Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Research on Trajectory Prediction and Risk Control for Distributed-Drive Electric Vehicles Based on a Spatiotemporal Transformer and an Improved DRA Algorithm

Hu Wang

wanghu20260822@163.com

Hunan Institute of Traffic Engineering

Yuepeng Li

Hunan Institute of Traffic Engineering

Qizheng Yang

Hunan Institute of Traffic Engineering

Research Article

Keywords: distributed-drive electric vehicles, Trajectory prediction, Improved divine religions algorithm, spatiotemporal transformer, Risk control, spatiotemporal features

Posted Date: October 29th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11092749/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Research on Trajectory Prediction and Risk Control for Distributed-Drive Electric Vehicles Based on a Spatiotemporal Transformer and an Improved DRA Algorithm

Hu Wang, Yuepeng Li, Qizheng Yang

Hunan Institute of Traffic Engineering, Hengyang, 421009, China;
Email:wanghu20260822@163.com
Corresponding author: " Hu Wang "
Email: wanghu20260822@163.com

Abstract: Considering the strong nonlinearity and dynamic coupling characteristics of distributed-drive electric vehicle trajectories, as well as the slow convergence and susceptibility to local optima of conventional optimization algorithms when tuning deep temporal models, this paper proposes a trajectory prediction and risk identification method based on an improved Divine Religions Algorithm (IDRA)-optimized spatiotemporal Transformer. First, an adaptive computation-saving mechanism and a dynamic community interaction mechanism are introduced into DRA to reduce redundant fitness evaluations, maintain population diversity, and improve convergence performance. Second, a spatiotemporal Transformer integrating temporal self-attention, spatial interaction attention, and spatiotemporal feature fusion is constructed to capture vehicle motion characteristics and multi-vehicle interaction dependencies. IDRA is then employed to automatically optimize key Transformer hyperparameters. Experimental results on eight benchmark functions show that IDRA achieves approximately 1.5-3 times faster convergence and improves optimization accuracy by 1-15 orders of magnitude compared with the comparison algorithms. On the self-constructed Multi-Vehicle Interaction Trajectory Dataset for Distributed-Drive Electric Vehicles, the proposed method achieves an MAE of 0.03 m, RMSE of 0.04 m, ADE of 0.02 m, and FDE of 0.05 m, outperforming GAT, GNN, and CNN-LSTM across all evaluated trajectory prediction metrics. These results demonstrate that the proposed method can effectively improve trajectory prediction accuracy under complex multi-vehicle interaction scenarios and provide reliable trajectory information for early risk identification and subsequent active-safety decision-making.

Keywords: distributed-drive electric vehicles; Trajectory prediction; Improved divine religions algorithm; spatiotemporal transformer; Risk control; spatiotemporal features.

1. Introduction

Against the backdrop of the accelerating integration of new energy vehicles and autonomous driving technologies, vehicle operational safety is gradually shifting from traditional passive protection toward an intelligent control paradigm of “prediction–warning–active intervention.” As a core technological foundation for understanding future vehicle motion intentions and potential risk evolution, trajectory prediction directly affects collision-avoidance decision-making, multi-vehicle cooperative control, and the overall safety of transportation systems [1-2]. For distributed-drive electric vehicles, the independently driven wheel-end architecture significantly improves the degrees of freedom in vehicle dynamic regulation and the precision of attitude control, but also makes longitudinal–lateral dynamic coupling more complex. Under complex road conditions, multi-vehicle interactions, and sudden behavioral disturbances, vehicle trajectories

exhibit strong nonlinearity, pronounced spatiotemporal correlations, and rapid state transitions. Without accurate characterization of such highly dynamic evolution, reliable risk prediction and active control are difficult to achieve [3-4].

For vehicle trajectory prediction, research has gradually evolved from physics-based models to data-driven approaches. Early studies mainly constructed prediction frameworks based on kinematic equations and state-estimation methods. For example, Ramadevi N et al. [5] employed an extended Kalman filter for continuous target trajectory estimation, achieving good stability in regular-motion scenarios but showing limited adaptability to strongly nonlinear behaviors. Subsequently, Hamedi, H. et al. [6] introduced support vector regression to realize nonlinear mapping of vehicle paths, improving prediction flexibility under complex road conditions, although long-term dependency structures remained difficult to capture. With the development of deep learning, recurrent neural networks have gradually become mainstream tools for trajectory modeling. Among them, the Long Short-Term Memory Network (LSTM), owing to its gated memory mechanism and advantages in temporal dependency modeling, has been widely applied to vehicle trajectory prediction. Qiao, S. et al. [7] developed a socially aware LSTM model to characterize multi-vehicle interactions, while Min, H. et al. [8] used LSTM to model trajectory evolution under complex traffic flows. Both approaches achieved notable improvements in prediction accuracy. However, such methods primarily focus on temporal dynamics and remain limited in representing spatial structural relationships and long-range dependencies.

To overcome the limitations of temporal models in spatial correlation modeling, researchers have further introduced graph-based learning and attention mechanisms to enhance trajectory feature representation. Wang, R. et al. [9] constructed a vehicle-interaction topology model based on a Graph Neural Network (GNN), effectively improving prediction robustness in multi-vehicle scenarios. Teng, F. et al. [10] dynamically assigned influence weights to neighboring vehicles through an attention mechanism, enhancing model sensitivity to changes in high-risk behaviors. In recent years, the Transformer architecture has increasingly been introduced into trajectory prediction because of its advantages in modeling long-range dependencies through self-attention. Zhong, Y. et al. [11] proposed a spatiotemporal Transformer network that simultaneously captures temporal evolution and spatial interaction features, significantly outperforming conventional recurrent network models in complex traffic environments. However, the Transformer is highly parameterized, and its performance is extremely sensitive to hyperparameter settings such as the number of attention heads, network depth, and learning rate. As scenario complexity increases, training instability and fluctuations in generalization performance can readily occur [12].

To improve the adaptability of deep trajectory prediction models, some studies have begun to employ intelligent optimization algorithms for automatic hyperparameter tuning. Jlifi, B. et al. [13] introduced a genetic algorithm into the parameter-search process of deep models to obtain improved network configurations. Xie, H. et al. [14] used the Grey Wolf Optimizer to optimize temporal prediction models, demonstrating strong global search capability in complex dynamic data modeling. However, these methods still commonly suffer from slow convergence, limited search efficiency, and susceptibility to local optima in high-dimensional parameter spaces. These limitations become even more pronounced when tuning structurally complex spatiotemporal Transformer models, thereby constraining further improvements in trajectory prediction accuracy and the real-time performance of risk identification [15].

Based on the above research, trajectory prediction and risk control for distributed-drive

electric vehicles still face two major bottlenecks. On the one hand, although high-performance spatiotemporal deep models possess strong dynamic modeling capabilities, their effectiveness depends heavily on hyperparameter configurations, while efficient and stable automatic optimization mechanisms remain lacking. On the other hand, conventional intelligent optimization algorithms exhibit insufficient search efficiency and robustness in complex high-dimensional search spaces, making it difficult to satisfy the requirements of real-time prediction and safety control. Accordingly, this study proposes a trajectory prediction and risk control method based on a spatiotemporal Transformer optimized by an improved Divine Religions Algorithm. An adaptive computational-saving mechanism and a dynamic community interaction mechanism are developed to improve search efficiency and optimization stability, while spatiotemporal dependency modeling is integrated to achieve high-accuracy vehicle trajectory prediction and early risk identification in complex traffic scenarios. The proposed method provides an efficient and reliable intelligent decision-support approach for the active safety control of distributed-drive electric vehicles.

2. Methods

2.1 DRA algorithm

The Divine Religion Algorithm (DRA) is a novel metaheuristic optimization algorithm inspired by religious societies. The algorithm draws on a politico-religious framework and addresses optimization problems by modeling leaders who compete for religious prestige and followers who seek to strengthen their beliefs. Just as leaders seek devout followers whose beliefs align with their own religious doctrines, the algorithm aims to identify the optimal solution under the guidance of an objective function. The effectiveness of each follower is evaluated according to the strength of their political and religious views. Similar to other bio-inspired metaheuristic algorithms, the population evolution and iterative optimization rules of DRA are implemented based on population initialization. The population initialization operation is expressed in Eq. (1):

$$C_j = \begin{bmatrix} c_{1j}^1 & c_{1j}^2 & \cdots & c_{1j}^d \\ c_{2j}^1 & c_{2j}^2 & \cdots & c_{2j}^d \\ \vdots & \vdots & \ddots & \vdots \\ c_{nj}^1 & c_{nj}^2 & \cdots & c_{nj}^d \end{bmatrix} \tag{1}$$

The DRA algorithm identifies the follower with the best belief value as the most prominent follower $w_{\hat{i}}$. To this end, it searches for the follower with the highest fitness across all communities. The specific operation is expressed in Eq. (2):

$$\hat{i}\,\hat{j} = \arg\max F \tag{2}$$

where $\hat{j}$ denotes the community in which the most prominent follower $w_{\hat{i}}$ is located.

The belief profile of the follower is expressed in Eq. (3):

$$\hat{c}_{\hat{i}\,\hat{j}} = \left(\hat{c}_{\hat{i}\,\hat{j}}^1, \hat{c}_{\hat{i}\,\hat{j}}^2, \ldots \hat{c}_{\hat{i}\,\hat{j}}^d\right) \tag{3}$$

The selection of the missionary is similar to the above process, except that the selection is local rather than global. The missionary corresponding to community m_j is the individual with

the highest fitness within that community, and its profile is denoted by $C_{\hat{j}}$.

Before incorporating the new individual, the algorithm adjusts its belief value by probabilistically applying either the "selection" operator or the "miracle" operator. If the generated random number falls within the belief selection probability, the selection operator is executed; otherwise, another random number is generated to determine whether the miracle operator is applied.

