

Physics-Informed Neural Network Modeling of Vehicle Collision Dynamics in Precision Immobilization Technique Maneuvers

Yangye Jiang^a, Jiachen Wang^a and Daofei Li^{a*}

^aHuman Mobility Automation Group, Institute of Power Machinery and Vehicular Engineering, College of Energy Engineering, Zhejiang University, Hangzhou, China

ARTICLE HISTORY

Compiled October 16, 2025

ABSTRACT

Accurate prediction of vehicle collision dynamics is crucial for advanced safety systems and post-impact control applications, yet existing methods face inherent trade-offs among computational efficiency, prediction accuracy, and data requirements. This paper proposes a dual Physics-Informed Neural Network framework addressing these challenges through two complementary networks. The first network integrates Gaussian Mixture Models with PINN architecture to learn impact force distributions from finite element analysis data while enforcing momentum conservation and energy consistency constraints. The second network employs an adaptive PINN with dynamic constraint weighting to predict post-collision vehicle dynamics, featuring an adaptive physics guard layer that prevents unrealistic predictions while preserving data-driven learning capabilities. The framework incorporates uncertainty quantification through time-varying parameters and enables rapid adaptation via fine-tuning strategies. Validation demonstrates significant improvements: the impact force model achieves relative errors below 15.0% for force prediction on finite element analysis (FEA) datasets, while the vehicle dynamics model reduces average trajectory prediction error by 63.6% compared to traditional four-degree-of-freedom models in scaled vehicle experiments. The integrated system maintains millisecond-level computational efficiency suitable for real-time applications while providing probabilistic confidence bounds essential for safety-critical control. Comprehensive validation through FEA simulation, dynamic modeling, and scaled vehicle experiments confirms the framework's effectiveness for Precision Immobilization Technique scenarios and general collision dynamics prediction.

KEYWORDS

Precision immobilization technique; Post impact control; Vehicle collision dynamics; Physics-informed neural networks; Gaussian mixture model; Uncertainty quantification

1. Introduction

1.1. Background and Motivation

Road traffic accidents are still a significant threat to public safety. Vehicle collisions in traffic accidents often go through three dynamic stages: pre-collision, during-collision, and post-collision [1]. Although its time duration with direct vehicle collision is very short, the second stage of ‘during collision’ is crucial in the entire vehicle behavior

progression after accidents. Therefore, accurately predicting the vehicle motion dynamics in collision is not only of great significance for accident reconstruction analysis and vehicle safety system design, but also a key technological foundation for developing next-generation active safety technologies, e.g. post-impact control strategies [2; 3; 4]. Vehicle rear-quarter collisions, where impacts occur at the vehicle's rear corner area as in Precision Immobilization Technique (PIT) maneuvers, represent one of the most complex and dynamically challenging impact scenarios in traffic safety research [5; 6; 7]. The research of PIT technique and post-impact control systems have created new demands for accurate, real-time collision prediction capabilities that traditional methods struggle to meet. With advancing research in vehicle active safety control, there is a growing urgent need for a modeling approach that balances accuracy and computational efficiency to support model-based control systems, enhance post-impact reliability, and promote practical applications.

1.2. Literature Review

1.2.1. Impact Force Modeling Methods

To better model vehicle post-impact dynamics, it is necessary to understand the vehicle behavior changes during the collision process, which are intuitively reflected in the impact force curve. For lite collisions such as PIT maneuvers, the force curve can be estimated from accelerometer data [8]. Huibers et al. [8] found that for collisions with the same barrier type, the trend of force curve $F(x)$ remains consistent across different vehicle impacts. However, the magnitude of impact force varies depending on specific operational condition and the structural characteristic of the vehicle involved.

Various modeling methods for impact force have been developed. Lumped parameter models based on spring-damper systems [9; 10] and pulse models [11; 12] represent common approaches. Pulse approximation methods, including haversine, half-sine, and triangular waveforms, have been widely adopted in the vehicle safety domain [13; 14]. These simplified mathematical formulations provide computationally efficient representations of impact forces, enabling rapid analysis and preliminary design evaluations while maintaining acceptable accuracy for certain crash scenarios. For more detailed analysis, multibody dynamics models in the discrete time domain [15] and finite element models [16; 17] can be employed. These approaches enable comprehensive investigation of vehicle behavior in terms of deformation, displacements, velocity, and accelerations throughout the entire impact duration. However, the substantial computational demand limits their application to analyzing vehicle dynamics and modeling transient collision behaviors, rendering them unsuitable for direct use in developing vehicle active safety algorithms. Nevertheless, the data generated by these methods can still, to some extent, substitute for real-world collision data that are prohibitively expensive to obtain.

Chen et al. [16] developed a deep neural network-based approach for solving inverse problems in vehicle collisions using finite element data, achieving prediction accuracy with mean errors below 3%. Yildiz and Kiran [17] applied the finite element method to particle swarm-based optimization for vehicle crash safety.

1.2.2. Post-impact Vehicle Dynamics Modeling

Vehicle post impact motion prediction faces multiple challenges. Firstly, the collision process involves complex nonlinear dynamic phenomena, including the coupled effects

of various factors such as structural deformation, changes in tire mechanical properties, and suspension system responses [18]. Secondly, post-impact vehicle motion often exhibits large sideslip angles and yaw rates that far exceed the dynamic response range under normal driving conditions [19]. Furthermore, in special collision scenarios such as PIT maneuvers, the vehicle dynamic response becomes even more complex, making it difficult for traditional vehicle dynamics models to accurately describe its motion patterns. Research by the Michigan State Police has revealed that performing PIT maneuvers on vehicles equipped with Electronic Stability Control (ESC) systems can yield unpredictable outcomes at both low and high speeds, whereas vehicles without ESC demonstrate significantly more consistent and predictable responses [20].

Existing collision modeling methods face inherent limitations. Momentum conservation approaches, such as the Kudlich-Slibar method, offer high computational efficiency but assume instantaneous collision and neglect critical physical phenomena during impact, such as tire forces [21; 22; 23; 24; 25]. To address these issues, Zhou et al. [5; 6] developed a multi-stage vehicle dynamics model for PIT, quantitatively analyzing the influence of operational parameters on impact outcomes. Elmarakbi et al. [26] simulated vehicle oblique frontal collisions using a 6-degree-of-freedom mathematical model, while Gidlewski et al. [27] modeled the dynamics and energy balance of vehicle frontal-to-side collisions. However, these studies rely on oversimplified linear tire slip angle models, which fail to accurately capture actual vehicle responses. Kim et al. [12] introduced an effective factor to approximate the nonlinear region of tire behavior, enabling a 4DOF model to describe vehicle responses under collision forces and validate against CarSim simulations. Nevertheless, this method still requires extensive precise model parameters and calibration based on tire force data, which is impractical in emergency situations characterized by high uncertainty.

1.2.3. Vehicle Dynamic Modeling Based on Data-driven Methods

In recent years, with the rapid development of artificial intelligence technology, especially the successful application of deep learning in various fields, machine learning methods have demonstrated great potential in vehicle dynamics modeling [28; 29]. Spielberg et al. [30] developed a high-performance autonomous driving vehicle model based on neural networks, using a feedforward-feedback control architecture and historical state information to handle vehicle dynamics under different road friction conditions. Williams et al. [31] combined the model predictive path integral (MPPI) algorithm and a neural network-learned dynamics model to achieve real-time control of complex nonlinear systems. Pan et al. [32] integrated complex multibody dynamics simulation with deep learning to achieve real-time vehicle dynamics prediction. These methods utilize large amounts of data for neural network-based vehicle dynamics modeling, particularly the nonlinear tire mechanics in vehicles [30; 33]. However, these methods rely on extensive training data, and lack physical model transparency and intuitiveness, while the trained models can only be used for specific vehicle types and operating conditions. Although purely data-driven neural network models excel at learning complex nonlinear patterns, they often fail to guarantee consistency with physical constraints, potentially generating physically unrealizable state prediction for the vehicle.

To overcome the limitations of purely data-driven methods, Physics-Informed Neural Networks (PINN) have emerged as an effective solution. PINN represents a class of universal function approximators that embed physical laws, formulated as partial differential equations, into the neural network learning process [34; 35]. Unlike meth-

ods that rely on data generated from physical models [30], PINN incorporates physical prior knowledge as regularization terms to constrain the solution space of acceptable functions, thereby enhancing the generalization capability of function approximation even under data-scarce conditions. This advantage is particularly evident for simple vehicle models [36; 37; 38; 39]. Such approach is especially suitable for engineering scenarios where data availability is limited. For example, Tan et al. [40] utilize PINN to model the dynamics of experimental vehicles. To address the challenge of neural network convergence with limited data, Fang et al. [41] proposed a Fine-Tuning Hybrid Dynamics (FTHD) method, which fine-tunes a pre-trained Deep Dynamics Model (DDM) using a small training dataset.

Similarly, the PINN-based approach proposed by Long et al. [42] enhances conventional neural networks by incorporating physical prior knowledge such as vehicle kinematics and boundary conditions. These methods demonstrate excellent extrapolation capability across dynamic scenarios, enabling the reconstruction of high-speed trajectories using only low-speed training data. However, their limitations primarily lie in excessive reliance on physical models, restricting their applicability to relatively simple scenarios and making them inadequate for characterizing uncertainties in complex dynamic environments.

1.2.4. Research Gaps

Despite significant advances in both collision modeling and physics-informed learning, several critical gaps remain:

- (1) Existing neural network architectures struggle to effectively handle the strong discontinuities of contact and force during collisions.
- (2) Collisions involve multiple time scales, from millisecond impact duration to seconds-long post-collision motion. Most existing studies can only simplify the impact process based on experience [1; 6].
- (3) Traditional neural networks require extensive training data, which is costly and dangerous to obtain for collision scenarios.
- (4) To address security issues, existing models struggle to balance computational complexity and accuracy.

Based on the current research status and existing challenges, this paper proposes two PINN-based neural networks. The first network integrates finite element analysis (FEA) data with baseline physical models to model millisecond-level impact forces, while the second network incorporates fine-tuning techniques to achieve accurate modeling of post-collision vehicle attitude changes. Both models integrate gaussian mixture models (GMM) to handle uncertain state distributions. The first model is validated using a self-constructed FEA database, demonstrating its capability for accurate impact force modeling. The second model is validated through a combined FEA-CarSim simulation environment and scaled vehicle experiments.

1.3. Contributions

This paper addresses these gaps through the following contributions:

- (1) We propose a dual physics-informed neural network framework where the first network models time-varying collision force distributions and the second predicts post-collision vehicle dynamics, achieving complete collision process modeling

- across temporal scales.
- (2) We design physics guard layer that balances physical consistency with data learning through dynamic constraint weight adjustment and soft boundary constraints, effectively addressing the issue of physical constraints suppressing neural network learning.
 - (3) We establish a multi-fidelity validation framework from FEA to CarSim simulation and scaled vehicle experiments, reducing trajectory prediction error by 63.6% compared to traditional 4DOF models while maintaining millisecond-level computational efficiency.

2. Baseline Vehicle Collision Dynamics Modeling

2.1. Vehicle Dynamics Model Considering Impact Forces

Since PINN training requires incorporating physical constraints, a continuous baseline model must first be established to provide the necessary physical foundations for the vehicle dynamics modeling framework. As shown in Figure 1, to capture essential collision dynamics while maintaining computational tractability, we adopt a four-degree-of-freedom (4DOF) vehicle model that extends traditional planar approaches by including roll motion, which is critical for accurate representation of vehicle behavior during high-acceleration maneuvers and collision events [12]. The state vector is defined as:

$$\mathbf{x} = [v_x, v_y, \psi, \dot{\psi}, \phi, \dot{\phi}, X, Y]^T \in \mathbb{R}^8 \quad (1)$$

where v_x and v_y are longitudinal and lateral velocities, $\dot{\psi}$ is yaw rate, $\dot{\phi}$ is roll rate, ϕ is roll angle, (X, Y) is global position, and ψ is heading angle.

