14 %	21 %
Steering prediction accuracy	77.1 %	73.6 %	75.4 %	68.2 %
Speed prediction accuracy	78.3 %	74.2 %	76.0 %	69.5 %

From Table 2, it can be seen that compared to ideal daytime conditions, rain, fog, and nighttime environments will reduce the quality of image features, and the overall performance of the model will decrease to varying degrees; When sensor noise is added at night, the image information is severely damaged, and the success rate of the task is further reduced. The results indicate that the model has a certain ability to resist harsh environmental interference, but its robustness is still insufficient in the face of strong noise and extremely low visibility scenes. In the future, the model's anti-interference ability can be further improved through multi environment sample expansion and noise enhancement training.

4.2 Compare the Analysis Results

To assess the performance of IEIDIHDE, it was compared with end-to-end autonomous driving decision-making models based on Improved Human Feedback Reinforcement Learning (IHFRL), Bird's-Eye View Proximal Policy Optimization (BEV-PPO), Recyclable Proximal Policy Optimization (RPPO), and 3D Window Self-Attention (3DWSA). The IHFRL, BEV-PPO, RPPO, and 3DWSA compared are all publicly available decision-based baseline algorithms in autonomous driving simulation environments. To ensure fair comparison, the original network infrastructure is retained when reproducing each model, and the backbone network and output header structure are implemented according to the corresponding literature configuration; All benchmark models are trained and tested in the same hardware environment using the same dataset partitioning, input data format, optimizer, learning rate, and iterative training epochs as the model in this article. It should be noted that CILRS, TransFuser, TCP, and InterFuser are classic and cutting-edge models in the field of end-to-end autonomous driving. TransFuser and InterFuser rely on multi view images and LiDAR inputs, while CILRS and TCP additionally rely on navigation path instructions, which differs from the input conditions of only monocular sequence images and vehicle historical states used in this paper. Direct comparison is difficult to ensure fairness. This article currently selects IHFRL, BEV-PPO, RPPO, and 3DWSA as baselines, all of which are decision methods that only use image observation. The accuracy and real-time performance of the different methods are presented in Table 3.

As shown in Table 3, different autonomous driving methods performed differently in terms of accuracy and real-time performance. The IEIDIHDE method performed the best, with steering and speed accursions of 88.7% and 89.1% respectively, a score of 0.964, and an FPS of 420, leading

overall performance. 3DWSA followed with 78.4% and 78.7% steering and speed accuracy, 0.794 score and 437 FPS. The accuracy and real-time performance of BEV-PPO, IHFRL, and RPPO were relatively balanced, with RPPO having the lowest accuracy and FPS. To further analyze the inference time of the model, the inference delay of each module was decomposed: the single inference delay of the encoder module, Conv-LSTM module, self-attention memory module SAMM, and one-dimensional Conv-LSTM historical state sub network were 1.2 ms, 0.7 ms, 0.3 ms, and 0.2 ms, respectively. This frame rate was obtained under the conditions of high computing power graphics card, low resolution input, and single sample. Due to the influence of hardware, input size, and batch size, the inference frame rate will significantly decrease after hardware degradation. Overall, IEIDIHDE shows a significant advantage in both accuracy and real-time performance and is the best method in terms of overall performance.

Table 3. Accuracy and real time performance of different methods

Model	Accuracy/%		Score	FPS	Parameter count/M	Floating point computational load/G-FLOPs
	Steering	Speed				
IHFRL	73.5	74.7	0.743	458	9.3	2.11
BEV-PPO	76.8	77.2	0.761	462	15.8	3.47
RPPO	71.6	72.9	0.735	419	8.1	1.83
3DWSA	78.4	78.7	0.794	437	14.2	3.12
IEIDIHDE	88.7	89.1	0.964	420	12.6	3.84

Note: FPS stands for frames per second

The simulation results of different methods are denoted in Table 4.

Table 4. Simulation experiment results of different methods

Model	Experimental scenario	Success rate/%	Collision rate/%	Optimal lane change probability/%	Average offset distance/m
IHFRL	Simple	80	16	84	0.48
	Complex	68	20	80	0.57
BEV-PPO	Simple	76	20	76	0.52
	Complex	60	22	72	0.61
RPPO	Simple	72	28	72	0.55
	Complex	56	32	66	0.66
3DWSA	Simple	84	12	88	0.45
	Complex	76	16	82	0.51
IEIDIHDE	Simple	100	0	100	0.20
	Complex	94	4	96	0.32

As shown in Table 3, IEIDIHDE performed best in both simple and complex scenarios, with a 100% success rate and 0% collision rate in simple scenarios, a 94% success rate and only 4% collision rate in complex scenarios, an optimal lane change probability of 96%, and an average offset distance of only 0.32m, significantly superior to other methods. 3DWSA followed with 76% success rate and 16% collision rate in complex scenarios. RPPO performed the worst, with a complex scene success rate of 56% and a collision rate of 32%. Overall, IEIDIHDE has a clear advantage in terms of safety and stability across different scenarios. To verify the effectiveness of the four core innovative modules in this article, ablation experiments were conducted by sequentially removing the improved residual unit, vehicle historical state sub network, SAMM self-attention memory module, and future sequence prediction auxiliary task. The results of the ablation experiments are shown in Table 5 on the Town03 dataset.