As commonly observed in real-world human communities, attracting new followers requires a minimum level of belief. However, applicants exhibiting extraordinary beliefs in one or more characteristics may occasionally attract particular attention. To enable the model to capture this behavior, one characteristic of the most prominent follower $w_{\hat{i}}$ within the same community m_j is randomly selected. Specifically, a uniformly distributed random integer is selected from the interval [1, *d*]. This value is denoted by d in this study, and the specific calculation is given in Eq. (4):

$$d = \lfloor Uniform(1, d) \rfloor \tag{4}$$

The *d*-th attribute of the newly introduced $followerw_i$ must be replaced with the corresponding attribute of the most prominent $followerw_{\hat{i}i}$. The specific operation is expressed in Eq. (5):

$$c_{ij}^{d'} = c_{\hat{i}\hat{j}}^{\prime d'} \times \left(rand - \cos\left(rand \right) \right) \tag{5}$$

Religious leaders can attract new followers by demonstrating extraordinary abilities and performing miracles. These miracles may cause new followers with different beliefs to join the community.

This process improves the religious beliefs of current followers and enables them to achieve better performance and status within the religious society. To perform the miracle operation, a random number is generated within the interval [0,1]. The generated random number is then compared with the miracle probability threshold, which is also defined within [0,1]. Based on this comparison, the belief value corresponding to one of the followers is calculated according to the following formulation, or a new follower with a randomly generated religious belief value is added to the community.

During the missionary operator stage, a religious missionary attempts to bring other individuals to the same belief level. This stage represents an exploitation process that can effectively approach the global optimum. Based on the guidance provided by each religious missionary, followers attempt to reach a similar belief state by modifying their characteristics. To this end, a random number is generated within [0,1] and compared with the MP threshold. The follower characteristics are then updated by assigning values through the miracle and conversion operators according to the following formulations. The specific calculations are given in Eqs. (6)-(8):

$$MP = \left(1 \times rand\right) \times \left(1 - \left(\frac{t}{Max_Iter \times 2} \right) \right) \times rand \tag{6}$$

$$rand \le MP \begin{cases} rand \le 0.5 \;\; c_{i'} = c_{i'} \times \cos\left(\dfrac{\pi}{2}\right) \times \\ \left[rand - \cos(rand)\right] \\ rand > 0.5 \;\; c_{i'} = c_{i'} + rand \times \\ \left[c_{i'} - round\left(1^{rand} \times c_{i'}\right)\right] \end{cases} \tag{7}$$

Miracles Operator

$$and > MP \begin{cases} rand > (1-MP) \;\; c_{i'} = (c_{i'} \times 0.01) + \\ [\bar{c}_{i'} \times (1-MP) + (1-\bar{c}_{i'}) - \\ (rand - 4 \times \sin(\pi \times rand)] \\ rand \le (1-MP) \;\; c_{i'} = (rand - 2 \times rand - \\ c_{i'}) \times (2 \times rand - (1-MP) \times randn) \end{cases} \tag{8}$$

Pr*oselytism Operator*

After the above operations on the belief of the new follower are completed, its fitness is recalculated and compared with that of the worst existing follower in the religion. If the new follower performs better, it replaces the worst follower in the original population. The specific operation is expressed in Eq. (9):

$$\begin{cases} follower(i) == penalty \;\; Add \;\; followers \; to \; the \\ \;\; forbidden \; list \\ else \qquad Continue \end{cases} \tag{9}$$

To create a competitive atmosphere among followers, some followers are identified either randomly or according to their performance and are then rewarded or penalized. This process can improve the performance level of the followers. To this end, a random number within the interval [0,1] is first generated to execute the reward or punishment operator, and the generated value is then compared with RP. Subsequently, the value of one corresponding attribute of the selected follower is set based on a uniformly distributed random number, as expressed in Eq. (10):

$$\begin{cases} rand \ge RP \;\; c_{i'} = c_{i'} \times \\ (1-randn)\,\mathrm{Re}\,ward \;\; Operator \\ rand < RP \;\; c_{i'} = c_{i'} \times \\ (1+randn)\,Penalty \;\; Operator \end{cases} \tag{10}$$

As individual beliefs improve, the fitness of a follower may surpass that of the missionary of the corresponding religion. In this case, the "missionary replacement" operator exchanges the roles of the original missionary and the superior follower, thereby ensuring that the leader within the religion always represents the optimal solution. The specific operation is expressed in Eq. (11):

$$\begin{cases} \left(C_{(i'j)} > \breve{C}_{(i'j)}\right) \; swap\left(C_{(i'j)}, \breve{C}_{(\tilde{i}j)}\right), \\ w_{\tilde{i}} \; is \; the \; worst \; follower \\ \left(C_{(i'j)} \le \breve{C}_{(\tilde{i}j)}\right) \; Continue \end{cases} \tag{11}$$

After all religions and individuals have been updated, the algorithm checks whether the stopping criterion has been met, for example, whether the preset maximum number of iterations has been reached or whether the overall fitness no longer changes after the current iteration. If the stopping criterion is not satisfied, the algorithm proceeds to the next iteration, where each religion continues to recruit new followers, perform miracles and selection, apply rewards or penalties, and dynamically update the optimal solution.

The computational flow of the DRA algorithm is shown in Fig. 1:

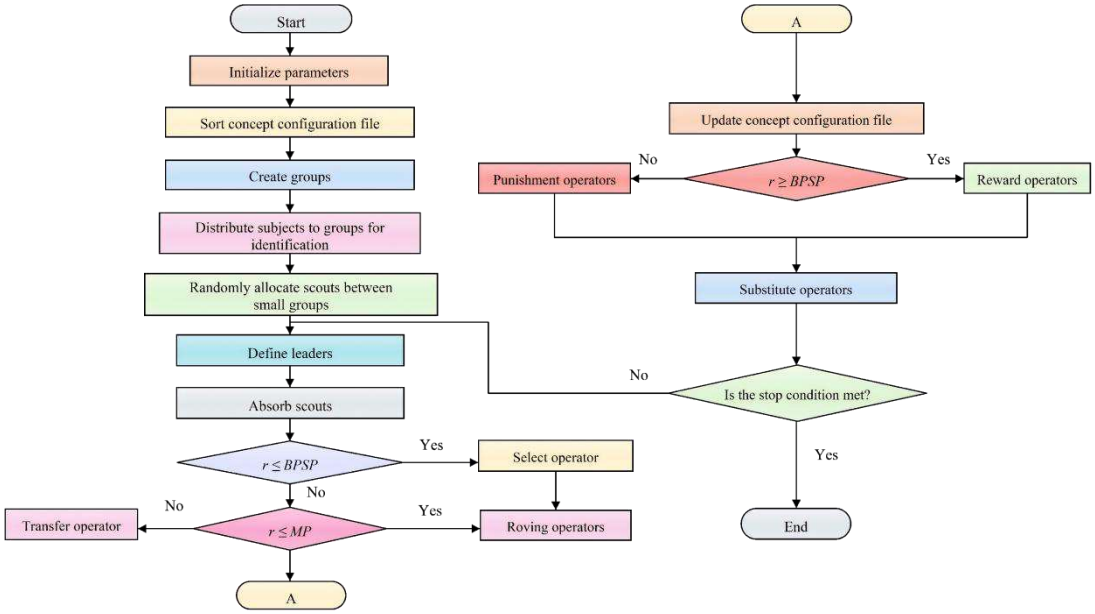


Fig. 1 The calculation flowchart of the DRA algorithm

2.2 Problems of the DRA algorithm

Although the DRA algorithm has a certain degree of global exploration capability, it still suffers from the following two key problems, which limit its effectiveness in complex trajectory prediction and optimization tasks for distributed-drive electric vehicles.

Problem 1: High computational complexity and challenges in real-time performance.

As a population-based metaheuristic algorithm, DRA involves a large number of stochastic operations and fitness evaluations during the iterative optimization process. In each generation, operators such as "selection," "miracle," "reward or punishment," and "missionary replacement" must be applied to each "follower" in the population. Each operation involves random-number generation, comparison, and the numerical updates described in Eqs. (4)–(11), followed by recalculation of the fitness of the newly generated individual. For a population of size N, the computational cost of a single iteration is considerable. When DRA is used to optimize the hyperparameters or training process of a spatiotemporal Transformer model, each fitness evaluation requires one or more forward passes or even training procedures of the Transformer, which is itself computationally intensive. The outer iterative optimization loop introduced by DRA therefore substantially increases the overall computational burden and results in a high time cost for convergence. For electric-vehicle trajectory prediction and risk-control systems that require online or rapid response, such computational efficiency is difficult to satisfy real-time requirements.

Problem 2: The "religious" community structure is prone to premature convergence and local optima.

The DRA algorithm guides the search by dividing the population into "communities" or religions and performing internal "missionary" operations. Although the update mechanisms for the "most prominent follower" and the "missionary" in Eqs. (2) and (11) can accelerate convergence, they also introduce the risk of premature convergence. During the early stage of optimization, if a community happens to discover a locally superior solution, its missionary and

most prominent follower may quickly become concentrated around that region. The subsequent missionary operators in Eqs. (6)-(8) essentially guide other individuals within the community toward exploitation around the current local optimum. Although the miracle operator provides opportunities for random exploration, its activation is restricted by a probability threshold, and the random modification of attribute values may lack directional guidance. When optimizing the multimodal and nonlinear loss function of a trajectory prediction model, this mechanism can lead to premature loss of population diversity, causing an entire religion to become trapped within the basin of attraction of a local optimum and making it difficult to escape and discover a better global region. For the complex trajectory prediction problem of distributed-drive electric vehicles, the solution space usually contains multiple local optima. This structural characteristic of DRA may therefore prevent it from identifying truly optimal trajectory prediction model parameters for subsequent risk-control strategies.

2.3 Improved DRA algorithm

To address the high computational complexity and susceptibility to local optima of the DRA algorithm, two improvement mechanisms are proposed to enhance its convergence speed and global search capability, making it more suitable for complex trajectory prediction and optimization tasks.

Improvement 1 of the DRA algorithm: adaptive computation-saving mechanism.