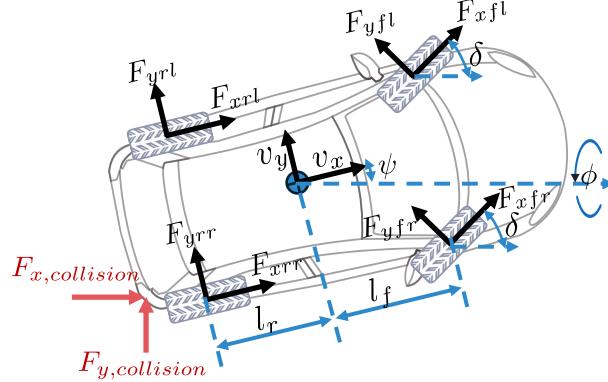


Figure 1. Schematic diagrams of the 4DOF vehicle model with impact forces applied

The governing equations follow the lateral-yaw-roll model with impact force aug-

mentation, incorporating the coupling effects between lateral, yaw, and roll dynamics:

$$\begin{aligned} m\dot{v}_x &= mv_y\dot{\psi} + F_{x,\text{tire}} + F_{x,\text{collision}} \\ m\dot{v}_y &= -mv_x\dot{\psi} + F_{y,\text{tire}} + F_{y,\text{collision}} \\ I_{zz}\ddot{\psi} &= M_{z,\text{tire}} + M_{z,\text{collision}} \\ I_{xx,s}\ddot{\phi} + I_{xz}\ddot{\psi} &= m_s g h_{\text{rc}} \sin \phi - K_s \phi - B_s \dot{\phi} + M_{x,\text{collision}} \end{aligned} \quad (2)$$

where m is vehicle mass, I_{zz} and $I_{xx,s}$ are yaw and roll moments of inertia, I_{xz} is product of inertia representing the coupling between yaw and roll motions, K_s and B_s are roll stiffness and damping coefficients, m_s is sprung mass, and h_{rc} is roll center height.

The resultant tire forces and moments are computed as the sum of contributions from all four wheels:

$$\begin{aligned} F_{x,\text{tire}} &= F_{x,fl} + F_{x,fr} + F_{x,rl} + F_{x,rr} \\ F_{y,\text{tire}} &= F_{y,fl} + F_{y,fr} + F_{y,rl} + F_{y,rr} \\ M_{z,\text{tire}} &= l_f(F_{y,fl} + F_{y,fr}) - l_r(F_{y,rl} + F_{y,rr}) + \frac{t_w}{2}(F_{x,fr} - F_{x,fl} + F_{x,rr} - F_{x,rl}) \end{aligned} \quad (3)$$

where l_f and l_r are distances from the center of gravity to the front and rear axles, respectively, and t_w is track width.

The individual tire forces are modeled using a simplified magic formula tire model [1], which provides a good balance between computational efficiency and physical accuracy. The lateral forces are expressed as:

$$F_{y,i} = -\sin(\tan^{-1}(C_i \alpha_i)) \sqrt{(\mu_s F_{zi})^2 - F_{xi}^2}, \quad \text{with } i = \{fl, fr, rl, rr\} \quad (4)$$

where $i \in fl, fr, rl, rr$ are tire position (front-left, front-right, rear-left, rear-right), μ_s is tire-road friction coefficient and slip angles α_i incorporate the effects of vehicle kinematics and steering input:

$$\begin{aligned} \alpha_{fl} &= \delta - \arctan\left(\frac{v_y + l_f \dot{\psi}}{v_x - \frac{w}{2} \dot{\psi}}\right) \\ \alpha_{fr} &= \delta - \arctan\left(\frac{v_y + l_f \dot{\psi}}{v_x + \frac{w}{2} \dot{\psi}}\right) \\ \alpha_{rl} &= -\arctan\left(\frac{v_y - l_r \dot{\psi}}{v_x - \frac{w}{2} \dot{\psi}}\right) \\ \alpha_{rr} &= -\arctan\left(\frac{v_y - l_r \dot{\psi}}{v_x + \frac{w}{2} \dot{\psi}}\right) \end{aligned} \quad (5)$$

where δ is the front wheel steering angle. While longitudinal forces can generally be mapped from driver pedal inputs, for the specific PIT scenarios examined in this paper where drivers do not apply acceleration or deceleration inputs post-collision,

the longitudinal force is simplified to rolling resistance only:

$$F_{x,i} = (f_{r0} + f_{r1} \cdot v + f_{r2} \cdot v^2) \cdot F_{z,i} \quad (6)$$

where f_r is the rolling friction coefficient, f_{r0} denotes the basic rolling friction coefficient, f_{r1} and f_{r2} are velocity-dependent coefficients, and v is the vehicle's longitudinal speed. The vertical load distribution is crucial for accurate tire force prediction and accounts for both longitudinal and lateral load transfer effects caused by vehicle accelerations and body roll:

$$\begin{aligned} F_{z,fl} &= \frac{m}{2} \left(\frac{gl_r}{l_f + l_r} - \frac{h_{cog}a_x}{l_f + l_r} \right) - \frac{K_f}{K_f + K_r} \left(\frac{mh_{cog}a_y}{w} + \frac{m_sgh_{rc}\sin\phi}{w} \right) \\ F_{z,fr} &= \frac{m}{2} \left(\frac{gl_r}{l_f + l_r} - \frac{h_{cog}a_x}{l_f + l_r} \right) + \frac{K_f}{K_f + K_r} \left(\frac{mh_{cog}a_y}{w} + \frac{m_sgh_{rc}\sin\phi}{w} \right) \\ F_{z,rl} &= \frac{m}{2} \left(\frac{gl_f}{l_f + l_r} + \frac{h_{cog}a_x}{l_f + l_r} \right) - \frac{K_r}{K_f + K_r} \left(\frac{mh_{cog}a_y}{w} + \frac{m_sgh_{rc}\sin\phi}{w} \right) \\ F_{z,rr} &= \frac{m}{2} \left(\frac{gl_f}{l_f + l_r} + \frac{h_{cog}a_x}{l_f + l_r} \right) + \frac{K_r}{K_f + K_r} \left(\frac{mh_{cog}a_y}{w} + \frac{m_sgh_{rc}\sin\phi}{w} \right) \end{aligned} \quad (7)$$

where h_{cog} is the height of the center of gravity; a_x and a_y are the longitudinal and lateral accelerations; K_f and K_r are the front and rear cornering stiffnesses; w is the vehicle track width; m_s is the sprung mass; h_{rc} is the height of the roll center; and ϕ is the roll angle.

2.2. Momentum-conservation-based Collision Model

Vehicle collision processes are fundamentally momentum transfer phenomena, with their physical foundation rooted in the momentum conservation principle from Newton's laws of motion. In collision systems where external forces are negligible compared to impact forces, the total system momentum remains conserved before and after collision. This principle provides a theoretical foundation for deterministic impulse calculations, offering higher certainty compared to directly modeling the complex nonlinear processes of impact forces.

The momentum conservation-based collision modeling approach was first systematically developed by Brach et al. [21], treating colliding vehicles as rigid bodies and determining collision impulse through analysis of momentum changes before and after impact. Compared to direct impact force modeling, this approach offers advantages: impulse calculations primarily depend on relatively deterministic physical quantities such as collision geometry parameters, vehicle masses, and relative velocities, while complex factors like material nonlinearity and structural deformation have relatively minimal impact on impulse calculations.

Consider a two-dimensional planar problem of two-vehicle collision. A ground-fixed coordinate system XOY is established, where the X-axis is along the tangential direction of the road. As shown in Figure 2 Let the target vehicle be target vehicle and the striking vehicle be bullet vehicle. The pre-collision vehicle motion states are described by six parameters: $v_{tx}, v_{ty}, \dot{\psi}_t, v_{bx}, v_{by}, \dot{\psi}_b$, and the corresponding post-collision motion states are: $V_{tx}, V_{ty}, \dot{\Psi}_t, V_{bx}, V_{by}, \dot{\Psi}_b$.

According to the law of momentum conservation, the system's linear momentum is

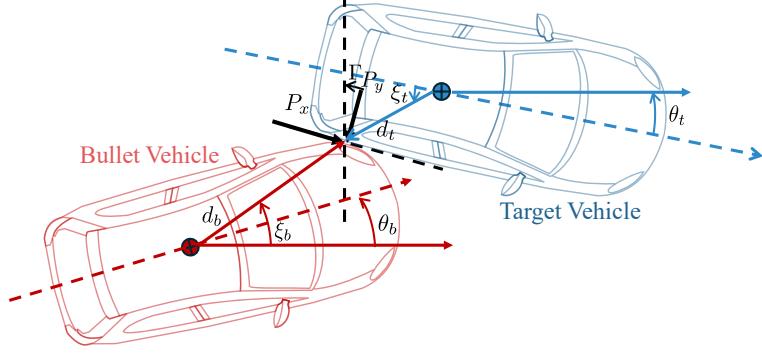


Figure 2. A view of two-vehicle collision model

conserved in both X and Y directions as

$$\begin{aligned} m_t \cdot (V_{tx} - v_{tx}) &= -m_b \cdot (V_{bx} - v_{bx}) = P_x \\ m_t \cdot (V_{ty} - v_{ty}) &= -m_b \cdot (V_{by} - v_{by}) = P_y \end{aligned} \quad (8)$$

where P_x and P_y are the collision impulses in the X and Y directions, respectively. For angular momentum conservation, with each vehicle's center of mass as the moment center:

$$\begin{aligned} I_{zzt}(\dot{\Psi}_t - \dot{\psi}_t) &= P_x d_t \sin(\theta_t + \xi_t) - P_y d_t \cos(\theta_t + \xi_t) \\ I_{zzb}(\dot{\Psi}_b - \dot{\psi}_b) &= P_x d_b \sin(\theta_b + \xi_b) - P_y d_b \cos(\theta_b + \xi_b) \end{aligned} \quad (9)$$

To solve for the impulses P_x and P_y , two additional physical constraint conditions are introduced. The restitution coefficient e is defined as the negative ratio of the relative separation velocity after collision to the relative approach velocity before collision:

$$e = -\frac{V_{bn} - V_{tn}}{v_{bn} - v_{tn}} \quad (10)$$

where the subscript n denotes normal component of the collision contact surface. The tangential friction coefficient μ is defined as the ratio of tangential impulse to normal impulse:

$$\mu = \frac{P_t}{P_n} \quad (11)$$

Decomposing the impulse into the normal-tangential coordinate system yields:

$$\mu \cdot (P_x \cos \Gamma + P_y \sin \Gamma) = P_y \cos \Gamma - P_x \sin \Gamma \quad (12)$$

By solving coupled equations 8-12, the collision impulses P_x and P_y can be determined. A key characteristic of this solution process is that the impulse calculation primarily depends on the following relatively deterministic physical parameters: geometric parameters (collision point location, vehicle dimensions, collision angle, etc.) can be accurately determined through collision geometry analysis; inertial parameters (vehicle mass, moment of inertia, etc.) possess high certainty; kinematic parameters (pre-

collision velocity states) can be obtained through trajectory analysis or sensor data; and material parameters (coefficient of restitution and friction coefficient), while exhibiting some variability, have relatively limited impact and can be determined through empirical data. In contrast, directly modeling the impact force process requires considering numerous uncertain factors such as material nonlinearity, structural deformation, and contact stiffness variations, making impulse calculation inherently more deterministic. The uncertainties in practical applications will be implicitly incorporated in the GMM impact forces training discussed below. According to the impulse-momentum theorem, impulse equals the integral of force over time

$$\begin{aligned} P_x &= \int_0^{\Delta t} F_x(t) dt \\ P_y &= \int_0^{\Delta t} F_y(t) dt \end{aligned} \quad (13)$$

Based on the determined impulses P_x and P_y , simplified force-time models can be employed to represent the temporal distribution of impact forces. Here we used a GMM model.

3. Impact Force Modelling Considering Uncertainties

3.1. FEA Environment Setting

To establish a reliable foundation for the stochastic impact force model, this paper conducted systematic simulation research using LS-DYNA R12.0 FEA software. LS-DYNA is one of the most advanced explicit nonlinear FEA software packages internationally, with extensive applications and validation in automotive crash simulation. The software employs explicit time integration algorithms that effectively handle complex problems including large deformation, material nonlinearity, and contact collisions, making it particularly suitable for high-speed impact and collision analysis.

The FEA scenarios designed in this paper are shown in the Figure 3, (a) is the grid diagram and (b) is the diagram of the FEA simulation collision, where a vehicle approaches from the rear-quarter and collides with the target vehicle at the rear axle body location. The target vehicle is set to 2000kg, while the bullet vehicle considers three different mass conditions: 1500kg/2000kg/2500kg. High-fidelity vehicle models are employed, with two different vehicle models selected. Each simulation adopts rigorous numerical settings to ensure computational accuracy and stability. Time integration uses the central difference scheme with a time step of 1×10^{-6} seconds, which can capture high-frequency phenomena during collision while maintaining numerical stability. Spatial discretization employs mixed hexahedral and tetrahedral meshes, with mesh sizes in critical regions controlled at 2-5mm to accurately capture stress gradients and deformation details.

