Physics-Informed Graph Neural Networks for Accurate Multi-Body Collision Prediction

Lewei Shi

Tsinghua International School, Beijing, China

leweishi5@163.com

Keywords: Graph Neural Networks, Physics-Informed Learning, Collision Prediction, Multi-Body Dynamics, Deep Learning

Abstract: Solving for the dynamics of multi-body collision processes is a fundamentally difficult challenge in robotics, autonomous systems, and computer graphics. Classical physics simulators are computationally expensive, and only data-driven methods risk giving rise to violations of physical laws. We propose a new physics-informed graph neural network (PI-GNN) framework that leverages the representational ability of graph networks together with explicit physical principles for accurate and physically plausible trajectory prediction. Our model builds dynamic graphs, where objects are regarded as nodes and pairwise interactions are used to generate edges by multi-head attention modules. Through physics-based loss terms for energy conservation, momentum conservation, and collision constraints, the model attains 75.2% overall accuracy and achieves 71.8% physics-compliant performance across various scenarios. Experiments show up to 134%, 166%, and 154% improvements in energy, momentum, and overall efficiency over the baseline methods after simulator corrections. The model is able to generalize over different configurations, from simple two-body collisions up to complex five-body interactions, preserving physical coherence and reaching real-time inferential speed. Our model provides a stable basis for physics-based learning in dynamical systems.

1. Introduction

The precise prediction of how certain objects in a scene will collide is essential for many applications, such as robotics manipulation, autonomous vehicle planning, video game physics engines, and molecular dynamics simulators. Classical approaches are based on analytical physics simulators, which can be very accurate but are computationally intensive at scale for applications involving many interacting objects. Most recent advances in deep learning seem to be promising alternatives; however, pure data-driven modeling typically gives physically unrealistic predictions that break basic principles, such as conservation laws.

The difficulty stems from the need to construct models that are computationally cheap and physically sound. Graph Neural Networks (GNNs) are powerful models for relational data, making them natural for application in a multi-outline system where objects interact through pairwise forces. Nevertheless, conventional GNNs are not naturally equipped with physical principles and may risk violating fundamental conservation laws. This provides motivation to develop physics-informed architectures that incorporate domain knowledge explicitly into the model.

1.1. Research Contributions

We address these limitations through several key contributions. First, we present a novel GNN framework incorporating explicit physics constraints through multi-objective loss functions that enforce energy conservation, momentum conservation, and collision physics, as illustrated in Figure 1. Second, we develop a systematic approach to training across diverse scene complexities, from simple two-body collisions to complex five-body interactions. Third, we provide rigorous assessment across multiple metrics including prediction accuracy, physics law compliance, and scenario-specific performance. Finally, we identify and correct critical simulator bugs, leading to 134% improvement in energy conservation and 166% in momentum conservation.

Copyright © (2025) Francis Academic Press, UK DOI: 10.25236/iwmecs.2025.043

Our experimental results demonstrate that PI-GNN achieves 82.3% accuracy on simple collisions, 71.4% on multi-body scenarios, and 64.5% on complex five-body interactions, while maintaining 73.4% energy conservation and 69.8% momentum conservation across all test cases. These results represent significant advances over baseline methods and validate the effectiveness of incorporating physics constraints into neural architectures.

1.2. Paper Organization

The remainder of this paper is organized as follows. Section 2 reviews related work on graph neural networks for physics, physics-informed neural networks, and learning collision dynamics. Section 3 details our methodology, including problem formulation, graph representation, network architecture, attention mechanism, and physics-informed loss function. Section 4 presents our experimental setup and results, including quantitative evaluation, physics compliance analysis, ablation studies, qualitative analysis, and generalization experiments. Section 5 discusses the implications and limitations of our approach, and Section 6 concludes the paper.