To reduce the computational burden of the DRA algorithm in each iteration, an adaptive computation-saving mechanism is introduced. Based on the historical fitness variation of each individual, this mechanism dynamically determines whether the complete update and evaluation of that individual should be skipped in the current iteration, thereby reducing unnecessary fitness evaluations. It is particularly suitable for model optimization tasks with high evaluation costs.

The core of this mechanism is the introduction of an adaptive computation-saving factor $\alpha_i^{(t)}$, which controls whether follower i undergoes the complete update and evaluation process at iteration t. The specific calculation is given in Eq. (12):

$$\alpha_i^{(t)} = \begin{cases} 1, & if\ \Delta f_i^{(t-1)} \geq \tau\ or\ t \leq T_{warm} \\ 0, & otherwise \end{cases} \tag{12}$$

where $\alpha_i^{(t)}$ indicates whether individual i undergoes a complete update at iteration t; $\Delta f_i^{(t-1)}$ denotes the absolute fitness change of individual i in the previous iteration, i.e., $\left| f_i^{(t-1)} - f_i^{(t-2)} \right|$; τ denotes the variation threshold used to determine whether the individual remains in an active optimization state; and T_{warm} denotes the number of warm-up iterations to ensure sufficient population exploration during the early optimization stage.

If $\alpha_i^{(t)} = 0$, individual i retains its original belief value in the current iteration and does not participate in operators such as “selection” and “miracle.” Its fitness is not recalculated, and the value from the previous generation is directly retained. This mechanism significantly reduces the computational cost per generation and is particularly suitable for scenarios in which fitness evaluation is computationally expensive.

Improvement 2 of the DRA algorithm: dynamic community interaction mechanism.

To alleviate premature convergence caused by the religious community structure, a dynamic community interaction mechanism is proposed. This mechanism allows information exchange among different communities under certain conditions, thereby enhancing population diversity and promoting global exploration.

At the end of each iteration, two different communities are selected with a certain probability $p_{exchange}$ for individual exchange. The specific steps are as follows:

Step 1: Select interacting communities. Two communities, C_m and C_n, are randomly selected.

Step 2: Select individuals for exchange. One non-missionary follower is selected from each community, denoted as $x_{m,k}$ and $x_{n,l}$, respectively.

Step 3: Perform the exchange. The two individuals exchange their community memberships

while retaining their original belief values. Subsequent updates are then carried out within their new communities.

The exchange probability $p_{exchange}$ is adaptively adjusted according to the number of iterations. The specific calculation is given in Eq. (13):

$$p_{exchange}^{(t)} = p_{\max} - \left(p_{\max} - p_{\min}\right) \times \frac{t}{T_{\max}} \tag{13}$$

where $p_{exchange}^{(t)}$ denotes the community interaction probability at iteration t; $p_{\max}$ and $p_{\min}$ denote the maximum and minimum interaction probabilities, respectively; t denotes the current iteration number; and $T_{\max}$ denotes the maximum number of iterations.

In addition, to prevent the loss of high-quality individuals, missionaries are excluded from the exchange process, and each community is allowed to exchange at most one individual at a time. By introducing cross-community mobility, this mechanism effectively alleviates excessive convergence within individual communities and enhances the algorithm's ability to escape from local optima.

2.4 Transformer model based on spatiotemporal feature fusion

With its global self-attention mechanism, the spatiotemporal Transformer architecture has revolutionized long-range dependency modeling in the field of natural language processing. Its core advantage lies in its ability to directly compute correlation weights between any two elements in a sequence while maintaining a high degree of parallelism. The computational flow of the spatiotemporal Transformer model is as follows:

Assume that there are N distributed-drive electric vehicles in the scenario, with an observation horizon of T_{obs}, a prediction horizon of T_{pred}, and a sampling interval of Δt.

Step 1: Define the vehicle state. The state of vehicle i at time t is expressed in Eqs. (14)–(20):

$$s_i^{(t)} = \left[p_i^{(t)}, v_i^{(t)}, \psi_i^{(t)}, \tau_i^{(t)}, \delta_i^{(t)} \right] \in R^{d_s} \tag{14}$$

$$p_i^{(t)} = \left[x_i^{(t)}, y_i^{(t)} \right] \in R^2 \tag{15}$$

$$v_i^{(t)} = \left[v_{x,t}^{(t)}, v_{y,i}^{(t)} \right] \in R^2 \tag{16}$$

$$\psi_i^{(t)} = \left[\psi_i^{(t)}, \dot{\psi}_i^{(t)} \right] \in R^2 \tag{17}$$

$$\tau_i^{(t)} = \left[\tau_{fl}^{(t)}, \tau_{fr}^{(t)}, \tau_{rl}^{(t)}, \tau_{rr}^{(t)} \right] \in R^4 \tag{18}$$

$$\delta_i^{(t)} \in R \tag{19}$$

$$d_s = 2 + 2 + 2 + 4 + 1 = 11 \tag{20}$$

where $p_i^{(t)}$ denotes the vehicle center-of-mass position; $v_i^{(t)}$ denotes the vehicle velocity vector; $\psi_i^{(t)}$ denotes the yaw angle and yaw rate; $\tau_i^{(t)}$ denotes the four-wheel driving/braking torque; $\delta_i^{(t)}$ denotes the front-wheel steering angle; and d_s denotes the dimension of the state vector.

Step 2: Observe the historical tensors of all vehicles and perform prediction. The historical observation tensor of all vehicles is expressed in Eq. (21):

$$S_{obs} = \left[s_i^{(t)} \right] \in R^{N \times T_{obs} \times d_s}, t \in \left\{1, 2, \ldots, T_{obs}\right\} \tag{21}$$

The future trajectory of the target vehicle is calculated according to Eq. (22):

$$Y = \left[p_{target}^{(t)} \right] \in R^{T_{pred} \times 2}, t \in \left\{ T_{obs} + 1, \ldots, T_{obs} + T_{pred} \right\} \tag{22}$$

To eliminate the influence of the global coordinate system, a target-vehicle-centered relative coordinate system is adopted. The specific calculations are given in Eqs. (23) and (24):

$$p_{i,rel}^{(t)} = R\left(\psi_{target}^{(T_{obs})} \right)^{\top} \left(p_i^{(t)} - p_{target}^{(T_{obs})} \right) \tag{23}$$

$$v_{i,rel}^{(t)} = R\left(\psi_{target}^{(T_{obs})} \right)^{\top} v_i^{(t)} \tag{24}$$

where $R(\psi) = \begin{bmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{bmatrix}$ denotes the rotation matrix, and $\psi_{target}^{(T_{obs})}$ denotes the yaw angle of the target vehicle at the final observation time.

Feature embedding is then performed to map the state of each vehicle into the feature space. The specific operations are expressed in Eqs. (25) and (26):

$$X_i^{(t)} = MLP_{embed}\left(s_{i,rel}^{(t)} \right) \in R^{d_{model}} \tag{25}$$

$$X = \left[X_i^{(t)} \right] \in R^{N \times T_{obs} \times d_{model}} \tag{26}$$

where MLP_{embed} denotes the multilayer perceptron with a dimension of $\left[d_s, 64, d_{model} \right]$; d_{model} denotes the model hidden dimension, which is set to 128.

Step 3: Process the time series of each vehicle i independently. The specific calculation is expressed in Eq. (27):

$$\begin{aligned} Q_i^{temp} &= X_i W_Q^{temp}, K_i^{temp} = X_i W_K^{temp}, \\ V_i^{temp} &= X_i W_V^{temp} \end{aligned} \tag{27}$$

where $X_i \in R^{T_{obs} \times d_{model}}$ denotes the feature sequence of vehicle i; $W_Q^{temp}, W_K^{temp}, W_V^{temp} \in R^{d_{model} \times d_k}$ denotes the temporal attention weight matrix; $d_k = \frac{d_{model}}{h}$ denotes the dimension of each attention head; and h denotes the number of attention heads.

The attention calculation is expressed in Eqs. (28) and (29):

$$A_i^{temp} = Soft\max\left(\frac{Q_i^{temp} \left(K_i^{temp} \right)^{\top}}{\sqrt{d_k}} + M^{temp} \right) \tag{28}$$

$$Z_i^{temp} = A_i^{temp} V_i^{temp} \in R^{T_{obs} \times d_{model}} \tag{29}$$

where $M^{temp} \in R^{T_{obs} \times T_{obs}}$ denotes the temporal mask, ensuring that attention is restricted to historical time steps; $Soft\max$ denotes row-wise normalization.

The attention mechanism of the Transformer model is shown in Fig. 2:

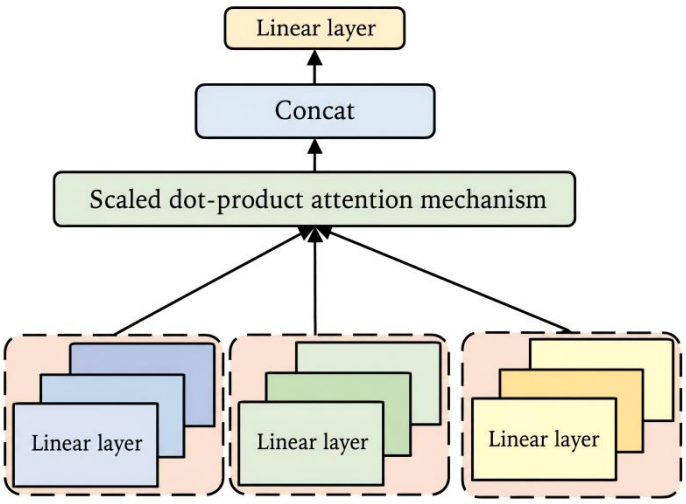


Fig.2 The attention mechanism of the Transformer model

Step 4: Perform residual connection and layer normalization. The specific operation is expressed in Eq. (30):

$$H_i^{temp} = LayerNorm\left(X_i + Z_i^{temp}\right) \tag{30}$$

At time step t, the spatial interactions among vehicles are computed according to Eq. (31):

$$\begin{aligned} &Q_t^{spa} = H^{(t)}W_Q^{spa}, K_t^{spa} = H^{(t)}W_K^{spa}, \\ &V_t^{spa} = H^{(t)}W_V^{spa} \end{aligned} \tag{31}$$

where $H^{(t)} = \left[H_1^{temp}(t), \ldots, H_N^{temp}(t)\right] \in R^{N \times d_{model}}$ denotes the features of all vehicles at time t; $W_Q^{spa}, W_K^{spa}, W_V^{spa} \in R^{d_{model} \times d_k}$ denotes the spatial attention weight matrix.