As shown in Figure 4, 6 cases with different parameter distributions are sampled from the 100 cases. Figure 4 (a) presents the different parameter distributions, revealing that the values of P_x/P_y under different conditions can be essentially classified by the collision angle. As shown in Figure 4(b), since the collision scenarios are primarily side-impact collisions, the actual P_y is generally larger than P_x .

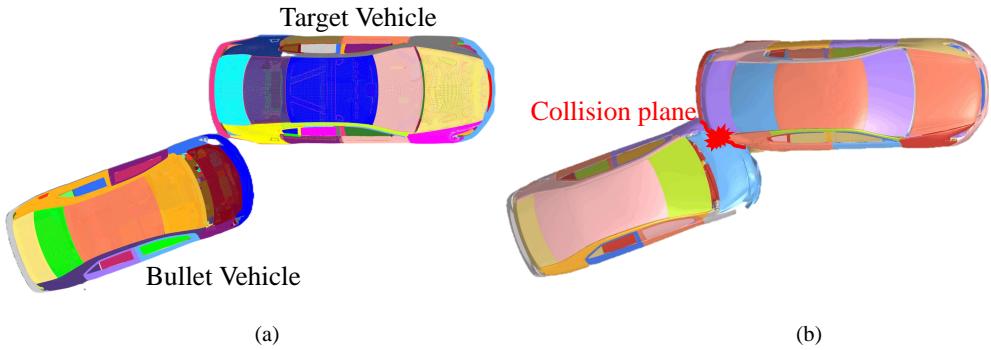


Figure 3. The PIT scenario setting in FEA simulation: (a) before and (b) after collision

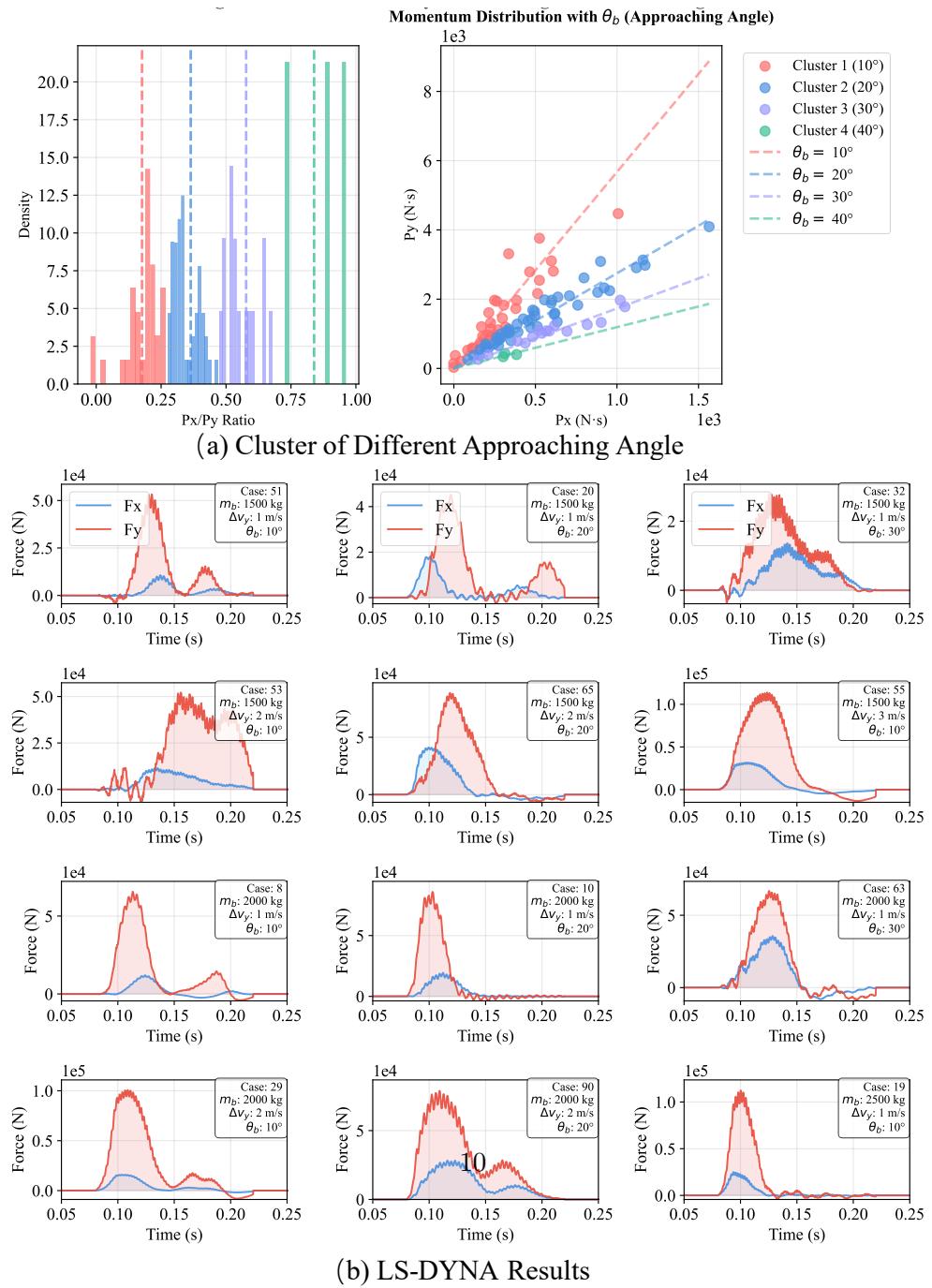


Figure 4. Twelve sample results of FEA force calculated via a_x and a_y

3.2. PINN Based Impact force Modeling Considering Uncertainties

Traditional approaches to impact force modeling typically rely on preset geometric profiles (e.g., triangular or rectangular pulses). While computationally efficient, such methods exhibit notable limitations in handling complex collision scenarios: (1) fixed geometric configurations fail to accommodate the inherent variations in force time-histories under differing collision conditions; (2) these methods cannot accurately predict vehicle state transitions under collision when serving as inputs for subsequent collision dynamics analyses; (3) quantification of uncertainty in prediction outcomes remains challenging.

To address these challenges, this paper proposes an innovative modeling framework that abandons traditional basis function methods, adopting a hybrid strategy integrating Physics-Informed Neural Networks (PINN) and Gaussian Mixture Models (GMMs). The core idea is to enable the neural network to autonomously learn the temporal distribution patterns of impact forces under strict physical constraints.

The complete PINN-GMM modeling framework can be expressed as

$$\mathbf{F}_{\text{collision}}(t; \boldsymbol{\theta}, \mathbf{W}) = \mathcal{NN}_{\text{PINN-GMM}}(\boldsymbol{\theta}, t; \mathbf{W}) \quad (14)$$

where $\mathbf{F}_{\text{collision}}(t) = [F_x(t), F_y(t)]^T$ denotes the impact force vector at time t , $\boldsymbol{\theta}$ represents the collision scenario feature vector, $\boldsymbol{\theta} = [m_t, m_b, v_{ty}, v_{by}, v_{tx}, v_{bx}, \theta_b]$, \mathbf{W} stands for the set of neural network weight parameters, and $\mathcal{NN}_{\text{PINN-GMM}}$ indicates the neural network mapping function integrated with physical constraints.

The key innovation of this framework resides in embedding physical laws as hard constraints into the network training process to ensure the physical rationality of prediction results, while quantifying prediction uncertainty through the GMM component. Compared with traditional methods, this framework exhibits three distinct features: (1) integration of physical laws as hard constraints; (2) probabilistic modeling of time-varying uncertainty; (3) adaptive pulse shape learning.

3.2.1. Physics Constraints Design

To ensure model outputs comply with physical laws, this paper designs a multi-level physical constraint system. The first level is the momentum conservation constraint, where impulse during collision processes must satisfy the law of momentum conservation. In the x and y directions respectively:

$$\begin{aligned} \int_{t_{\text{start}}}^{t_{\text{end}}} F_x(t) dt &= P_x = m_{\text{eff}} \Delta v_x \\ \int_{t_{\text{start}}}^{t_{\text{end}}} F_y(t) dt &= P_y = m_{\text{eff}} \Delta v_y \end{aligned} \quad (15)$$

where $m_{\text{eff}} = \frac{m_1 m_2}{m_1 + m_2}$ is the effective mass, and Δv_x and Δv_y are the relative velocity changes before and after collision. The corresponding impulse conservation loss function is:

$$\mathcal{L}_{\text{impulse}} = \lambda_{\text{imp}} \left[\left(\int_{t_{\text{start}}}^{t_{\text{end}}} F_x(t) dt - P_x \right)^2 + \left(\int_{t_{\text{start}}}^{t_{\text{end}}} F_y(t) dt - P_y \right)^2 \right] \quad (16)$$

The second level is the energy consistency constraint, where energy constraints are introduced to ensure energy conversion during collision processes complies with physical laws:

$$\mathcal{L}_{\text{energy}} = \lambda_{\text{eng}} \left| \int_{t_{\text{start}}}^{t_{\text{end}}} \mathbf{F}(t) \cdot \mathbf{v}_{\text{rel}}(t) dt - E_{\text{dissipated}} \right|^2 \quad (17)$$

where the dissipated energy $E_{\text{dissipated}} = \frac{1}{2}m_{\text{eff}}V_{\text{rel}}^2(1-e^2)$, e is the coefficient of restitution which is set 0.5 as normal semi plastic deformation, and V_{rel} is the magnitude of relative velocity before collision.

3.2.2. Neural Network Architecture and Time-Varying Probabilistic Modeling

As shown in the Figure 5, the network adopts an encoder-decoder architecture, primarily comprising four core components with deep integration. The multi-scale feature encoder processes the input collision scenario features θ , employing a multi-branch structure to extract feature representations at different scales: $\mathbf{h}_{\text{feat}} = \text{Swish}(\mathbf{W}_3 \cdot \text{Swish}(\mathbf{W}_2 \cdot \text{Swish}(\mathbf{W}_1 \cdot \theta)))$. The sinusoidal positional time encoder employs sinusoidal positional encoding to process temporal information, enhancing the network's understanding of temporal patterns: $\text{PE}(t, 2i) = \sin\left(\frac{t}{10000^{2i/d}}\right)$, $\text{PE}(t, 2i+1) = \cos\left(\frac{t}{10000^{2i/d}}\right)$. This encoding approach enables the network to better capture the periodicity and temporal dependencies of impact forces. The temporal attention mechanism introduces self-attention to enhance the network's focus on critical temporal nodes: $\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$. The physical constraint integration layer serves as the output layer design, ensuring final outputs satisfy physical laws.

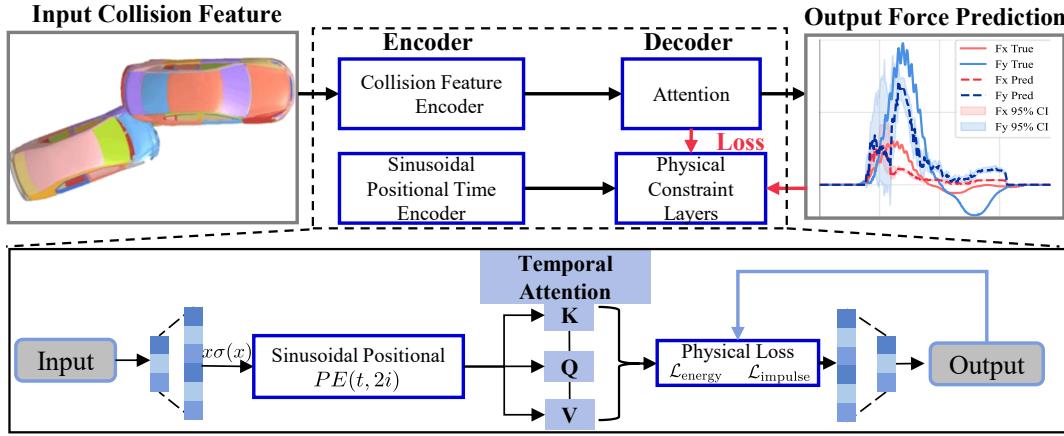


Figure 5. Scheme of PINN-GMM neural network framework

The network employs the Swish activation function: $\text{Swish}(x) = x \cdot \sigma(x) = \frac{x}{1+e^{-x}}$. The Swish function exhibits excellent numerical stability and gradient propagation characteristics, particularly suitable for training deep physical networks. Compared to traditional ReLU or Tanh functions, Swish demonstrates superior convergence performance in handling physics-constrained optimization. The network output layer is designed to include three main components: instantaneous impact force prediction

$\mathbf{F}(t) = [F_x(t), F_y(t)]^T$, adaptive collision duration $\Delta t = \sigma(z_{\Delta t}) \times \Delta t_{\max}$, and dynamic GMM parameters $\Phi = [\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}]$.