According to Table 5, after removing any of the above modules, the success rate of the model task, driving score, and prediction accuracy all decreased to varying degrees, while the collision rate increased. The removal of the SAMM module (c) resulted in the greatest performance degradation, indicating that the self-attention memory module plays a crucial role in long-term spatiotemporal

modeling; Removing the historical state sub network (b) and replacing the ordinary residual (a) also results in significant performance degradation, demonstrating the gain of feature extraction from historical state information and lightweight improved residual units; The future sequence prediction auxiliary task (d) plays an auxiliary constraint role, which can further improve the accuracy of model decision-making. All four innovative modules contribute positively to the performance of the final model.

Table 5. Results of module ablation experiment

Model configuration	Task success rate	Collision rate	Driving Score	Steering prediction accuracy
Full IEIDIHDE (all modules enabled)	84.0 %	9.0 %	72.6	82.3 %
w/o a: replace improved residual unit with vanilla residual unit	78.2 %	12.4 %	65.1	77.5 %
w/o b: remove 1D-CNN-LSTM historical-state sub-network	76.5 %	13.1 %	63.3	76.1 %
w/o c: remove SAMM self-attention memory module	74.1 %	14.7 %	60.8	73.8 %
w/o d: remove future-sequence prediction auxiliary task	79.4 %	11.6 %	66.7	78.6 %

5. Conclusions

To solve the issues of error accumulation and insufficient adaptability in the traditional modular intelligent driving decision-making architecture, as well as the weak modeling of long-term temporal dependencies and the easy loss of spatial information in the conventional end-to-end methods, a Conv-LSTM intelligent driving end-to-end decision-making algorithm integrating human driving behavior and SAMM - IEIDIHDE - is proposed. Simulation findings denote that in simple scenarios, the maximum lateral deviation between the model lane-changing trajectory and the expert trajectory is less than 0.2m, the peak deviation of the heading angle is less than 0.5°, the steering wheel angle deviation is less than 0.005, and the peak deviation of the lateral acceleration is controlled within 0.3m/s^2. In complex scenarios, the maximum lateral deviation of the trajectory is approximately 0.3m, and the deviations of the heading angle and steering wheel angle are controlled within a reasonable range, with no abnormal acceleration fluctuations. The findings of the comparative experiments show that the model has a 100% success rate in simple scenarios, a 0% collision rate, a 94% success rate in complex scenarios, a 4% collision rate, an 96% optimal lane-changing probability, and an average offset distance of 0.32m. The steering and speed prediction accuracy is 88.7% and 89.1% respectively, the overall score is 0.96, the inference frame rate is 420FPS, and the overall performance is superior to the comparison method.

The results above demonstrate the effectiveness and superiority of IEIDIHDE in end-to-end decision-making for intelligent driving. However, while the SAMM enhances long-sequence modeling capabilities, it increases computational overhead, making deployment on embedded in-vehicle platforms challenging. Therefore, future work will consider applying pruning, quantization, and knowledge distillation to the SAMM and Conv-LSTM to reduce parameter counts and computational demands, thereby adapting the system for in-vehicle edge computing platforms. At the level of model deployment, feasibility studies will be conducted for embedded hardware in vehicles. The specific implementation steps are as follows: first, complete the structured pruning of the model, remove redundant convolutions and fully connected channels, and reduce the number of parameters; Secondly, implement 8-bit quantization on the pruned model to compress its storage size and simplify calculations; Thirdly, export the pruned quantized model into a deployment format supported by the embedded inference framework; Fourth, complete inference deployment on embedded hardware in vehicles, test inference latency and memory usage, and evaluate performance degradation caused by quantitative pruning operations. Balance and optimize between model accuracy and real-time performance to verify the feasibility of model deployment in real vehicles.

References

[1] Lan Y, Xu X, Liu J, Zhang X, Lu Y, Cheng L. A Survey on Reinforcement Learning for Optimal Decision-Making and Control of Intelligent Vehicles. CAAI Transactions on Intelligence Technology. 2025;10(6):1593-1615. https://doi.org/10.1049/cit2.70073

[2] Abdel-Basset M, Mohamed R, Chang V. A Multi-Criteria Decision-Making Framework to Evaluate the Impact of Industry 5.0 Technologies: Case Study, Lessons Learned, Challenges and Future Directions. Information Systems Frontiers. 2025;27(2):791-821. https://doi.org/10.1007/s10796-024-10472-3

[3] Viswanadhapalli V. Integrating AI and RPA in Pega for Intelligent Process Automation: A Comparative Study. International Journal of Scientific Research and Management (IJSRM). 2024;12(6):1317-1333. https://doi.org/10.18535/ijsrm/v12i06.ec11