A distance-aware attention weight is introduced, and its specific calculation is given in Eqs. (32)-(34):

$$D_{ij}^{(t)} = \exp\left(-\frac{\| p_{i,rel}^{(t)} - p_{j,rel}^{(t)} \|^2}{2\sigma^2}\right) \tag{32}$$

$$A_t^{spa} = Soft\max\left(\frac{Q_t^{spa}\left(K_t^{spa}\right)^{\top}}{\sqrt{d_k}} + \log\left(D^{(t)} + \in\right)\right) \tag{33}$$

$$Z_t^{spa} = A_t^{spa}V_t^{spa} \in R^{N \times d_{model}} \tag{34}$$

where σ denotes the distance decay coefficient, set to 10; $\in$ denotes the numerical stability term, set to 10^{-6}; and $\log\left(D^{(t)}\right)$ denotes the enhancement of the influence of nearby vehicles.

The residual connection operation is expressed in Eq. (35):

$$H_t^{spa} = LayerNorm\left(H^{(t)} + Z_t^{spa}\right) \tag{35}$$

Step 5: Perform spatiotemporal feature fusion by reorganizing the spatiotemporal attention outputs. The specific operation is expressed in Eq. (36):

$$H^{fusion} = MLP_{fusion}\left(Flatten\left(H^{spa}\right)\right) \in R^{d_{model}} \tag{36}$$

where $Flatten$ denotes the flattening operation, which flattens $H^{spa} \in R^{N \times T_{obs} \times d_{model}}$ into

$R^{(N\cdot T_{obs}\cdot d_{model})}$; MLP_{fusion} denotes the fusion network.

The conditional prior distribution is calculated according to Eq. (37):

$$p_\theta\left(z \mid H^{fusion}\right)=N\left(z;\mu_\theta,\sigma_\theta^2 I\right) \tag{37}$$

$$\mu_\theta=MLP_\mu\left(H^{fusion}\right),\log\sigma_\theta^2=MLP_\sigma\left(H^{fusion}\right) \tag{38}$$

where $z\in R^{d_z}$ denotes the latent variable; μ_θ, $\sigma_\theta\in R^{d_z}$ denotes the mean and standard deviation of the prior distribution; MLP_μ, MLP_σ denotes the multilayer perceptron with a dimension of $\left[d_{model},64,d_z\right]$.

The trajectory is generated based on the latent variable and encoded features:

$$h_0=Concat\left(z,H^{fusion}\right) \tag{39}$$

$$h_k=GRU\left(h_{k-1},y_{k-1}\right),k=1,\ldots,T_{pred} \tag{40}$$

$$p_\phi\left(y_k \mid h_k\right)=N\left(y_k;\mu_{\phi,k},\sum_{\phi,k}\right) \tag{41}$$

$$\begin{aligned}&\mu_{\phi,\ k}=MLP_\mu^{dec}\left(h_k\right),\\&\sum\nolimits_{\phi,k}=diag\left(\exp\left(MLP_\sigma^{dec}\left(h_k\right)\right)\right)\end{aligned} \tag{42}$$

where $y_k\in R^2$ denotes the position at the *k*-th prediction time step; GRU denotes the gated recurrent unit with a hidden dimension of 128; and $MLP_\mu^{dec}, MLP_\sigma^{dec}$ denotes decoder MLP with dimensions of [128, 64, 2].

The schematic framework of the Transformer model based on spatiotemporal feature fusion is shown in Fig. 3:

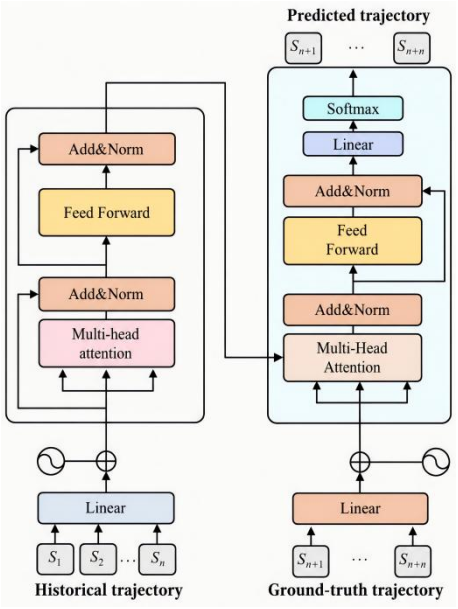


Fig.3 Schematic diagram of the Transformer model structure framework based on the fusion of spatiotemporal features

2.5 Computational flow of the distributed-drive electric vehicle trajectory prediction and risk control method based on a spatiotemporal Transformer and an improved DRA algorithm

This study develops an end-to-end trajectory prediction and risk control framework by

integrating the improved DRA algorithm with a spatiotemporal Transformer model, thereby combining intelligent parameter optimization with deep spatiotemporal modeling. First, spatiotemporal feature sequences are constructed from historical multi-vehicle trajectory data. Second, the improved DRA algorithm is employed to automatically perform global optimization of the key hyperparameters of the spatiotemporal Transformer model, thereby improving prediction accuracy and generalization capability. Finally, the optimized model is used for future trajectory prediction, and the prediction results are further utilized for early risk identification and control strategy generation. The computational flow is as follows:

Step 1: Historical trajectory data acquisition and preprocessing.

Historical trajectory data of distributed-drive electric vehicles are collected from real or simulated scenarios, including multidimensional state information such as vehicle position, velocity, yaw angle, wheel-end torque, and steering angle. The data sampling frequency should satisfy the requirements for characterizing vehicle dynamics and capturing interaction behaviors. The raw data are cleaned, aligned, and interpolated to ensure continuity and consistency of the time series. According to the requirements of the prediction task, the data are divided into fixed-length observation sequences and corresponding prediction sequences to construct sample pairs. Each sample is normalized to eliminate dimensional differences and transformed into a target-vehicle-centered relative coordinate system to enhance the model's generalization capability for learning motion patterns.

Step 2: Hyperparameter optimization of the spatiotemporal Transformer model using the improved DRA algorithm.

The improved DRA algorithm is employed to perform global automatic optimization of the key hyperparameters of the spatiotemporal Transformer model. The optimization objective is to minimize the trajectory prediction error, such as the average displacement error. Each "follower" in the DRA population represents a hyperparameter combination, such as the hidden-layer dimension, number of attention heads, learning rate, and distance decay coefficient. During optimization, the improved adaptive computation-saving mechanism dynamically skips complete evaluations of selected individuals according to changes in individual fitness, thereby significantly reducing the high computational cost caused by repeatedly invoking Transformer training. Meanwhile, the dynamic community interaction mechanism maintains population diversity by probabilistically exchanging individuals among different "religious" communities, effectively preventing premature convergence and enhancing the global exploration capability of the algorithm in a complex non-convex loss-function space.

Through iterative optimization using the DRA algorithm, a set of hyperparameter configurations that yields the optimal prediction performance is finally obtained and used to initialize and train the spatiotemporal Transformer model.

Step 3: Trajectory prediction and risk identification.

The spatiotemporal Transformer model is constructed using the optimized hyperparameters. The preprocessed historical trajectory sequences are fed into the model and jointly modeled through modules including feature embedding, temporal self-attention, spatial interaction attention, and spatiotemporal feature fusion, thereby capturing both individual vehicle motion patterns and multi-vehicle interaction dependencies. The model learns latent representations of trajectory distributions through a conditional variational autoencoder structure and decodes them to generate multi-step future trajectory predictions. During training, the error between the predicted and ground-truth trajectories is used as the loss function, and the model weights are optimized through backpropagation and gradient descent. After training, the model can perform end-to-end trajectory prediction for new observation sequences. Based on the predicted trajectories, future vehicle motion states, such as velocity, acceleration, and yaw rate, as well as the relative relationships with surrounding vehicles and road boundaries, are further calculated to identify potential collision and instability risks. Combined with risk evaluation indicators, corresponding warning signals or preliminary control suggestions, such as desired acceleration and steering-angle correction, are generated to provide decision inputs for subsequent active vehicle control, thereby enabling early risk identification and intervention.

The computational flow of the distributed-drive electric vehicle trajectory prediction and risk control method based on a spatiotemporal Transformer and an improved DRA algorithm is shown in Fig. 4:

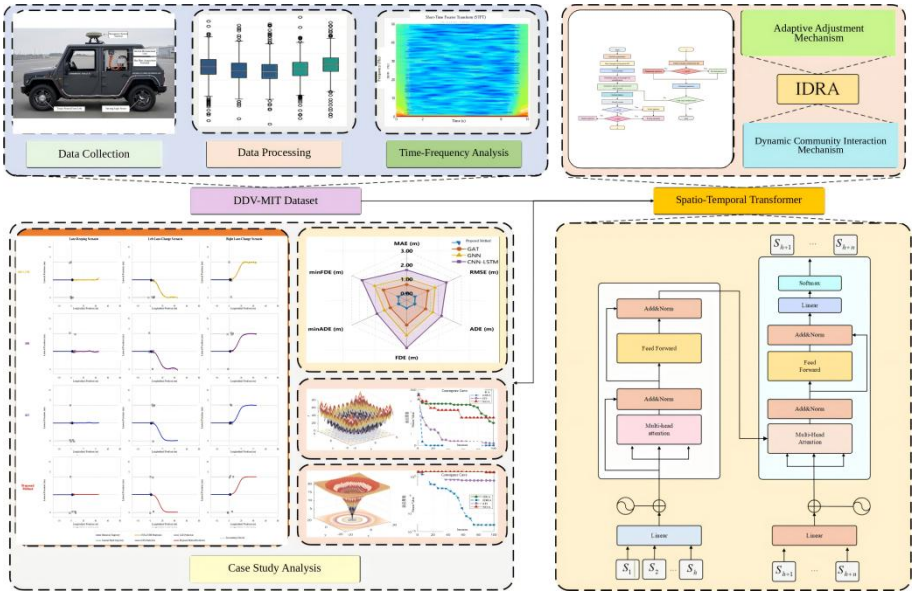


Fig.4 The computational flowchart of the distributed driving trajectory prediction and risk control method for electric vehicles based on the spatiotemporal Transformer and the improved DRA algorithm

3. Results and discussion

3.1 Test experiments of the improved DRA algorithm

To systematically verify the performance superiority of the proposed improved DRA algorithm (IDRA) and the effectiveness of its improvement strategies, rigorous comparative test experiments are designed in this section. The experiments focus on evaluating the global search capability, local convergence accuracy, convergence speed, and stability of the algorithm. Through benchmark-function testing and multi-algorithm comparison, a comprehensive performance evaluation system is established to ensure the reliability, fairness, and persuasiveness of the experimental results, thereby providing a theoretical and experimental foundation for the practical application of the improved algorithm.