Finally, this paper designs a time-varying GMM where parameters are modeled as continuous functions of time and scenario features:

$$p(\mathbf{F}(t)|\boldsymbol{\theta}) = \sum_{k=1}^K \pi_k(t, \boldsymbol{\theta}) \mathcal{N}(\mathbf{F}(t)|\boldsymbol{\mu}_k(t, \boldsymbol{\theta}), \boldsymbol{\Sigma}_k(t, \boldsymbol{\theta})) \quad (18)$$

The time-varying weight function $\pi_k(t, \boldsymbol{\theta}) = \frac{\exp(w_k(t, \boldsymbol{\theta}))}{\sum_{j=1}^K \exp(w_j(t, \boldsymbol{\theta}))}$; the time-varying mean function $\boldsymbol{\mu}_k(t, \boldsymbol{\theta}) = \mathbf{F}_{\text{base}}(t, \boldsymbol{\theta}) + \boldsymbol{\delta}_k(t, \boldsymbol{\theta})$ and the time-varying covariance function $\boldsymbol{\Sigma}_k(t, \boldsymbol{\theta}) = \text{diag}(\sigma_{\min}^2 + \exp(\mathbf{s}_k(t, \boldsymbol{\theta})))$. This design enables the model to provide dynamically adjusted uncertainty estimates at different time instants and under different collision scenarios, offering significant advantages over traditional static uncertainty modeling approaches.

3.3. Model Analysis Based on FEA Dataset

Numerical simulation data from 100 unique collision scenarios are utilized, with the training set and test set split at a 3:1 ratio. The network architecture adopts an 8-layer fully connected network, featuring hidden layer neuron counts of [256, 512, 512, 256, 128, 64, 32, 16]. Training parameters are set as follows: learning rate of 1e-3, batch size of 32, and 500 training epochs. The PINN-GMM-based prediction model is trained on a computing platform equipped with an NVIDIA GeForce RTX 4070 graphics card. As illustrated in the Figure 6, the loss function of the trained model converges and stabilizes at approximately 300 epochs, demonstrating favorable convergence characteristics. Both training loss and validation loss decrease synchronously, with no significant overfitting observed.

As shown in Figure 6, all cases in the test set exhibit favorable overall predictions and uncertainty distributions, particularly regarding the trend and magnitude of force peaks. Thus, the proposed model can be deemed to achieve satisfactory performance. The RMSE values of F_x and F_y across all cases are 2.1e+03 and 3.9e+03, respectively, indicating that the model attains a relatively high level of prediction accuracy.

4. Adaptive PINN Vehicle Dynamics Response Modeling Framework

Traditional PINN approaches face a fundamental dilemma: rigid physical constraints can completely suppress data-driven learning, effectively reducing neural networks to complex implementations of existing physical models. This paper addresses this critical challenge through a Adaptive Physics Integration mechanism that maintains physical consistency while preserving the network's learning capability.

4.1. PINN Network Architecture Design

Following the impact force prediction from the first network, this section designs a second Adaptive physics-informed neural network to model vehicle dynamic responses under uncertain impact forces. Unlike traditional methods that apply physical constraints as post-processing corrections, this framework directly embeds adaptive phys-

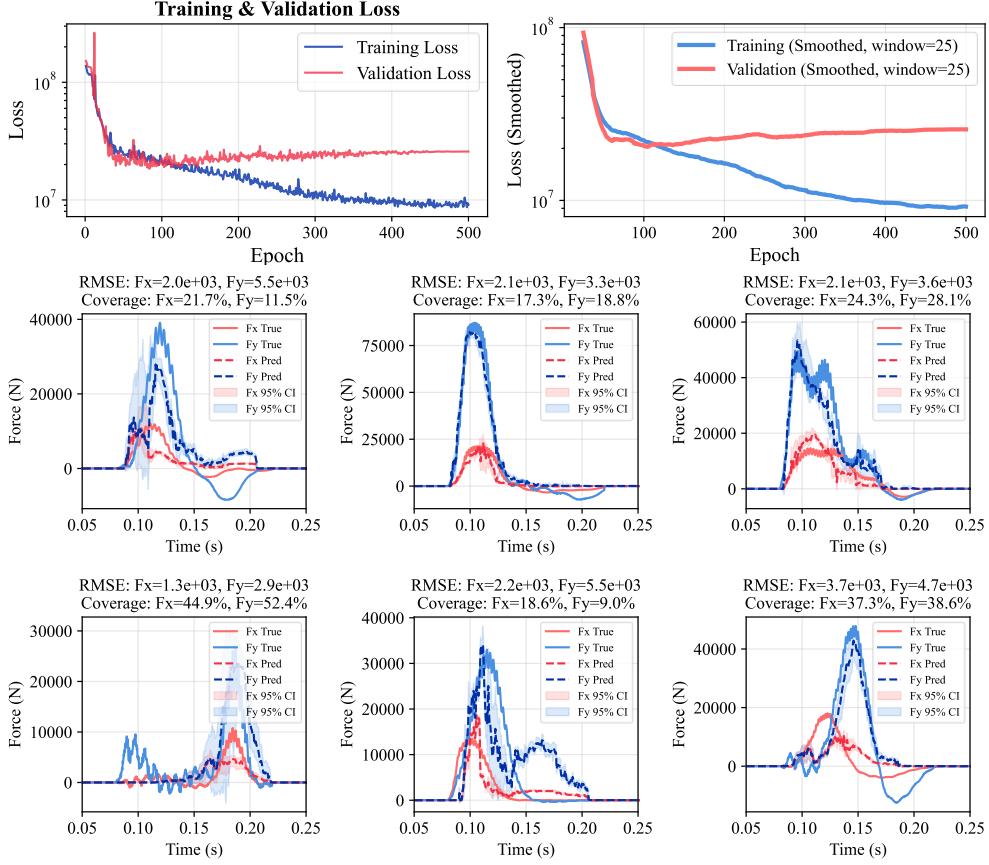


Figure 6. PINN-GMM losses and six samples of force prediction results

ical constraints into the network architecture, effectively addressing the core issue of constraint-induced learning inhibition.

The Adaptive PINN can be expressed as:

$$\hat{\mathbf{x}}(t+1) = \mathcal{NN}_{\text{Adaptive-PINN}}(\mathbf{F}_{\text{collision}}(t), \boldsymbol{\Theta}_{\text{vehicle}}, t; \mathbf{W}_{\text{adaptive}}) \quad (19)$$

where $\hat{\mathbf{x}}(t+1) = [v_x(t+1), v_y(t+1), \dot{\psi}(t+1), \dot{\phi}(t+1)]^T$ is the predicted vehicle state vector at time $t+1$, $\mathbf{F}_{\text{collision}}(t) = [F_x(t), F_y(t)]^T$ is the impact force vector from the first PINN network, $\boldsymbol{\Theta}_{\text{vehicle}}$ contains vehicle parameters (mass m , moment of inertia $I_{zz}, I_{xx,s}, I_{xz}$, geometric parameters l_f, l_r, t_w , and tire parameters), and $\mathbf{W}_{\text{adaptive}}$ represents the adaptive neural network parameters, whose constraint sensitivity is dynamically adjusted based on training progress and data quality.

The vehicle dynamics PINN employs a specialized encoder-decoder architecture optimized for multi-physics constraint integration. As shown in Figure 7

First, the input force $\mathbf{F}_{\text{collision}}(t)$ is encoded by the Multi-Scale Feature Encoder module, where the vehicle parameter vector $\boldsymbol{\Theta}_{\text{vehicle}}$ is densely embedded to create a feature representation of the encoded physical properties:

$$\mathbf{h}_{\text{vehicle}} = \text{Swish}(\mathbf{W}_3 \cdot \text{Swish}(\mathbf{W}_2 \cdot \text{Swish}(\mathbf{W}_1 \cdot \boldsymbol{\Theta}_{\text{vehicle}}))) \quad (20)$$

where $\mathbf{W}_1, \mathbf{W}_2, \mathbf{W}_3$ are learnable weight matrices that progressively transform vehicle

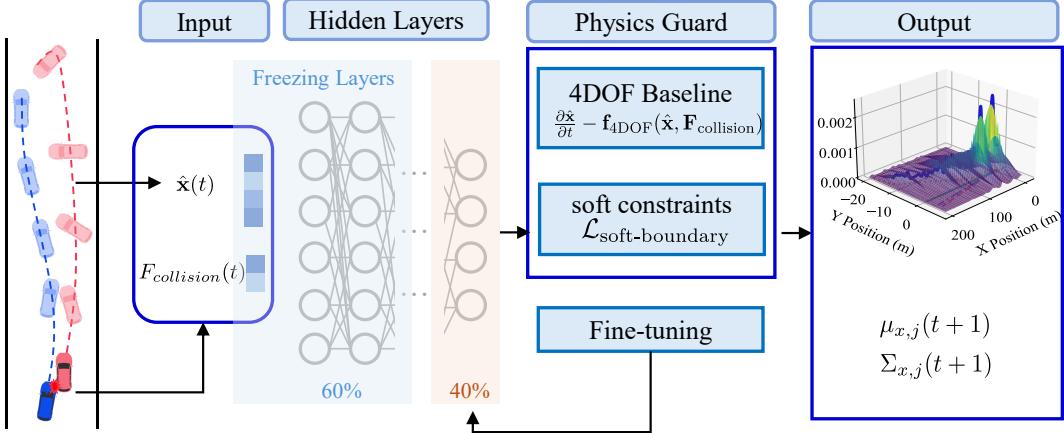


Figure 7. Scheme of A-PINN architecture for post-impact vehicle dynamics prediction

parameters into high-dimensional feature representations.

The decoder then integrates temporal and vehicle features via a cross-attention mechanism, followed by an output layer of physical constraints:

$$\mathbf{h}_{\text{fused}} = \text{CrossAttention}(\mathbf{h}_{\text{combined}}, \mathbf{h}_{\text{vehicle}}) \quad (21)$$

Traditional PINN methods use fixed constraint weights, which can easily lead to excessive dominance of physical constraints or insufficient data learning. This paper proposes adaptive constraint weights based on training progress:

$$\lambda_{\text{physics}}(t, \text{epoch}) = \lambda_{\min} + (\lambda_{\max} - \lambda_{\min}) \cdot \sigma \left(\frac{\text{epoch} - t_0}{\tau} \right) \cdot \text{DataQuality}(t) \quad (22)$$

where $\lambda_{\min} = 0.1$ and $\lambda_{\max} = 10.0$ are the minimum and maximum constraint weights, $\sigma(\cdot)$ is the sigmoid activation function, $t_0 = 300$ indicates the transition round at which the constraint strength begins to decay, $\tau = 100$ controls the smoothness of the transition, and $\text{DataQuality}(t)$ measures the local data reliability at time t .

Data quality assessment is calculated using nearest neighbor distance:

$$\text{DataQuality}(t) = \exp \left(-\frac{d_{\text{nearest}}(t)}{h_{\text{bandwidth}}} \right) \quad (23)$$

where $d_{\text{nearest}}(t)$ is the euclidean distance to the nearest training sample in the temporal feature space at time t , and $h_{\text{bandwidth}} = 0.1$ is the bandwidth parameter that controls the locality of data quality assessment.

The CrossAttention mechanism allows the network to selectively focus on relevant temporal features based on vehicle characteristics, while PhysicsGuard ensures outputs remain within physically meaningful bounds.

4.2. Adaptive Physics Guard

To address the fundamental challenge of ensuring physical consistency while enabling the network to learn systematic model discrepancies from limited real data, an Adaptive Physical Guard layer is introduced as the final processing stage. This layer serves as a critical safeguard that prevents physically impossible predictions while preserving the network's ability to capture unmodeled dynamics.

To prevent the vanishing gradients of the model, the physical protection layer is enforced through soft constraints:

$$\mathcal{L}_{\text{soft-boundary}} = \lambda_{\text{boundary}} \sum_i \text{SoftPlus} \left(\frac{|\hat{x}_i| - b_i}{s_i} \right)^2 \quad (24)$$

where $\lambda_{\text{boundary}} = 1.0$ is the boundary constraint weight, \hat{x}_i represents the i -th predicted state variable, b_i is the physical boundary of the state variable \hat{x}_i , s_i is the learnable smooth parameter, and $\text{SoftPlus}(\cdot) = \ln(1 + \exp(\cdot))$ ensures the differentiability of the entire constraint region.