[4] Weng S, Huang Z, Lv Y. RETRACTED: Probability numbers for multi-attribute decision-making. Journal of Intelligent & Fuzzy Systems. 2024;46(3):6109-6132. https://doi.org/10.3233/JIFS-223565

[5] Silva ESD, Pedrini H, Santos ALD. Applying Graph Neural Networks to Support Decision Making on Collective Intelligent Transportation Systems. IEEE Transactions on Network and Service Management. 2023;20(4):4085-4096. https://doi.org/10.1109/TNSM.2023.3257993

[6] Xu C, Zhao H, Lu X, Sun K, Gao B, Chen H. A Deep Reinforcement Learning Method for Autonomous Driving Integrating Multi-Modal Fusion. IEEE Transactions on Intelligent Transportation Systems. 2025;26(8):11850-11863. https://doi.org/10.1109/TITS.2025.3573781

[7] Chen Y, Yan G, Yu H, Xi J. Intention-Guided Heuristic Partially Observable Monte Carlo Planning for Off-Ramp Decision-Making of Autonomous Vehicles. IEEE Transactions on Intelligent Transportation Systems. 2025;26(7):10834-10849. https://doi.org/10.1109/TITS.2025.3547906

[8] Biradar S, Saumya S, Kavatagi S. Safeguarding the Integrity of Online Social Networks (OSN): Leveraging the Efficacy of Conv-LSTM-Based Siamese Network to Predict Hate Speech in Low Resource Hindi-English Code-Mixed Text. IEEE Access. 2025;13:141598-141608. https://doi.org/10.1109/ACCESS.2025.3597144

[9] Murali E, Sheela ACS, Paul MA, Muthu V, Felix AY. Learning normal patterns via conv-LSTM for video anomaly detection using likelihood statistical texture feature representation in surveillance videos. International Journal of System Assurance Engineering and Management. 2025;16(12):3851-3862. https://doi.org/10.1007/s13198-025-02892-4

[10] Muiruri SM, Okong'O M, Mwathi D. Enhancing Public Safety Through Advanced Video Analysis: A Conv-LSTM-SVM Model for Violence Detection in Surveillance Footage. East African Journal of Information Technology. 2024;7(1):202-214. https://doi.org/10.37284/eajit.7.1.2117

[11] Shelke SS, Degadwala SD. Multi-Class Recognition of Soybean Leaf Diseases using a Conv-LSTM Model. International Journal of Scientific Research in Computer Science, Engineering and Information Technology. 2024;10(2):249-257. https://doi.org/10.32628/cseit2410217

[12] Dar AA, Hirani T, Arora V, Kakkar S. Predictive Analytics for Personalized Debt Management: Leveraging Machine Learning to Provide Actionable Financial Advice. FinTech and Sustainable Innovation. 2025;1:A4. https://doi.org/10.47852/bonviewFSI52025591

[13] Chen H, Xue S, Zheng X. Coal mine microseismic identification and first-arrival picking based on Conv-LSTM-Unet. Acta Geophysica. 2023;71(1):161-173. https://doi.org/10.1007/s11600-022-00898-1

[14] Kurchaniya D, Kumar S. D-SCAN: Dual Stream Spatiotemporal Channel-Wise Attention Network With Point-Wise ConvBi-LSTM for Activity Recognition. IEEE Transactions on Consumer Electronics. 2024;70(4):7244-7251. https://doi.org/10.1109/TCE.2024.3425610
[15] Chen C, Zhang R. Using upsampling CONV-LSTM with metadata embedding for respiratory sound classification. Theoretical and Natural Science. 2023;28(1):78-84. https://doi.org/10.54254/2753-8818/28/20230401
[16] Sun Y, Di K, Deng Y, Hu J. Demodulation of Different Quantities of Overlapping Spectra in FBG Sensors Based on Combined Conv-TasNet and LSTM. IEEE Sensors Journal. 2024;24(18):28860-28866. https://doi.org/10.1109/JSEN.2024.3434462
[17] Ma S, Zhang T, Bai P. TCLN: A Transformer-based Conv-LSTM network for multivariate time series forecasting. Applied Intelligence. 2023;53(23):28401-28417. https://doi.org/10.1007/s10489-023-04980-z
[18] Kanna PR, Santhi P. An Enhanced Hybrid Intrusion Detection Using Mapreduce-Optimized Black Widow Convolutional LSTM Neural Networks. Wireless Personal Communications. 2024;138(4):2407-2445. https://doi.org/10.1007/s11277-024-11607-0
[19] Bhattad R, Patel V, Patel U. Deep learning fusion for dynamic cropland monitoring: integrating MUNet with Siamese networks and convolutional LSTM2D. Modeling Earth Systems and Environment. 2024;10(4):5381-5398. https://doi.org/10.1007/s40808-024-02068-2
[20] Cikambasi CL, Muriira LM, Murungi RM. Deep Learning Network Intrusion Detection with the Conv1d-Lstm Model: Integrating CNN and LSTM For Superior Performance. International Journal of Professional Practice. 2024;12(4):41-49. https://doi.org/10.71274/ijpp.v12i4.475