To avoid performance evaluation bias caused by a single type of test function and to ensure the comprehensiveness and representativeness of the experiments, eight classical benchmark functions with different characteristics are selected as the experimental test set. These functions can effectively cover the complex optimization requirements that the algorithm may encounter in practical optimization scenarios and comprehensively evaluate the overall optimization performance of IDRA. The selected test functions are widely used for performance verification of intelligent optimization algorithms, and their details are listed in Table 1.

Table 1. Test function

No.	Function name	Domain	Characteristic description
F1	Maximum function	[-100,100]	Computes the element with the largest absolute value in a vector; non-differentiable and multimodal.
F2	Rosenbrock function	[-30,30]	A classical optimization benchmark function with a narrow valley-shaped structure; unimodal but difficult to converge.
F3	Shifted sphere function	[-100,100]	Computes the sum of squares after adding 0.5 to each variable; a simple convex and unimodal function.
F4	Rastrigin function	[-5.12,5.12]	A multimodal function with numerous local minima, commonly used to evaluate global optimization algorithms.
F5	Ackley function	[-32,32]	A high-dimensional multimodal function with flat regions and numerous local minima, making it difficult to optimize.
F6	Griewank function	[-600,600]	A complex function with multiple local minima but a relatively flat overall landscape.

F7	Rational function	[-5,5]	An optimization problem based on rational-function fitting of real-world data, with practical relevance.
F8	Three-dimensional exponential function	[0,1]	A three-dimensional function with multiple Gaussian peaks, commonly used to evaluate local search capability.

To verify the effectiveness of the proposed improvements to the DRA algorithm, comparative experiments are conducted using the original DRA, Nutcracker Optimization Algorithm (NOA) [16], Enterprise Development (ED) algorithm, and IDRA. The eight benchmark functions described above are used for testing, and each algorithm is independently run 40 times. The convergence curves of functions F1-F8 are visualized, and the corresponding simulation results are shown in Fig. 5:

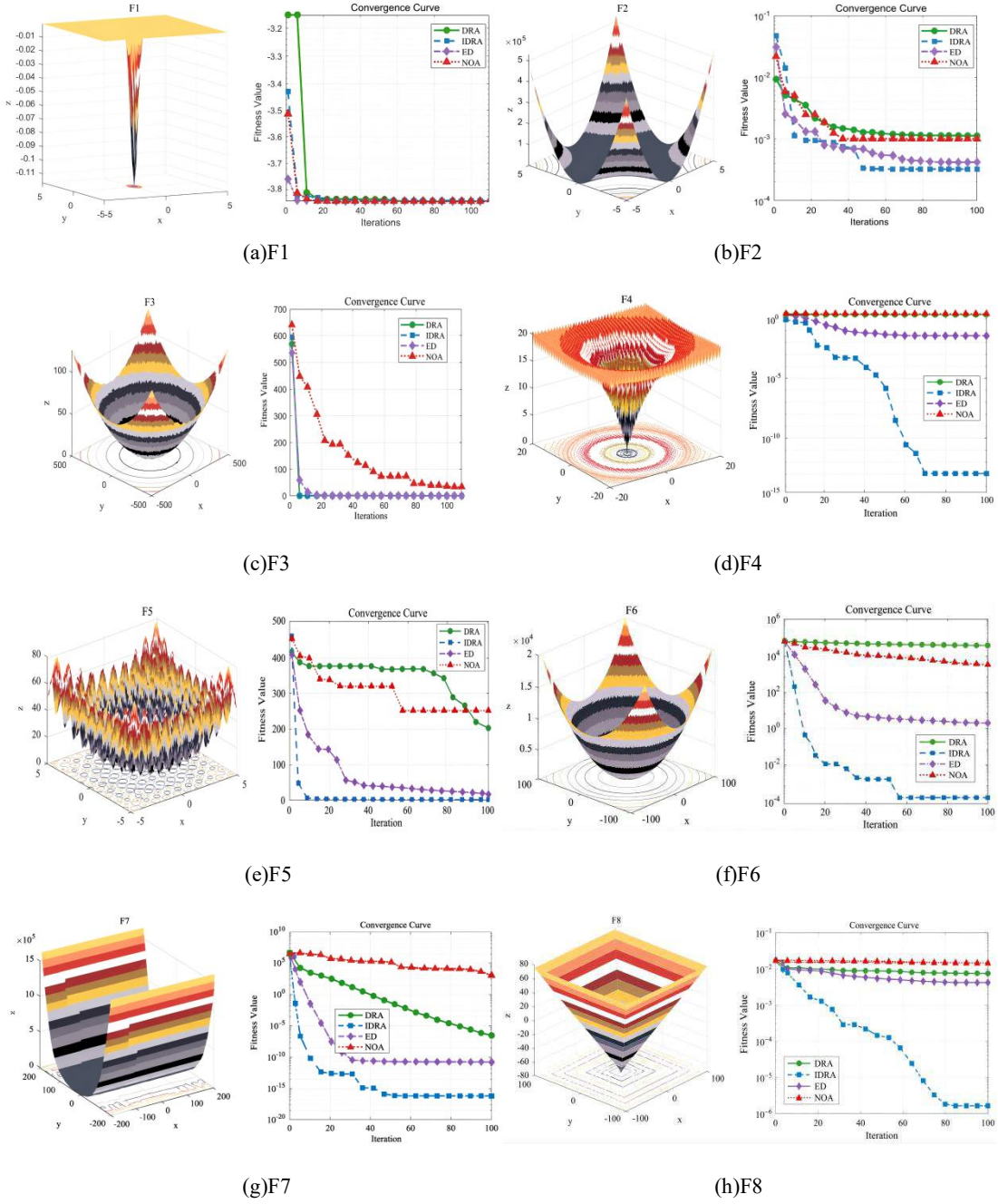


Fig.5 Simulation comparison results of the effectiveness of IDRA algorithm improvement

As shown in Fig. 5, the optimization results obtained on the eight benchmark functions, covering typical complex landscapes such as unimodal, multimodal, noisy, and step-type functions, clearly reveal quantitative performance differences among the four algorithms (DRA, IDRA, ED,

and NOA). For the unimodal functions F1, F4, and F8, IDRA converges to the global optimal accuracy of -3.91 for F1 and 10^{-3} for F4 and F8 after approximately 20 iterations, whereas DRA, ED, and NOA become stable only after approximately 30-50 iterations, indicating that IDRA has clear advantages in both convergence speed and accuracy. For the multimodal functions F2, F3, and F6 containing local extrema, IDRA converges within approximately 10-60 iterations to the optimal values of 10^{-3} and 10^{-2} for F2 and F6, respectively, and 0 for F3. In contrast, ED, DRA, and NOA converge to 10^{-3}, 10^{-2}, and 10^{-1} for F2; 0, 5, and 80 for F3; and 10^{0}, 10^{4}, and 10^{5} for F6, respectively. In particular, IDRA achieves higher accuracy than DRA and NOA on F6. For the highly rugged multimodal function F5 and the step-type function F7, IDRA also outperforms the other three algorithms.

Overall, IDRA demonstrates superior robustness and optimization performance across different types of benchmark functions. Its convergence speed is approximately 1.5-3 times faster than that of the other algorithms, while its accuracy is improved by 1-15 orders of magnitude, providing a more efficient solution for complex optimization problems.

3.2 Data sources for the case study

The experimental data used in this study are obtained from a self-constructed "Multi-Vehicle Interaction Trajectory Dataset for Distributed-Drive Electric Vehicles." The dataset is specifically designed for trajectory prediction and risk control tasks involving distributed-drive electric vehicles and covers a wide range of driving scenarios, vehicle states, and interaction behaviors, making it suitable for spatiotemporal sequence modeling and optimization algorithm validation. The dataset features a high sampling frequency, multidimensional state information, realistic physical constraints, and diverse interaction patterns, thereby effectively supporting the training and evaluation of trajectory prediction models under complex driving environments.