To prevent physical constraints from completely suppressing data learning, implement a dynamic balance monitoring system:

$$\text{Balance Ratio}(t) = \frac{\mathcal{L}_{\text{data}}(t)}{\mathcal{L}_{\text{data}}(t) + \mathcal{L}_{\text{physics}}(t)} \quad (25)$$

where $\mathcal{L}_{\text{data}}(t)$ is the data fit loss at time t , $\mathcal{L}_{\text{physics}}(t)$ is the physical constraint loss at time t , Balance Ratio quantifies the relative dominance of data versus physics in the current loss landscape, with values close to 0 indicating physics dominance and close to 1 indicating data dominance.

When physical constraints are overly dominant, the system automatically adjusts the constraint strength:

$$\lambda_{\text{physics}} = \lambda_{\text{physics}} \cdot \max(0.1, \text{Balance Ratio}) \quad (26)$$

where λ_{physics} is the current constraint weight, and a factor of 0.1 ensures that a minimum constraint strength is maintained even if data dominance is detected to maintain basic physical consistency. Physical residual learning can be expressed as:

$$\mathcal{L}_{\text{physics}} = \lambda_{\text{physics}}(t) \left\| \frac{\partial \hat{\mathbf{x}}}{\partial t} - \mathbf{f}_{\text{4DOF}}(\hat{\mathbf{x}}, \mathbf{F}_{\text{collision}}) \right\|^2 \quad (27)$$

where $\hat{\mathbf{x}}$ is the predicted vehicle state vector, $\mathbf{f}_{\text{4DOF}}(\cdot)$ represents the baseline 4DOF vehicle dynamics function from the equation 2, $\mathbf{F}_{\text{collision}}$ is the crash force input, and $\lambda_{\text{physics}}(t)$ is the time-varying physical constraint weight.

4.3. Vehicle State GMM Output and Uncertainty Modeling

Uncertainty in the second PINN model is primarily due to: (1) impact force uncertainty from the first network $\mathbf{F}_{\text{collision}}(t)$; (2) vehicle parameter measurement error Θ_{vehicle} ; (3) unmodeled dynamics of the baseline 4DOF model; (4) model uncertainty due to lim-

ited training data; and (5) uncertainty due to adaptive physical constraint weight adjustments.

The network output layer predicts the GMM parameters of the vehicle state vector $\hat{\mathbf{x}}(t+1) = [v_x(t+1), v_y(t+1), \dot{\psi}(t+1), \dot{\phi}(t+1)]^T$, where $v_x(t+1)$ and $v_y(t+1)$ are the longitudinal and lateral speeds, respectively, $\dot{\psi}(t+1)$ is the yaw rate, and $\dot{\phi}(t+1)$ is the roll rate:

$$p(\hat{\mathbf{x}}(t+1)|\mathbf{F}_{\text{collision}}(t), \Theta_{\text{vehicle}}) = \sum_{j=1}^J \omega_j(t) \mathcal{N}(\hat{\mathbf{x}}(t+1)|\boldsymbol{\mu}_{x,j}(t+1), \boldsymbol{\Sigma}_{x,j}(t+1)) \quad (28)$$

where J is the number of mixed components, $\omega_j(t)$ is the time-varying weight of the j th component, $\boldsymbol{\mu}_{x,j}(t+1) \in \mathbb{R}^4$ is the mean vector of the j th component, and $\boldsymbol{\Sigma}_{x,j}(t+1) \in \mathbb{R}^{4 \times 4}$ is the covariance matrix of the j th component.

The original network outputs $z_{\omega,j}, \mathbf{z}_{\mu,j}, \mathbf{z}_{\sigma,j}$ are transformed as follows to ensure a valid probability distribution:

$$\begin{aligned} \omega_j(t) &= \frac{\exp(z_{\omega,j})}{\sum_{k=1}^J \exp(z_{\omega,k})} \\ \boldsymbol{\mu}_{x,j}(t+1) &= \mathbf{f}_{\text{4DOF}}(\hat{\mathbf{x}}(t), \mathbf{F}_{\text{collision}}(t)) + \tanh(\mathbf{z}_{\mu,j}) \odot \boldsymbol{\sigma}_{\text{bound}} \\ \boldsymbol{\Sigma}_{x,j}(t+1) &= \text{diag}(\sigma_{\min}^2 + \exp(\mathbf{z}_{\sigma,j})) \end{aligned} \quad (29)$$

where the first equation ensures that the sum of weights is 1 and non-negative through the softmax function; The second equation centers on the 4DOF physics model prediction and limits the learning bias to $[-1, 1]$ via the $\tanh(\cdot)$ function, where $\boldsymbol{\sigma}_{\text{bound}} = [5.0, 3.0, 1.0, 0.8]^T$ is the maximum allowable deviation of $[v_x, v_y, \dot{\psi}, \dot{\phi}]^T$, respectively; the third equation ensures positive definite covariance matrix by diagonalization and exponential function, $\sigma_{\min} = 0.01$ prevents numerical instability.

Training objectives combine data fitting and physical consistency, ensuring that each GMM component satisfies the physical laws:

$$\mathcal{L}_{\text{GMM-physics}} = \lambda_{\text{phys-gmm}} \mathcal{L}_{\text{physics-gmm}} + \lambda_{\text{consistency}} \mathcal{L}_{\text{consistency}} \quad (30)$$

Physical consistency loss ensures that the mean of each GMM component follows the vehicle dynamics equation:

$$\mathcal{L}_{\text{physics-gmm}} = \sum_{j=1}^J \omega_j(t) \left\| \frac{\partial \boldsymbol{\mu}_{x,j}}{\partial t} - \mathbf{f}_{\text{4DOF}}(\boldsymbol{\mu}_{x,j}, \mathbf{F}_{\text{collision}}) - \boldsymbol{\delta}_{\text{learned},j} \right\|^2 \quad (31)$$

where $\frac{\partial \boldsymbol{\mu}_{x,j}}{\partial t}$ is the time derivative of the mean of the j component, $\mathbf{f}_{\text{4DOF}}(\cdot)$ is the 4DOF vehicle dynamics function (from the equation 2), $\boldsymbol{\delta}_{\text{learned},j}$ is the learnable physics correction for the j component to capture unmodeled dynamics.

Preventing GMM component over-clustering due to diversity consistency loss:

$$\mathcal{L}_{\text{consistency}} = \sum_{j=1}^{J-1} \sum_{k=j+1}^J \exp \left(-\frac{\|\boldsymbol{\mu}_{x,j} - \boldsymbol{\mu}_{x,k}\|^2}{2\tau^2} \right) \quad (32)$$

where $\tau = 2.0$ is the bandwidth parameter that controls the degree of separation between components. Losses increase when two components are too close together, encouraging diversity.

4.4. Uncertainty Propagation and Trajectory Distribution Modeling

Vehicle state uncertainty is propagated to the global trajectory coordinate system by traceless transformation. For each GMM component j , generate $2n + 1$ sigma points from its mean $\boldsymbol{\mu}_{x,j}(t+1)$ and covariance $\boldsymbol{\Sigma}_{x,j}(t+1)$ (where $n = 4$ is the state dimension):

$$\begin{aligned}\boldsymbol{\chi}_{j,0} &= \boldsymbol{\mu}_{x,j}(t+1) \\ \boldsymbol{\chi}_{j,i} &= \boldsymbol{\mu}_{x,j}(t+1) + \left(\sqrt{(n+\lambda)\boldsymbol{\Sigma}_{x,j}(t+1)} \right)_i, \quad i = 1, \dots, n \\ \boldsymbol{\chi}_{j,i} &= \boldsymbol{\mu}_{x,j}(t+1) - \left(\sqrt{(n+\lambda)\boldsymbol{\Sigma}_{x,j}(t+1)} \right)_{i-n}, \quad i = n+1, \dots, 2n\end{aligned}\tag{33}$$

where $\lambda = \alpha^2(n + \kappa) - n$ is the scaling parameter, $\alpha = 0.001$ controls the sigma point distribution, $\kappa = 0$ is the secondary scaling parameter, $(\sqrt{(n+\lambda)\boldsymbol{\Sigma}_{x,j}(t+1)})_i$ represents the i column of the square root of the matrix.

Each sigma point is transformed into the global coordinate system through kinematic equations:

$$\mathcal{Y}_{j,i} = \begin{bmatrix} X(t) + v_{x,i} \cos \psi(t) \Delta t - v_{y,i} \sin \psi(t) \Delta t \\ Y(t) + v_{x,i} \sin \psi(t) \Delta t + v_{y,i} \cos \psi(t) \Delta t \end{bmatrix}\tag{34}$$

where $X(t), Y(t)$ is the current global position, $\psi(t)$ is the current heading angle, $v_{x,i}, v_{y,i}$ is the velocity component of the i th sigma point, and Δt is the time step.

The parameter distribution of the transformed trajectory is calculated through weighted statistics:

$$\begin{aligned}\boldsymbol{\mu}_{\text{traj},j}(t+1) &= \sum_{i=0}^{2n} W_i^{(m)} \mathcal{Y}_{j,i} \\ \boldsymbol{\Sigma}_{\text{traj},j}(t+1) &= \sum_{i=0}^{2n} W_i^{(c)} (\mathcal{Y}_{j,i} - \boldsymbol{\mu}_{\text{traj},j}(t+1)) (\mathcal{Y}_{j,i} - \boldsymbol{\mu}_{\text{traj},j}(t+1))^T\end{aligned}\tag{35}$$

where the weights are defined as: $W_0^{(m)} = \lambda/(n+\lambda)$, $W_i^{(m)} = 1/(2(n+\lambda))$ ($i = 1, \dots, 2n$), $W_0^{(c)} = \lambda/(n+\lambda) + (1 - \alpha^2 + \beta)$, $W_i^{(c)} = 1/(2(n+\lambda))$ ($i = 1, \dots, 2n$), $\beta = 2$ is the higher-order moment correction parameter.

The final global trajectory GMM distribution combines all transformation components:

$$p(X(t+1), Y(t+1) | \mathbf{F}_{\text{collision}}(t)) = \sum_{j=1}^J \omega_j(t) \mathcal{N}([X(t+1), Y(t+1)]^T; \boldsymbol{\mu}_{\text{traj},j}(t+1), \boldsymbol{\Sigma}_{\text{traj},j}(t+1))\tag{36}$$

4.5. Training Strategy

The complete training loss integrates multiple physical and statistical objectives to ensure both accuracy and physical consistency:

$$\begin{aligned}\mathcal{L}_{\text{total}} &= \mathcal{L}_{\text{data}} + \lambda_{\text{phys}} \mathcal{L}_{\text{physics-GMM}} + \lambda_{\text{traj}} \mathcal{L}_{\text{trajectory}} + \lambda_{\text{smooth}} \mathcal{L}_{\text{smoothness}} + \lambda_{\text{bound}} \mathcal{L}_{\text{soft-boundary}} \\ \mathcal{L}_{\text{data}} &= \|\hat{\mathbf{x}}(t+1) - \mathbf{x}_{\text{true}}(t+1)\|^2 \\ \mathcal{L}_{\text{smoothness}} &= \left\| \frac{\partial^2 \hat{\mathbf{x}}}{\partial t^2} \right\|^2\end{aligned}\tag{37}$$

where the weight coefficients λ_{\bullet} control the relative importance of each loss component during training. \mathbf{x}_{true} is the true data. \mathbf{x}_0 is the known initial vehicle state at collision onset time t_0 . The PINN model proposed in this paper adopts a pre-training and fine-tune approach, and the physical prior knowledge or data knowledge based on true values used by all the compared Models are relatively fair.

The initial pre-training phase focuses on embedding basic physical constraints, with training more considering physics loss to enable the network to learn vehicle dynamic physical laws. 100 force data obtained from FEA are input into a 4DOF model for basic simulation, and the output of the 4DOF model is used as the basis for PINN pre-training, where the trajectory sampling frequency is 100Hz and consistent with the data frequency used for fine-tuning with Carsim data.

The fine-tune leverages the physically consistent foundation established during pre-training, achieving optimization performance with less training data through selective layer freezing and mixed loss optimization. In this paper, 5 and 20 real Carsim simulation trajectories which contain 2000 and 8000 samples are used for training during the fine-tune.

To prevent overfitting and preserve learned physics while adapting to limited real data, a layer freezing approach is employed:

$$\mathbf{W}_{\text{fine-tune}} = \{\mathbf{W}_{\text{frozen}}, \mathbf{W}_{\text{active}}\}\tag{38}$$

where $\mathbf{W}_{\text{frozen}}$ represents the frozen parameters from pre-training that encode fundamental physics relationships, and $\mathbf{W}_{\text{active}}$ represents the active parameters that adapt to specific collision scenarios. The number of frozen layers N_{frozen} is determined as $N_{\text{frozen}} = \lfloor 0.6 \times N_{\text{total}} \rfloor$. Ensuring that 60% of layers remain frozen to preserve physics knowledge while allowing 40% to adapt to trajectory-specific patterns.

4.6. Baseline Models

To validate the proposed PINN method, a comprehensive comparison was conducted with two baseline methods. Notably, the comparison between different models is based on the premise that the prior physical knowledge and true data are completely consistent, where 20% of the data obtained from the dataset was selected as known data. A conventional numerical integration approach using the same 4DOF vehicle dynamics equations without neural network approximation serves as the 4DOF baseline model:

$$\mathbf{x}(t + \Delta t) = \mathbf{x}(t) + \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), \mathbf{F}_{\text{collision}}(t)) \cdot \Delta t\tag{39}$$

where $\mathbf{f}(\cdot)$ represents the system dynamics function derived from the 4DOF equations, and $\mathbf{u}(t)$ represents control inputs (steering, throttle, braking). For basic and easily obtainable vehicle parameters, they are directly exported from the Carsim model, while for tire parameters K_i, C_i that have a significant dynamic impact on the vehicle under extreme conditions, particle swarm optimization (PSO) is used to obtain the best matching parameters from 8000 prior sampling data.

Another baseline model is a standard feedforward neural network without physical constraints, trained purely based on input-output mapping, removing the Physics Guard layer in PINN and only considering data loss during training, representing a purely empirical method for fitting the data:

$$\hat{\mathbf{x}}_{\text{data}}(t+1) = \mathcal{NN}_{\text{data}}(\mathbf{F}_{\text{collision}}(t), \Theta_{\text{vehicle}}, t; \mathbf{W}_{\text{data}}) \quad (40)$$

4.7. Carsim-FEA Force Dataset Combined Simulation Analysis

By inputting transient impact forces obtained from FEA software into the Carsim model, the corresponding vehicle trajectory is obtained. First, the vehicle's dynamic characteristics under different forces are analyzed based on the global X-Y trajectory. A clustering method for vehicle dynamics simulation data based on force integrals (P_x, P_y) is employed. This method, grounded in physical principles, uses P_x, P_y as the core classification criteria. First, four basic force integral features are extracted for each simulation case: X- and Y-direction force integrals (I_{fx}, I_{fy}), total impulse ($I_{total} = \int_0^T |F(t)|dt$), and force statistics (maximum value, mean, and standard deviation). The clustering process utilizes an improved K-means algorithm, adaptively determining the optimal number of clusters using the silhouette coefficient. 20 random initializations and a maximum of 500 iterations are used to ensure result stability. Finally, post-processing merges small clusters ($\leq 5\%$) to ensure the physical plausibility of the clustering results.

As shown in Figure 8, cluster analysis divides the 100 Carsim simulation cases into 4 categories based on impact characteristic patterns, and different clusters exhibit significant differences in X-Y trajectory features. Among them, the vehicles in C0-C4 experience gradually increasing impact, where most trajectories in C0 and C1 show that the vehicles do not spin significantly under impact force, whereas all vehicles in C2 and C3 exhibit clear spinning, leading to loss of control in the latter half of the vehicle trajectories. Since the tire characteristics of the vehicle in these two scenarios are quite different, and tire characteristics are precisely the hard-to-model parts in vehicle dynamics, the following section will present the specific performance of different models in these two types of cases.

Based on the performance comparison analysis in the Table 1, different models exhibit distinct characteristics and trade-offs in terms of tracking accuracy, computational efficiency, and physical consistency. The NN-WITHOUT-PINN model ranks first overall with the lowest average error (8.25m) and demonstrates superior performance in most RMSE metrics including X, Y positions and vx velocity, while maintaining the highest computational efficiency (7.45 ± 1.08 ms/step). However, this purely data-driven model lacks physical constraints, potentially leading to predictions that violate fundamental vehicle dynamics principles and pose safety risks in complex scenarios due to unrealistic outputs. The traditional 4DOF model, despite its fast computation speed (3.21 ± 2.03 ms/step), suffers from poor adaptability due to fixed parameter settings and exhibits the worst performance across all accuracy metrics, particularly

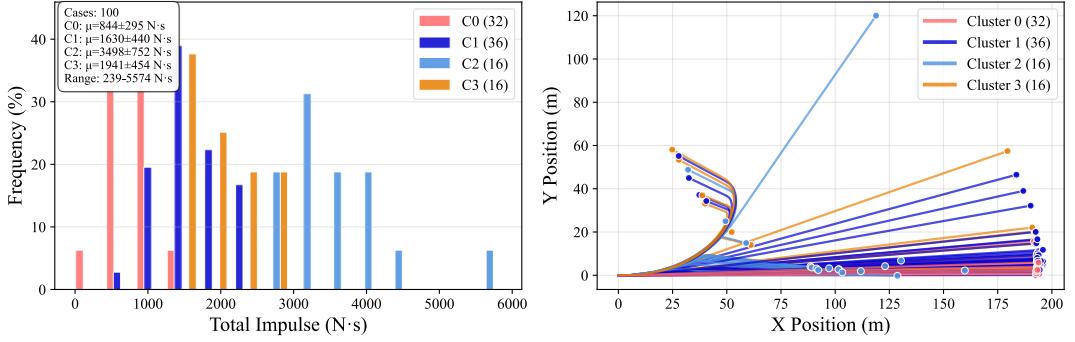


Figure 8. Comparing PINN-based vehicle state prediction with baselines under impact force with uncertainties

with X-direction position errors reaching 37.75 ± 29.47 m, failing to meet high-precision tracking requirements.

In contrast, the PINN-GMM-20TRAC model demonstrates the best comprehensive performance and practical value. While its average error (10.47 m) is slightly higher than the pure neural network approach, it excels in critical parameters for vehicle stability control: v_y velocity prediction error of 1.54 ± 1.87 km/h and yaw rate prediction error of 2.62 ± 2.68 deg/s, significantly outperforming other models. More importantly, PINN-GMM-20TRAC incorporates physics-informed neural network constraints to ensure predictions comply with vehicle dynamics principles, maintaining physical reasonableness while achieving high accuracy. The model achieves an optimal balance between computational efficiency and prediction precision, offering better stability and robustness compared to PINN-GMM-5TRAC. This makes it the optimal choice for practical engineering applications that require the integration of accuracy, efficiency, and safety considerations.

Table 1. Model Performance Comparison Analysis

Overall Performance Ranking on Tracking				
Rank	Model	Avg Error (m)	Score	Time (ms)/step
1	NN-ONLY	8.25	0.121	7.45 ± 1.08
2	PINN-20	10.47	0.096	10.78 ± 1.32
3	PINN-5	14.13	0.071	10.75 ± 1.34
4	4DOF	35.20	0.028	3.21 ± 2.03
RMSE Performance (Mean \pm Std)				
Variable	4DOF	PINN-5	PINN-20	NN-ONLY
$X(m)$	37.75 ± 29.47	10.77 ± 25.23	7.13 ± 12.20	9.55 ± 9.22
$Y(m)$	16.40 ± 13.66	13.76 ± 46.34	9.02 ± 6.08	4.88 ± 5.29
$v_x(\text{km}/\text{h})$	42.96 ± 57.71	7.78 ± 13.84	7.85 ± 13.83	5.94 ± 6.30
$v_y(\text{km}/\text{h})$	12.59 ± 8.32	1.36 ± 1.82	1.54 ± 1.87	1.58 ± 1.89
Yaw Rate(deg/s)	28.93 ± 99.78	15.03 ± 100.75	2.62 ± 2.68	6.26 ± 6.35

A typical case for C0 is shown in Figure 9. The 4DOF model in this case exhibits the best accuracy in trajectory tracking, as evidenced by its closest approach to the reference trajectory over the entire path, especially in the challenging curved portion, and achieves the lowest RMSE values on several metrics including position accuracy and yaw rate prediction, since the 4dof model with tires in the linear region theoretically also has excellent performance. However, the PINN-GMM-20 TRAC model has several significant advantages. Most importantly, PINN-based methods provide uncertainty quantification, as shown in (b) uncertainty distribution, where the proba-

bility density distribution is centered around the actual trajectory with a peak value of about 0.0015, providing valuable confidence estimates for safety-critical decisions. From this point of view, PINN also has good performance in this case. It is worth noting that training with 5 tracks outperforms training with 20 tracks, which proves that this case is a special case. Although the PINN model showed slightly higher trajectory bias in the latter half of the path, it maintained competitive performance in longitudinal velocity tracking and demonstrated robust handling of complex vehicle dynamics transitions, especially during the initial high yaw rate phase (approximately 15 degrees/sec).

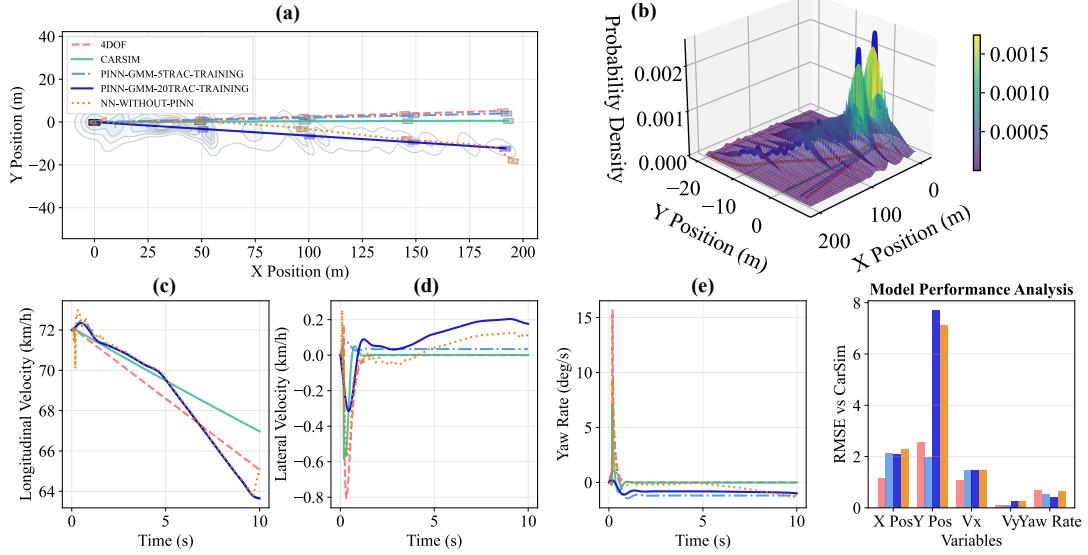


Figure 9. A typical Case for C0

As shown in Figure 10, the main reason for this is that the tire model of the vehicle has not been estimated well. Throughout the sliding process of the vehicle, the state of the tire keeps changing, and the current state of the vehicle affects the state of the tire. Therefore, the PINN method can better capture the changes in the tire, thus obtaining better vehicle state prediction results, while not deviating from the laws of physics to produce unrealistic scenarios. Pure neural network models completely violate the laws of physics, with trajectories exhibiting situations impossible in real-world scenarios, particularly evident in the increase in longitudinal velocity without force input.

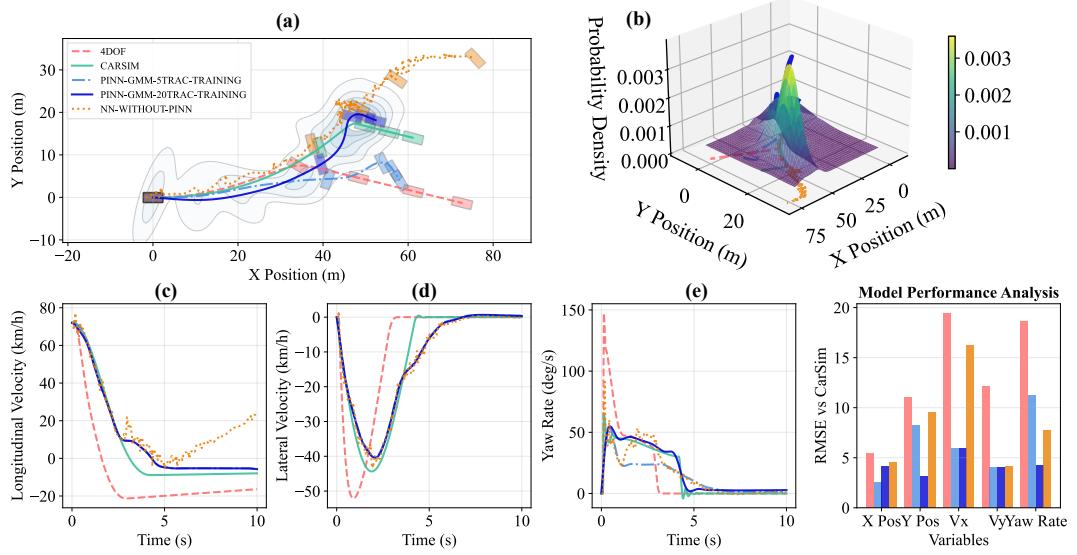


Figure 10. A typical Case for C2

5. Scaled Vehicle Experiments Results and Analysis

5.1. Experiment Setup

To validate the model's performance on real vehicles, a scaled vehicle experiment is conducted using two chassis platforms that accurately represent vehicle dynamic characteristics. The parameters of the Bullet Vehicle and Target Vehicle are detailed in Table 2 and Table 3, respectively. The scale ratio between the experimental vehicles and actual vehicles is approximately 1:4.