Data acquisition is conducted using eight experimental vehicles equipped with distributed-drive systems. Each vehicle is equipped with an RTK-GPS integrated navigation system, a six-axis IMU, wheel-end torque sensors, steering-angle sensors, and a complete CAN-bus data acquisition system. Vehicle motion states are synchronously recorded at a high sampling frequency of 100 Hz to ensure accurate characterization of dynamic details and interaction timing. The distributed-drive experimental vehicles are shown in Fig. 6:

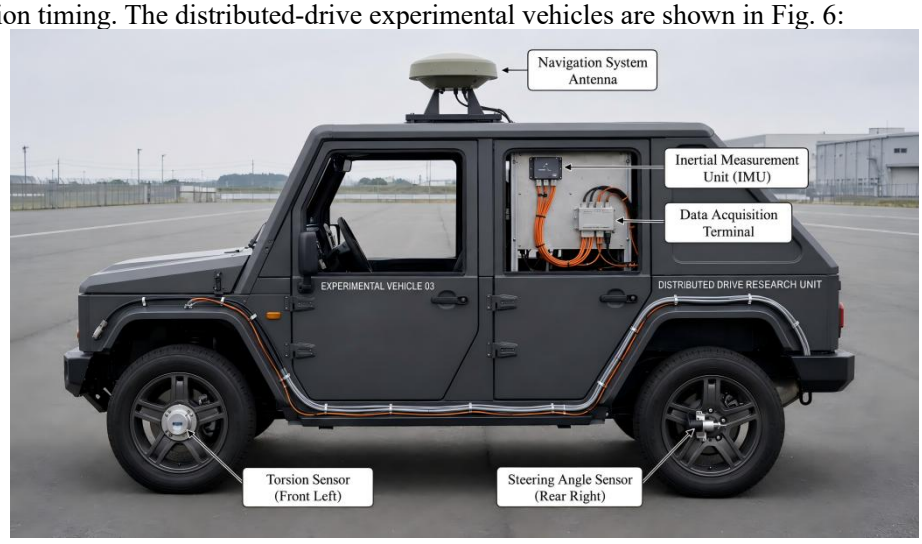


Fig.6 distributed-drive test vehicle

The dataset contains approximately 120 hours of continuous trajectory records and structurally stores multidimensional time-series state information for each vehicle, including global coordinate positions, longitudinal and lateral velocities, yaw angle and yaw rate, independent driving/braking torques of the four wheels, and front-wheel steering angle. All data are subjected to rigorous cleaning, timestamp alignment, and interpolation to ensure continuity and consistency of the sequences. To adapt the data to the model input and improve generalization capability, the raw data are further transformed into a target-vehicle-centered relative coordinate system and standardized using Z-score normalization to eliminate dimensional effects.

In terms of scenario coverage, the dataset includes detailed driving behaviors under different traffic densities and interaction patterns. Urban-road scenarios focus on vehicle interactions during

traffic-light start-stop processes and pedestrian disturbances; highway scenarios record high-speed car-following, active lane changes, and emergency braking behaviors; intersection scenarios include highly interactive events such as turning conflicts and merging; in addition, specially collected curved-road and ramp data are used to evaluate vehicle stability control boundaries. The dataset is finally divided into a training set (80 hours), validation set (20 hours), and test set (20 hours), with detailed scenario labels and interaction-type annotations provided to support supervised learning and algorithm validation.

With its high precision, multidimensionality, and strong interaction characteristics, the dataset provides a solid and reliable data foundation for the training, optimization, and performance evaluation of the spatiotemporal Transformer model and the improved Divine Religion Algorithm. To further investigate the dynamic responses and control characteristics of multiple distributed-drive electric vehicles under interactive scenarios, the main data in the dataset are visualized and analyzed, as shown in Fig. 7:

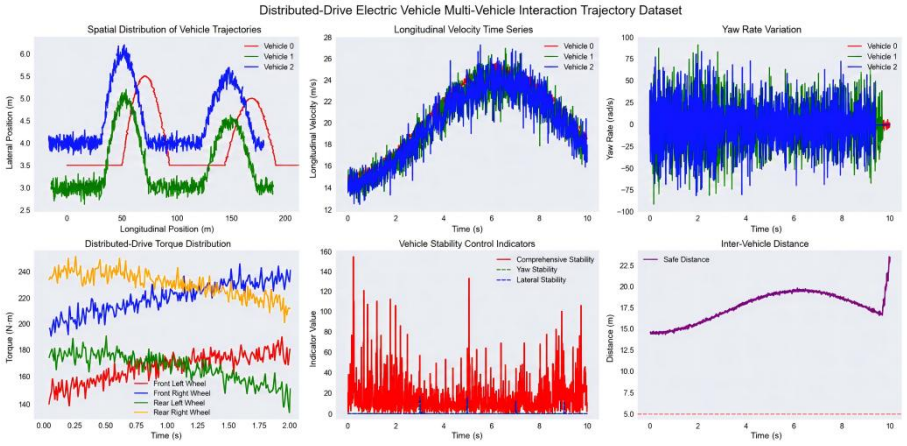


Fig.7 Analysis of dynamic responses and control characteristics of distributed-drive electric vehicle multi-vehicle interacting trajectory data set

As shown in Fig. 7, the spatial distribution of vehicle trajectories reveals distinct lateral-position behaviors among the three experimental vehicles. Vehicle 0 maintains a relatively stable lateral offset, whereas Vehicles 1 and 2 exhibit pronounced lateral oscillations during longitudinal motion, reflecting coordinated obstacle avoidance and trajectory planning strategies during multi-vehicle interactions.

The longitudinal velocity time series show that the three vehicles undergo a synchronized "acceleration–steady-state–deceleration" process. The consistency of velocity fluctuations reflects coordinated behavior during multi-vehicle following, while local high-frequency fluctuations indicate dynamic changes in driving conditions and the rapid response of the drive system.

In the yaw-rate curves, Vehicle 2 exhibits substantially larger fluctuations than the other two vehicles, while Vehicle 1 shows the second-highest fluctuation frequency. This indicates differences among the vehicles in steering frequency and magnitude, which are directly related to the dynamic obstacle-avoidance requirements of multi-vehicle interactions.

The distributed-drive torque allocation results show that the rear-axle wheel torques are generally higher than those of the front axle, reflecting a rear-wheel-dominant torque distribution strategy. Meanwhile, the torque of each wheel is dynamically adjusted over time to accommodate the longitudinal and lateral dynamic requirements during driving.

Among the vehicle stability control indicators, the comprehensive stability index exhibits frequent sharp peaks, whereas the yaw and lateral stability indices remain relatively stable. This suggests that, during multi-vehicle interactions, the system primarily performs high-frequency dynamic corrections to the comprehensive stability state to cope with disturbances under complex operating conditions.

The inter-vehicle distance remains consistently above the safe-distance threshold. Its trend of first increasing, then decreasing, and subsequently increasing again intuitively reflects the dynamic safety interaction characteristics during multi-vehicle following.

3.3 Time-frequency characteristic analysis of experimental data

Time-frequency characteristic analysis is a mathematical tool that can simultaneously

characterize the local features of signals in both the time and frequency domains and is particularly suitable for the processing and interpretation of non-stationary signals. In trajectory prediction and risk control for distributed-drive electric vehicles, vehicle motion states, such as yaw rate and lateral acceleration, are inherently non-stationary time series whose frequency-domain characteristics vary dynamically with driving behavior, road conditions, and multi-vehicle interaction states. Through time-frequency characteristic analysis of the experimental data, the time-varying frequency patterns hidden in the trajectory data can be further explored, and frequency-domain features associated with critical events such as vehicle instability and collision risk can be identified, thereby providing a more refined time-frequency-domain basis for early risk warning and control strategy formulation.

In this study, STFT is adopted as the primary method for time-frequency characteristic analysis. The experimental analysis results are shown in Fig. 8:

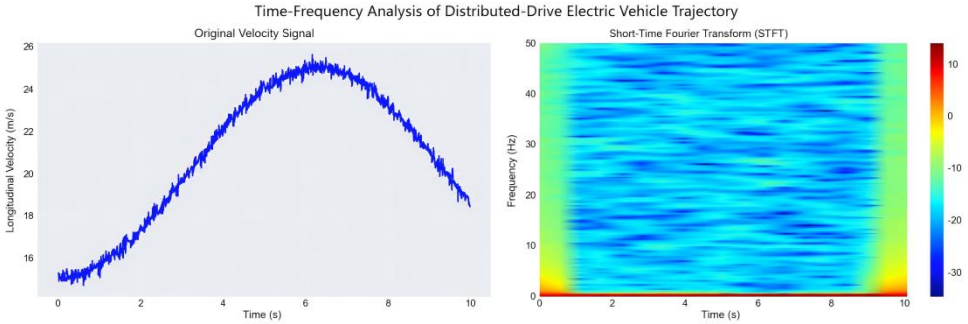


Fig.8 The analysis results of the time-frequency characteristics based on the STFT method

As shown in Fig. 8, the visualization presents a joint time-frequency analysis of the driving trajectory of a distributed-drive electric vehicle. The original velocity signal on the left clearly illustrates the dynamic evolution of the vehicle's longitudinal velocity over 0-10 s. During the initial stage, the vehicle accelerates from approximately 15.5 m/s, with the velocity increasing nonlinearly over time. The velocity reaches a peak of approximately 25.5 m/s at around 6 s and then enters the deceleration stage, exhibiting an overall "acceleration-peak-deceleration" quasi-parabolic pattern. Slight high-frequency fluctuations are also observed in the signal, reflecting velocity disturbances caused by road excitation or dynamic adjustments of the drive system during operation. The short-time Fourier transform (STFT) time-frequency map on the right further reveals the evolution of the frequency components of the velocity signal from a joint time-frequency perspective. The horizontal axis corresponds exactly to the time axis of the original signal, while the vertical axis covers the frequency range of 0-50 Hz. The color-mapped amplitude values intuitively represent the energy intensity at different time-frequency points. It can be observed that during the initial acceleration stage (0-2 s) and the final deceleration stage (8-10 s), significant high-energy concentrations appear in the low-frequency range of 0-10 Hz, corresponding to periods with relatively large velocity variation rates and reflecting the low-frequency dynamic response of the drive system during rapid acceleration and deceleration. In contrast, during the relatively steady driving stage (3-7 s), the energy distribution is more evenly dispersed over the mid- and high-frequency ranges, while the amplitude fluctuations are significantly reduced, indicating smoother velocity variations and more stable frequency components.

Overall, the time-domain dynamics of the original velocity signal correspond well with the time-frequency energy distribution obtained by STFT. The time–frequency analysis not only quantifies the differences in frequency characteristics among different driving stages but also reveals the time-frequency evolution of the dynamic response of distributed-drive electric vehicles during trajectory tracking, providing important quantitative support in the time–frequency domain for further understanding vehicle torque distribution strategies, driving stability, and dynamic load variations.

3.4 Experimental comparison results of different trajectory prediction and risk control methods for distributed-drive electric vehicles

To comprehensively evaluate the overall performance of the proposed method, comparative experiments are conducted using CNN-LSTM [17], GNN [18], GAT [19], and the proposed method.

All comparison methods are tested on the self-constructed multi-vehicle interaction trajectory dataset for distributed-drive electric vehicles. The experimental results are shown in Fig. 9:

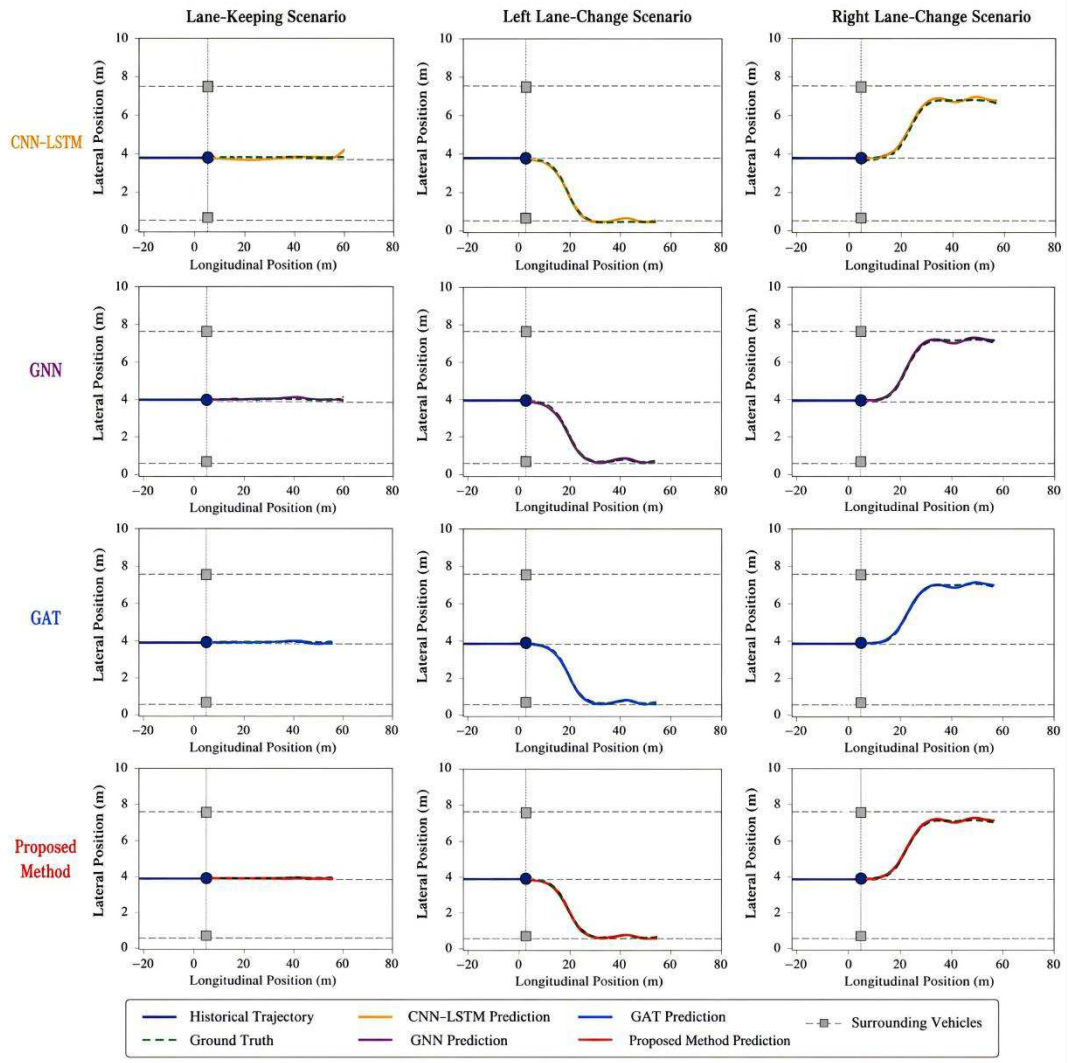


Fig.9 Experimental comparison results of different distributed driving methods for predicting vehicle trajectories and controlling risks in electric vehicles

Fig. 9 systematically compares the trajectory prediction performance of CNN-LSTM, GNN, GAT, and the proposed method under typical autonomous-driving scenarios, including lane keeping, left lane change, and right lane change. The variation curves of lateral position with longitudinal driving distance intuitively illustrate the ability of different methods to model vehicle driving intentions and their trajectory tracking accuracy. In the low-dynamic steady-state lane-keeping scenario, the predicted trajectories of all four methods closely follow the historical driving trajectory (dark blue line), with the lateral position remaining stable around the lane center at approximately 4 m. Among them, GAT produces the smoothest predicted trajectory, while CNN-LSTM and the proposed method also demonstrate stable lane-keeping capability. Only the GNN method exhibits slight lateral fluctuations, indicating that all methods possess basic trajectory prediction reliability under steady-state driving conditions.

In the highly dynamic left-lane-change scenario, the ground-truth trajectory exhibits a nonlinear decrease in lateral position from approximately 4 m to around 0 m, revealing clear differences among the methods. The purple trajectory predicted by GNN decreases too slowly and shows a relatively large deviation from the ground truth. Although the blue trajectory predicted by GAT captures the downward trend, its terminal position deviates from the actual trajectory. The orange trajectory of CNN-LSTM approximately follows the lane-change trend but exhibits a slight deviation in the later stage. In contrast, the red trajectory predicted by the proposed method closely matches both the descending slope and terminal position of the ground-truth trajectory,

accurately characterizing the lateral displacement variation during the left-lane-change process.

In the right-lane-change scenario, the ground-truth trajectory shows an increase in lateral position from approximately 4 m to around 8 m. The purple trajectory of GNN exhibits an insufficient increase and fails to fully match the actual displacement. Although the blue trajectory of GAT reaches the target lateral position, slight fluctuations occur in the later stage. The orange trajectory of CNN-LSTM increases steadily, but its agreement with the ground-truth trajectory can still be improved. In comparison, the red trajectory predicted by the proposed method not only accurately follows the increasing trend of lateral position but also closely matches the terminal position of the ground-truth trajectory, demonstrating excellent modeling capability for nonlinear trajectories in right-lane-change scenarios.

Overall, the performance differences among the methods are relatively small in the low-dynamic lane-keeping scenario. However, in highly dynamic lane-change scenarios, the proposed method achieves more accurate trajectory prediction than CNN-LSTM, GNN, and GAT through superior spatiotemporal feature fusion and intention modeling capability, providing more reliable technical support for autonomous-driving trajectory control in complex traffic environments. In addition, to further investigate the differences between the proposed method and other mainstream methods across various evaluation metrics, the results are visualized, as shown in Fig. 10.

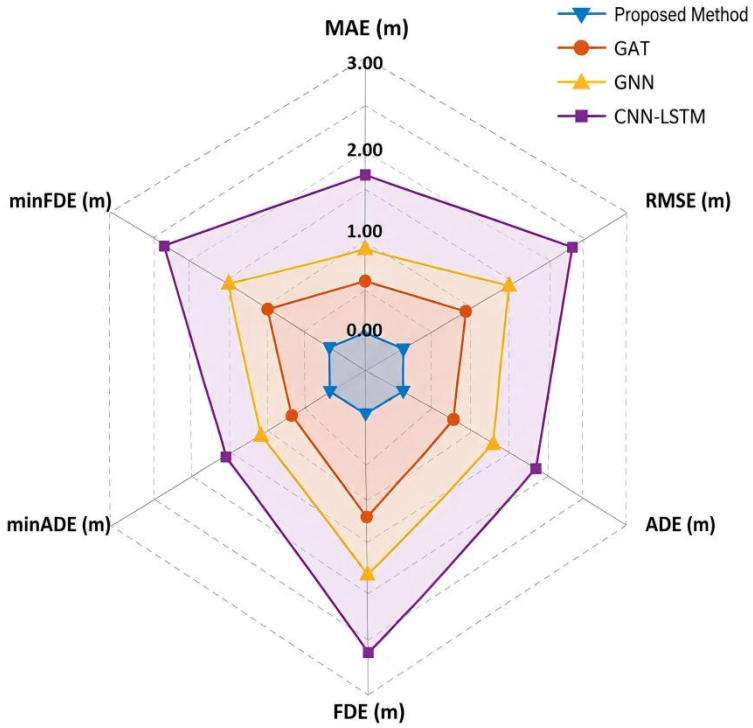


Fig. 10 Multi-index comparison radar chart of different methods

As shown in Fig. 10, the four comparison methods exhibit clear hierarchical differences in performance. The proposed method (blue curve) achieves the best prediction accuracy across all evaluation metrics, with all error values located closest to the center of the radar chart, indicating a clear advantage in reducing trajectory prediction errors. GAT (orange curve) and GNN (yellow curve) show intermediate performance, with similar error levels across the metrics. Their errors are significantly higher than those of the proposed method but lower than those of CNN-LSTM (purple curve). CNN-LSTM yields the largest errors for all metrics, particularly for FDE, minFDE, and minADE, where its values are substantially higher than those of the other three methods, revealing clear limitations in trajectory prediction error control.