Table 2. Bullet Vehicle Parameters

Parameter	Value
External Dimensions	2100mm×1120mm×455mm (L×W×H)
Ground Clearance	120mm
Weight & Payload	Weight: 290KG
Max Acceleration	0.8g (No Load)
Rated Power	12KW
Peak Power	24KW
Max Speed	100 km/h
Steering Type	Front-wheel steering
Drive Mode	Independent four-wheel drive
Communication	CAN Bus

As shown in Figure 11, PIT experiments are conducted to validate the feasibility and accuracy of the proposed modeling method. Both vehicles used in the experiment are equipped with steer-by-wire and drive-by-wire capabilities, and each is integrated with an Industrial Personal Computer (IPC) for executing planning and control algorithms. A GNSS-IMU system and a camera are installed for data acquisition. Communication between the two vehicles is achieved via TCP/IP protocol within a local area network for real-time data transmission.

Table 3. Target Vehicle Parameters

Parameter	Value
External Dimensions	1900mm×900mm×350mm (L×W×H)
Weight & Payload	Weight: 100KG
Max Speed	30 km/h
Steering Type	Front-wheel steering
Drive Mode	rear-wheel drive
Communication	Serial RS232

The bullet vehicle operates an autonomous driving algorithm designed to execute the PIT maneuver. Based on the real-time position of the target vehicle, it performs prediction, planning, and control to achieve the desired PIT state before initiating the PIT maneuver. Control commands are directly sent via the CAN bus to assign torque signals to each of the four wheels.

The entire PIT process can be divided into two phases. The first phase is the chasing process, where the primary control objective of the bullet vehicle is to achieve a parallel body motion and matching velocity with the target vehicle, while maintaining a lateral distance that ensures the required collision angle and speed for PIT execution. The second phase involves performing the PIT maneuver, whose main goal is to induce a significant yaw change in the target vehicle, thereby disabling its ability to continue moving forward. The model proposed in this paper primarily predicts the vehicle's posture changes during post-impact under the assumption that the steering wheel and throttle are released, and that no electronic stability control (ESC) or similar systems are active.

Similar to simulation, first use the 8000 sampling data obtained from simulating a 4DOF model containing force inputs as pre-training data to allow the network to fully learn the physical laws, then use the 100 sampling data of scaled experiment for Fine-tuning the model.

The test results are presented in Table 4. The proposed Physics-Informed Neural Network demonstrates superior trajectory tracking performance, achieving a 63.6% reduction in average trajectory error (0.73 ± 0.58 m vs. 2.01 ± 1.77 m) compared to the conventional 4DOF model, indicating its enhanced capability in capturing complex vehicle dynamics. Notably, the proposed model excels in longitudinal positioning ($R^2 = 0.984$) and velocity prediction ($R^2 = 0.708$), demonstrating successful integration of physical constraints with data-driven learning. However, this improvement in accuracy comes with a modest computational overhead, as the proposed method requires slightly longer inference times (13.76 ms vs. 4.42 ms) than the 4DOF model. The inference time remains stable across various operational conditions, indicating good computational stability.

This trade-off between accuracy and computational cost is reasonable, as more complex models inherently require greater computational resources. The increased inference time of the proposed model is primarily attributed to its neural network architecture, which is predictable and expected. Meanwhile, the model's prediction accuracy can be further improved with additional training data, as validated in previous Carsim simulation experiments. Although the proposed approach maintains high accuracy in short-to-medium term predictions, its physical constraints may become insufficient for extended prediction horizons, particularly when limited data availability hinders neural network convergence.

Therefore, the proposed model is particularly suitable for applications requiring

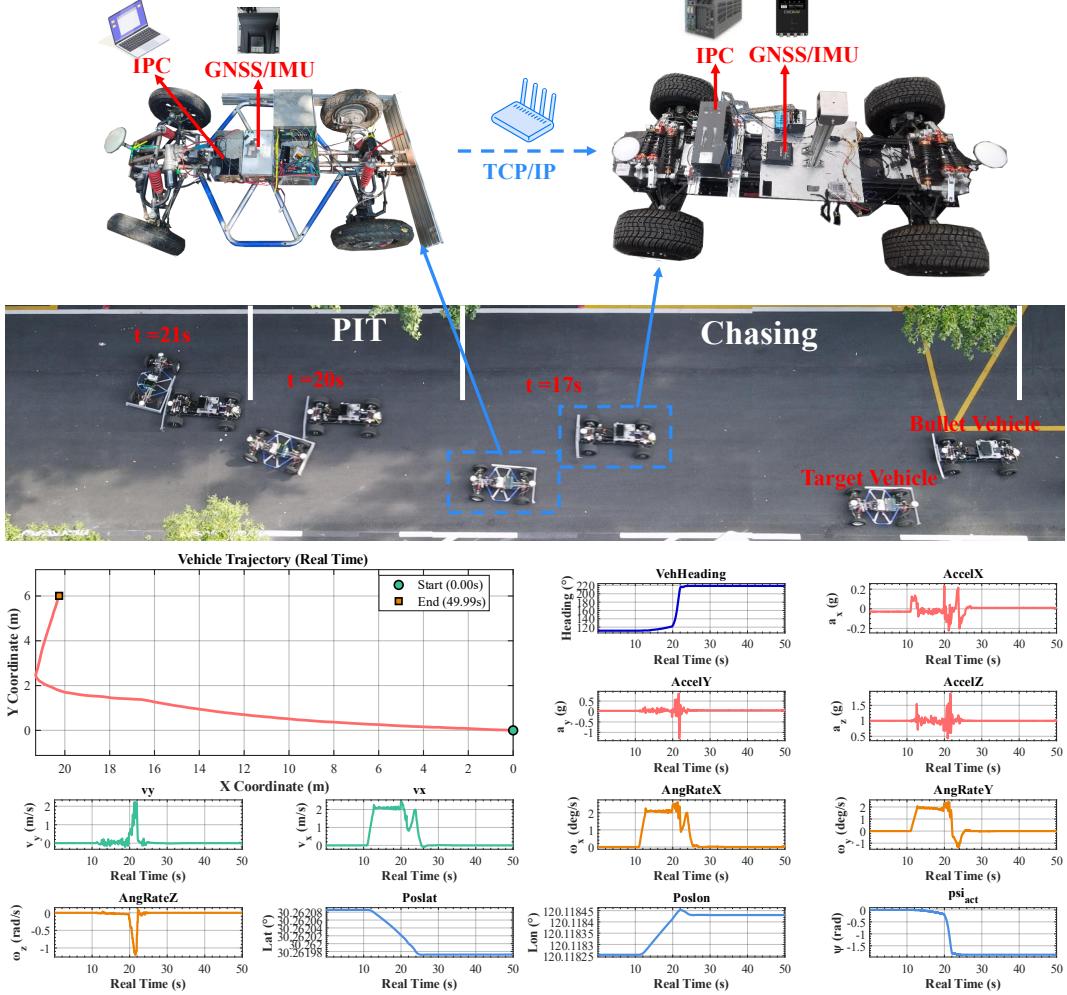


Figure 11. Scaled Vehicles and Experiment Scenario

high-precision short-term predictions (such as the PIT post-collision trajectory optimization discussed in this paper), while the traditional 4DOF model retains advantages in real-time control systems where computational efficiency is critical. However, when control systems require calibration through data-driven approaches, the proposed framework offers a valuable solution for model improvement through continuous data integration.

As shown in Figure 12, case 3 presents even with this earlier collision time, the proposed model maintains excellent trajectory tracking performance prior to collision. The 4DOF model exhibits more severe deviations in this case, particularly showing substantial errors in Y position predictions. The velocity time series observation indicates that the proposed model more accurately captures the vehicle's dynamic characteristics during steering maneuvers, while the 4DOF model demonstrates obvious deficiencies in lateral dynamics modeling.

Table 4. Vehicle Dynamics Model Performance Analysis of Scaled Vehicle Experiments

Overall Performance Ranking on Trajectory Tracking		
Model	Avg Error (m)	Time (ms)/step
PINN Model	0.73 ± 0.58	13.76 ± 1.89
4DOF Model	2.01 ± 1.77	4.42 ± 3.51
RMSE Performance (Mean \pm Std)		
Variable	4DOF Model	PINN Model
X Position (m)	3.55 ± 1.91	0.82 ± 0.51
Y Position (m)	1.87 ± 1.15	1.16 ± 1.28
Longitudinal Velocity (m/s)	0.81 ± 0.34	0.33 ± 0.21
Lateral Velocity (m/s)	0.21 ± 0.08	0.20 ± 0.08
Yaw Rate (rad/s)	0.13 ± 0.05	0.14 ± 0.06

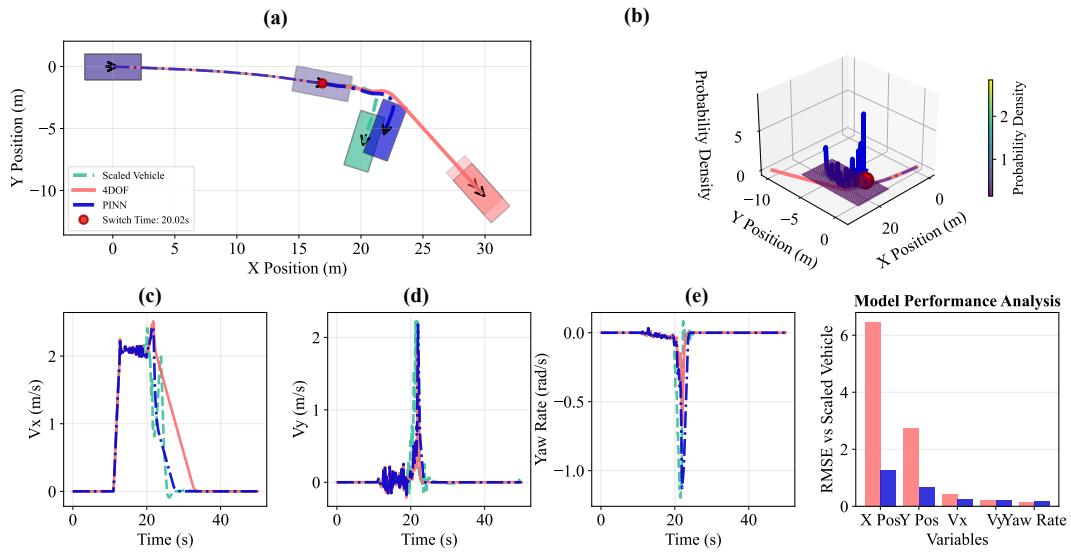


Figure 12. Case 3 Results in Scaled Vehicle Experiments

In case 1, as shown in Figure 13, the proposed model demonstrates exceptional trajectory tracking capability, maintaining near-perfect alignment with the reference trajectory until the collision time point at 35.59 seconds, while the 4DOF model shows significant deviation from the beginning. The time series analysis reveals that the proposed model provides more accurate predictions of both v_x and v_y . The RMSE comparison demonstrates the proposed model's overwhelming advantage in position prediction, with significantly lower errors than the 4DOF model. Probability density plots further confirm that the proposed model's predictions are more concentrated around the actual trajectory.

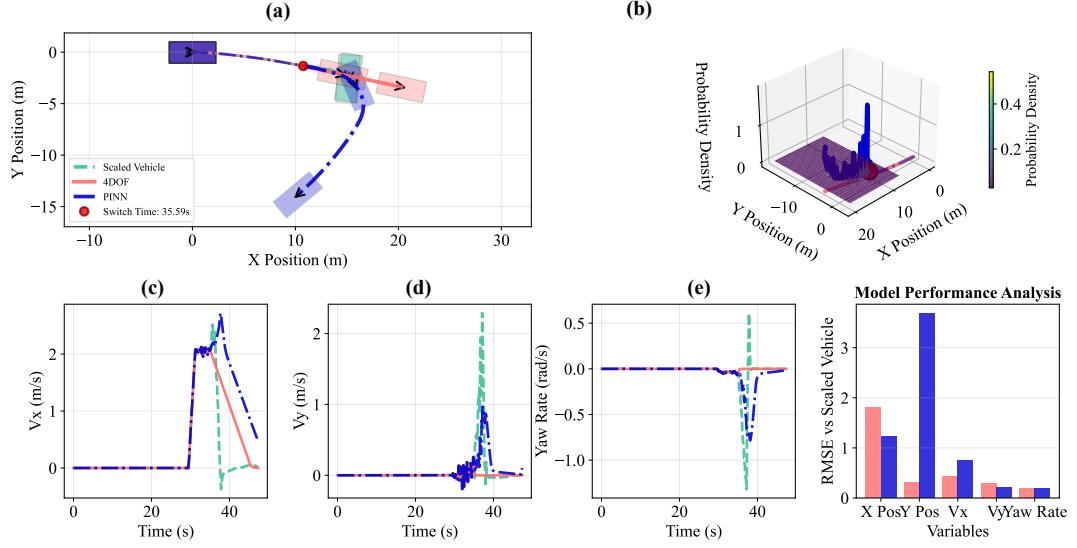


Figure 13. Case 1 Results in Scaled Vehicle Experiments

Both cases confirm the advantages of the proposed model's physical constraint integration, enabling better consistency with actual vehicle behavior in complex maneuvering scenarios. However, the collision phenomenon observed in the proposed model requires attention, potentially related to numerical stability under extreme conditions and the balance of physical constraint weights. Although the 4DOF model is less accurate than the proposed model, its predictive behavior is more stable without significant numerical divergence. This comparison indicates that model selection should be based on specific accuracy requirements and robustness needs in practical applications. The proposed model shows clear advantages for applications requiring high-precision short-term predictions, while the traditional 4DOF model may be more reliable for control systems requiring long-term stable predictions. Particularly in safety-critical applications, numerical stability and the ability to avoid sudden divergence are crucial.

6. Conclusion

This paper proposes two PINN frameworks for vehicle collision dynamics prediction in PIT maneuvers. The first network combines Gaussian Mixture Models with physics constraints to model impact forces, achieving relative errors below 15.0% on FEA datasets. The second Adaptive PINN predicts post-collision vehicle dynamics with dynamic constraint weighting, reducing average trajectory prediction error by 63.6% compared to traditional 4DOF model while maintaining millisecond-level computational efficiency.

The framework is validated through computer simulations and scaled vehicle experiments, demonstrating improved accuracy across different test conditions. The method provides uncertainty estimates that are valuable for safety applications, and successfully combines physics knowledge with data-driven learning to work effectively even with limited training data. Future work should extend the framework to broader collision scenarios, incorporate advanced tire models, and develop automated hyperparameter tuning. This approach bridges theoretical vehicle dynamics with practical collision scenarios, positioning it as a valuable tool for next-generation active safety

and post-impact control systems.

Future work should expand the framework to broader collision scenarios and develop automatic parameter tuning methods, while also applying the corresponding models to design autonomous driving algorithms.

Funding

This work was supported by National Natural Science Foundation of China (Grant No. 52372421).

Disclosure statement

The authors report there are no competing interests to declare.

7. References

References

- [1] Masoumeh Parseh, Mikael Nybacka, and Fredrik Asplund. Motion planning for autonomous vehicles with the inclusion of post-impact motions for minimising collision risk. *Vehicle System Dynamics*, 61(6):1707–1733, June 2023.
- [2] Mustafa Elkady, Ahmed Elmarakbi, John MacIntyre, and Mohammed Alhariri. Collision mitigation and vehicle transportation safety using integrated vehicle dynamics control systems. *Journal of Traffic and Transportation Engineering (English Edition)*, 4(1):41–60, February 2017.
- [3] Derong Yang, Tim Gordon, and Ulrich Sander. Post-impact motion control of road vehicles. *Vehicle System Dynamics*, 63(7):1300–1357, July 2025.
- [4] Derong Yang. *Vehicle Dynamics Control after Impacts in Multiple-Event Accidents*. Chalmers University of Technology, Göteborg, 2013.
- [5] Jing Zhou, Jianbo Lu, and Huei Peng. Vehicle dynamics in response to the maneuver of precision immobilization technique. In *Dynamic Systems and Control Conference*, volume 43352, pages 577–584, 2008.
- [6] Jing Zhou, Huei Peng, and Jianbo Lu. Collision model for vehicle motion prediction after light impacts. *Vehicle System Dynamics*, 46(sup1):3–15, September 2008.
- [7] David D.L. Mascareñas, Christopher J. Stull, and Charles R. Farrar. Autonomous execution of the Precision Immobilization Technique. *Mechanical Systems and Signal Processing*, 87:153–168, March 2017.
- [8] Jos Huibers and Eric de Beer. Current front stiffness of european vehicles with regard to compatibility. Technical report, SAE Technical Paper, 2001.
- [9] Par Jonsén, Erik Isaksson, Karl Gustaf Sundin, and Mats Oldenburg. Identification of lumped parameter automotive crash models for bumper system development. *International Journal of Crashworthiness*, 14(6):533–541, November 2009.
- [10] Sunday M Ofochebe, Chigbogu G Ozoegwu, and Samuel O Enibe. Performance evaluation of vehicle front structure in crash energy management using lumped mass spring system. *Advanced Modeling and Simulation in Engineering Sciences*, 2(1):2, December 2015.
- [11] Johan Iraeus and Mats Lindquist. Pulse shape analysis and data reduction of real-life frontal crashes with modern passenger cars. *International Journal of Crashworthiness*, 20(6):535–546, November 2015.
- [12] Byung-joo Kim and Huei Peng. Optimal Vehicle Motion Control to Mitigate Secondary

- Crashes After an Initial Impact. page V001T10A002, San Antonio, Texas, USA, October 2014. American Society of Mechanical Engineers.
- [13] Matthew Huang. *Vehicle Crash Mechanics*. CRC Press, 1st edition, 2002.
 - [14] Zuolong Wei, Kjell G. Robbersmyr, and Hamid R. Karimi. Data-based modeling and estimation of vehicle crash processes in frontal fixed-barrier crashes. *Journal of the Franklin Institute*, 354(12):4896–4912, August 2017.
 - [15] Luís Sousa, Paulo Veríssimo, and Jorge Ambrósio. Development of generic multibody road vehicle models for crashworthiness. *Multibody System Dynamics*, 19(1-2):133–158, February 2008.
 - [16] Qijun Chen, Yuxi Xie, Yu Ao, Tiange Li, Guorong Chen, Shaofei Ren, Chao Wang, and Shaofan Li. A deep neural network inverse solution to recover pre-crash impact data of car collisions. *Transportation Research Part C: Emerging Technologies*, 126:103009, May 2021.
 - [17] Ali R. Yildiz and Kiran N. Solanki. Multi-objective optimization of vehicle crashworthiness using a new particle swarm based approach. *The International Journal of Advanced Manufacturing Technology*, 59(1-4):367–376, March 2012.
 - [18] Dario Vangi. *Vehicle Collision Dynamics: Analysis and Reconstruction*. Butterworth-Heinemann, Oxford [England] ; Cambridge, MA, 2020.
 - [19] Cong Wang, Zhenpo Wang, Lei Zhang, Hui long Yu, and Dongpu Cao. Post-Impact Motion Planning and Tracking Control for Autonomous Vehicles. *Chinese Journal of Mechanical Engineering*, 35(1):54, December 2022.
 - [20] Office of Justice Programs. Studying how to maneuver suspects to a stop, 2024.
 - [21] Raymond M Brach. An impact moment coefficient for vehicle collision analysis. *SAE Transactions*, pages 30–37, 1977.
 - [22] Hirotoshi Ishikawa. Impact model for accident reconstruction-normal and tangential restitution coefficients. Technical report, SAE Technical Paper, 1993.
 - [23] Harald Kolk, Ernst Tomasch, Jörg Bakker, Jan Dobberstein, and Wolfgang Sinz. Evaluation of a momentum based impact model and application in an effectivity study considering junction accidents. In *International Conference "ESAR-Expert Symposium on Accident Research"*, 2016.
 - [24] R. Emori. Vehicle mechanics of intersection collision impact. SAE Technical Paper 700177, SAE International, 1970.
 - [25] Raymond M Brach. Analysis of planar vehicle collisions using equations of impulse and momentum. *Accident Analysis & Prevention*, 15(2):105–120, 1983.
 - [26] Ahmed Elmarakbi, Mustafa Elkady, and John MacIntyre. Numerical analysis of vehicle-to-vehicle impact using vehicle dynamics control systems for collision mitigation. *International Journal of Dynamics and Control*, 1(2):172–191, June 2013.
 - [27] Mirosław Gidlewski, Leon Prochowski, Leszek Jemioł, and Dariusz Żardecki. The process of front-to-side collision of motor vehicles in terms of energy balance. *Nonlinear Dynamics*, 97(3):1877–1893, August 2019.
 - [28] Mauro Da Lio, Daniele Bortoluzzi, and Gastone Pietro Rosati Papini. Modelling longitudinal vehicle dynamics with neural networks. *Vehicle System Dynamics*, 58(11):1675–1693, November 2020.
 - [29] John Chrosniak, Jingyun Ning, and Madhur Behl. Deep dynamics: Vehicle dynamics modeling with a physics-informed neural network for autonomous racing. *arXiv preprint arXiv:2312.04374*, 2023.
 - [30] Nathan A. Spielberg, Matthew Brown, Nitin R. Kapania, John C. Kegelman, and J. Christian Gerdes. Neural network vehicle models for high-performance automated driving. *Science Robotics*, 4(28):eaaw1975, March 2019.
 - [31] Grady Williams, Nolan Wagener, Brian Goldfain, Paul Drews, James M Rehg, Byron Boots, and Evangelos A Theodorou. Information Theoretic MPC Using Neural Network Dynamics.
 - [32] Yongjun Pan, Xiaobo Nie, Zhixiong Li, and Shuitao Gu. Data-driven vehicle modeling of longitudinal dynamics based on a multibody model and deep neural networks. *Measure-*

ment, 180:109541, August 2021.

- [33] Taekyung Kim, Hojin Lee, and Wonsuk Lee. Physics Embedded Neural Network Vehicle Model and Applications in Risk-Aware Autonomous Driving Using Latent Features. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4182–4189, Kyoto, Japan, October 2022. IEEE.
- [34] George Em Karniadakis, Ioannis G. Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, May 2021.
- [35] Maziar Raissi, Paris G. Perdikaris, and George Em Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, February 2019.
- [36] Zhaobin Mo, Rongye Shi, and Xuan Di. A physics-informed deep learning paradigm for car-following models. *Transportation Research Part C: Emerging Technologies*, 130:103240, September 2021.
- [37] Haotian Shi, Yang Zhou, Keshu Wu, Sikai Chen, Bin Ran, and Qinghui Nie. Physics-informed deep reinforcement learning-based integrated two-dimensional car-following control strategy for connected automated vehicles. *Knowledge-Based Systems*, 269:110485, June 2023.
- [38] Qixiu Cheng, Yang Zhang, Ziyuan Gu, Zelin Wang, Hongzhe Liu, Yuqian Lin, and Zhiyuan Liu. A hybrid physics-based and data-driven approach for car-following behavior modeling and analysis. *Transportation Research Part C: Emerging Technologies*, 177:105207, August 2025.
- [39] Maosi Geng, Junyi Li, Yingji Xia, and Xiqun (Michael) Chen. A physics-informed transformer model for vehicle trajectory prediction on highways. *Transportation Research Part C: Emerging Technologies*, 154:104272, 2023.
- [40] Chenkai Tan, Yingfeng Cai, Hai Wang, Long Chen, and Yubo Lian. Modeling vehicle dynamics with physics-informed deep operator network. *Vehicle System Dynamics*, pages 1–29, July 2025.
- [41] Shiming Fang and Kaiyan Yu. Fine-tuning hybrid physics-informed neural networks for vehicle dynamics model estimation. *IFAC-PapersOnLine*, 58(28):810–815, 2024.
- [42] Keke Long, Xiaowei Shi, and Xiaopeng Li. Physics-informed neural network for cross-dynamics vehicle trajectory stitching. *Transportation Research Part E: Logistics and Transportation Review*, 192:103799, December 2024.