Further analysis shows that the performance differences are particularly pronounced for FDE (Final Displacement Error) and minFDE (minimum Final Displacement Error). The error values of the proposed method are close to 0, whereas those of CNN-LSTM exceed 2 m, intuitively demonstrating the gap in core trajectory prediction capability among the different algorithms. Overall, the radar chart clearly quantifies the performance of the four algorithms across multiple error metrics, providing an intuitive and comprehensive basis for algorithm comparison and

subsequent optimization. The detailed numerical results of the multi-metric comparison among different methods are presented in Table 2:

Table 2. Numerical results of multi-index comparison radar chart experiments using different methods

Model	MAE (m)	RMSE (m)	ADE (m)	FDE (m)	minADE (m)	minFDE (m)
Proposed method	0.03	0.04	0.02	0.05	0.01	0.03
GAT	0.62	0.85	0.71	1.25	0.54	0.98
GNN	1.08	1.32	1.18	1.98	0.92	1.55
CNN-LSTM	1.65	2.04	1.72	2.85	1.38	2.35

4. Conclusion

(1) Conclusion 1: The IDRA-optimized spatiotemporal Transformer model proposed in this study significantly improves trajectory prediction accuracy and outperforms the comparison methods across all evaluated error metrics. On the self-constructed multi-vehicle interaction trajectory dataset, the proposed method achieves an MAE of 0.03 m, RMSE of 0.04 m, ADE of 0.02 m, and FDE of 0.05 m. Based on these four metrics, the prediction errors are reduced by approximately 95.9% on average compared with the second-best GAT method and by approximately 98.3% compared with the CNN-LSTM baseline. In addition, the proposed method achieves minADE and minFDE values of 0.01 m and 0.03 m, respectively, whereas the maximum FDE among the comparison methods reaches 2.85 m. These quantitative results demonstrate that IDRA-based hyperparameter optimization combined with spatiotemporal attention modeling can effectively enhance the representation of nonlinear vehicle-motion evolution and improve trajectory prediction performance under complex multi-vehicle interaction scenarios.

(2) Conclusion 2: The proposed method also demonstrates strong trajectory tracking capability under highly dynamic driving scenarios. In both left- and right-lane-change conditions, its predicted trajectories more closely follow the ground-truth lateral-position variations and terminal positions than those generated by CNN-LSTM, GNN, and GAT. The improved trajectory prediction accuracy provides more reliable information for identifying potential collision and instability risks and for generating subsequent warning and control-decision inputs. Therefore, the proposed framework provides an effective trajectory prediction basis for early risk identification and active-safety decision-making of distributed-drive electric vehicles under complex interactive traffic conditions.

5. List of abbreviations

Full name	Abbreviations
Divine Religions Algorithm	DRA
Improved Divine Religions Algorithm	IDRA
Long Short-Term Memory	LSTM
Convolutional Neural Network	CNN
Convolutional Neural Network–Long Short-Term Memory	CNN-LSTM
Graph Neural Network	GNN
Graph Attention Network	GAT
Nutcracker Optimization Algorithm	NOA
Enterprise Development Algorithm	ED
Multilayer Perceptron	MLP
Gated Recurrent Unit	GRU
Real-Time Kinematic Global Positioning System	RTK-GPS
Inertial Measurement Unit	IMU
Controller Area Network	CAN

Short-Time Fourier Transform	STFT
Mean Absolute Error	MAE
Root Mean Square Error	RMSE
Average Displacement Error	ADE
Final Displacement Error	FDE
Minimum Average Displacement Error	minADE
Minimum Final Displacement Error	minFDE

6. Declarations

Clinical Trial Number: Not applicable. This study does not involve any clinical trials.

Consent to Participate: Not applicable. This study does not involve human participants, human data, or human tissue.

Consent to Publish: Not applicable. This study does not contain any individual person’s data in any form (including individual details, images, or videos).

Ethics Approval: Not applicable. This research does not involve human or animal subjects, and therefore does not require ethical approval.

Availability of data and material: The data used in this study are primary data collected by the authors from a self-built rolling bearing accelerated life test platform. The datasets generated and analyzed during the current study are not publicly available due to the nature of the self-built experimental setup, but are available from the corresponding author on reasonable request.

Competing interests: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Funding: This study was supported by the Scientific Research Project of the Hunan Provincial Department of Education (Grant No. 25A0760) and the Teaching Reform Research Project of Regular Undergraduate Universities in Hunan Province (Grant No. 202502001951).

Authors' contributions: Hu Wang was responsible for the conceptualization, methodology development, research design, project administration, and preparation of the original manuscript. Yuepeng Li was responsible for data acquisition, data preprocessing, experimental investigation, and visualization of the results. Qizheng Yang contributed to formal analysis, result interpretation, validation, and review and editing of the manuscript. All authors discussed the results, contributed to the revision of the manuscript, and approved the final version.

Acknowledgements: The authors gratefully acknowledge the financial support provided by the Scientific Research Project of the Hunan Provincial Department of Education (Grant No. 25A0760) and the Teaching Reform Research Project of Regular Undergraduate Universities in Hunan Province (Grant No. 202502001951).

References

[1] Huang, Y., Du, J., Yang, Z., et al. (2022). A survey on trajectory-prediction methods for autonomous driving. IEEE Transactions on Intelligent Vehicles, 7(3), 652-674. https://doi.org/10.1109/TIV.2022.3167103.

[2] Long, K., Sheng, Z., Shi, H., et al. (2025). Physical enhanced residual learning (PERL) framework for vehicle trajectory prediction. Communications in Transportation Research, 5, 100166. https://doi.org/10.1016/j.commtr.2025.100166.

[3] Tian, T., Li, G., Li, Y., et al. (2023). Trajectory tracking control study of unmanned fully line-controlled distributed drive electric vehicles. Applied Sciences, 13(11), 6465. https://doi.org/10.3390/app13116465.

[4] Guan, Y., Li, N., Chen, P., et al. (2025). Vehicle trajectory prediction method integrating spatiotemporal relationships with hybrid time-step scene interaction. Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering, 239(10-11), 4666-4679. https://doi.org/10.1177/09544070241277412.

[5] Ramadevi, N., Subramanyam, V. M., & Bindu, S. C. (2024). Mobility target tracking with meta-heuristic aided target movement prediction scheme in WSN using adaptive distributed extended Kalman filtering. International Journal of Communication Systems, 37(11), e5789. https://doi.org/10.1002/dac.5789.

[6] Hamedi, H., & Shad, R. (2022). Lane-changing trajectory prediction modeling using neural networks. Advances in Civil Engineering, 2022, 9704632. https://doi.org/10.1155/2022/9704632.

[7] Qiao, S., Gao, F., Wu, J., et al. (2024). An enhanced vehicle trajectory prediction model leveraging LSTM and social-attention mechanisms. IEEE Access, 12, 1718-1726. https://doi.org/10.1109/ACCESS.2023.3345643.

[8] Min, H., Xiong, X., Wang, P., et al. (2024). A hierarchical LSTM-based vehicle trajectory prediction method considering interaction information. Automotive Innovation, 7(1), 71-81. https://doi.org/10.1007/s42154-023-00261-0.

[9] Wang, R., Lin, W., Ren, G., et al. (2025). Interaction-aware vehicle trajectory prediction using spatial-temporal dynamic graph neural network. Knowledge-Based Systems, 327, 114187. https://doi.org/10.1016/j.knosys.2025.114187.

[10] Teng, F., Jin, L., Wang, J., et al. (2026). High-precision multimodal vehicle trajectory prediction model based on cross-layer interleaved spatiotemporal attention mechanism. Engineering Applications of Artificial Intelligence, 167, 113937. https://doi.org/10.1016/j.engappai.2026.113937.

[11] Zhong, Y., Gui, Z., Gao, Z., et al. (2025). LITransformer: Transformer-based vehicle trajectory prediction integrating spatiotemporal attention networks with lane topology and dynamic interaction. Electronics, 14(24), 4950. https://doi.org/10.3390/electronics14244950.

[12] Alkaabi, N., Shakya, S., & Mizouni, R. (2026). A hyperparameter optimization framework for Transformer-based time series forecasting using evolutionary algorithms. Neural Computing and Applications, 38, 236. https://doi.org/10.1007/s00521-026-11959-7.

[13] Jlifi, B., Ferjani, S., & Duvallet, C. (2025). A genetic algorithm based three hyperparameter optimization of deep long short term memory (GA3P-DLSTM) for predicting electric vehicles energy consumption. Computers & Electrical Engineering, 123, 110185. https://doi.org/10.1016/j.compeleceng.2025.110185.

[14] Xie, H., Zhang, L., & Lim, C. P. (2020). Evolving CNN-LSTM models for time series prediction using enhanced Grey Wolf Optimizer. IEEE Access, 8, 161519-161541. https://doi.org/10.1109/ACCESS.2020.3021527.

[15] Rajwar, K., Deep, K., & Das, S. (2023). An exhaustive review of the metaheuristic algorithms for search and optimization: Taxonomy, applications, and open challenges. Artificial Intelligence Review, 56(11), 13187-13257. https://doi.org/10.1007/s10462-023-10470-y.

[16] Abdel-Basset, M., Mohamed, R., Jameel, M., et al. (2023). Nutcracker optimizer: A novel nature-inspired metaheuristic algorithm for global optimization and engineering design problems. Knowledge-Based Systems, 262, 110248. https://doi.org/10.1016/j.knosys.2022.110248.

[17] Wang, K., Qu, D., Shao, D., et al. (2024). Research on risk quantification methods for connected autonomous vehicles based on CNN-LSTM. Applied Sciences, 14(23), 11204. https://doi.org/10.3390/app142311204.

[18] Li, F. J., Zhang, C. Y., & Chen, C. L. P. (2023). STS-DGNN: Vehicle trajectory prediction via dynamic graph neural network with spatial-temporal synchronization. IEEE Transactions on Instrumentation and Measurement, 72, 1-13. https://doi.org/10.1109/TIM.2023.3307179.

[19] Zheng, X., Chen, X., & Jia, Y. (2024). Vehicle trajectory prediction based on GAT and LSTM networks in urban environments. Promet - Traffic&Transportation, 36(5), 867-884. https://doi.org/10.7307/ptt.v36i5.600.

Biography



Wang Hu, born in 1986 in Yiyang, Hunan Province, holds a master's degree in engineering and is an associate professor at Hunan Institute of Traffic Engineering. His main research interests include automotive control technology and multi-sensor fusion algorithms. He has published 14 academic papers and led six provincial-level or higher-level research projects.