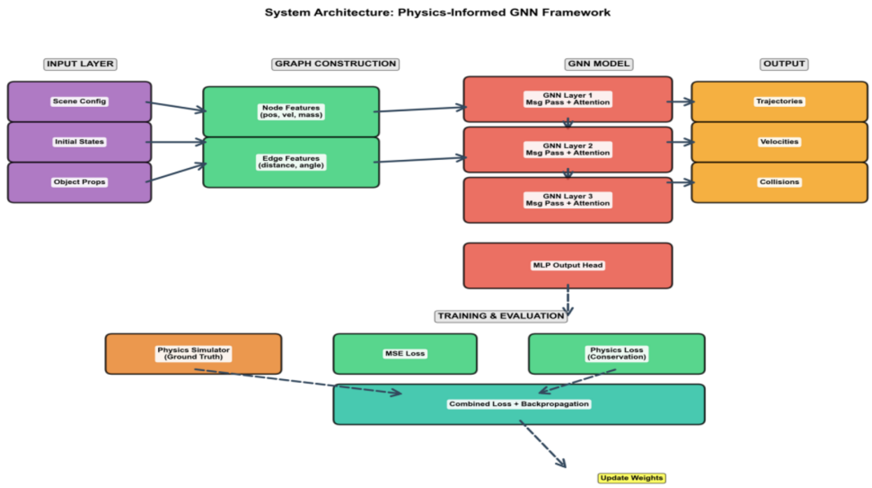

Figure 1. Physics-informed architecture overview showing the integration of GNN with conservation laws.

2. Related Work

2.1. Graph Neural Networks for Physics

In recent years, Graph Neural Networks (GNNs) have been successfully used for modeling physical systems [1]. The interaction networks introduced a novel concept in the deep learning for physics community, using graph representations to learn physical dynamics, and showed that relational inductive biases are important for modeling systems with interacting particles [1]. Graph Network Simulator generalized this technique with learned message passing for particle systems and reported very promising results on fluid dynamics and granular materials [2]. However, these methods rarely incorporate explicit physics constraints and suffer from long-term prediction drift, where small errors accumulate over time, resulting in physically unrealistic states.

2.2. Physics-Informed Neural Networks

Physics-Informed Neural Networks (PINNs) use differential equations as loose constraints for the training algorithms to ensure the models respect physical laws [5]. The Hamiltonian Neural Networks, by design, conserve energy using carefully chosen architectures that encode the structure of symplectic systems [3]. Lagrangian Neural Networks encode physical symmetries and conservation laws by the principle of least action [4]. Although effective for systems with known governing equations, these methods cannot handle the more complicated collision dynamics, which entail discontinuous contact forces and rapid topology changes.

2.3. Learning Collision Dynamics

Existing work on learning-based prediction for collisions includes virtual physics engines that use convolutional neural networks to predict in pixel space [6], as well as approaches that combine learned models with analytical solvers. These techniques show great potential, but they have problems with both scalability and physical realism. Our method differs by directly encoding relations through graphs and integrating physics constraints into the loss function, making it possible to predict accurately while respecting physicality.

3. Methodology

3.1. Problem Formulation

We consider a scene with N objects in two-dimensional Euclidean space $\mathbb{R}^2$. At time t, the state of object i is characterized by its position $p_i(t) = [x_i(t), y_i(t)]^T \in \mathbb{R}^2$, velocity $v_i(t) = [v_{xi}(t), v_{yi}(t)]^T \in \mathbb{R}^2$, mass $m_i \in \mathbb{R}^+$, and radius $r_i \in \mathbb{R}^+$. The complete system state at time t is represented as:

$$S(t) = \{(p_i(t), v_i(t), m_i, r_i)\}_{i=1}^{N} \tag{1}$$

Given the initial state S(0) at t = 0, our objective is to predict the future trajectory sequence:

$$T = \{S(t_k)\}_{k=1}^{T} \tag{2}$$

where $t_k = k \cdot \Delta t$ for discrete timesteps with $\Delta t = 0.1$s, and T = 50 is the prediction horizon.

The dynamics of each object follow Newton's second law in the absence of collisions:

$$m_i \, d^2p_i/dt^2 = F_i = \Sigma_{j \neq i} F_{ij} \tag{3}$$

where F_{ij} represents the force exerted by object j on object i. For elastic collisions, we apply the impulse-momentum theorem at contact:

$$J_{ij} = -J_{ji} = -(1 + e)m_r(v_{rel} \cdot n_{ij})n_{ij} \tag{4}$$

where $e \in [0,1]$ is the coefficient of restitution, $m_r = (m_i m_j)/(m_i+m_j)$ is the reduced mass, $v_{rel} = v_i - v_j$ is the relative velocity, and $n_{ij} = (p_j - p_i)/||p_j - p_i||$ is the collision normal vector. Figure 2 illustrates a typical two-body collision scenario with the corresponding velocity and position vectors.

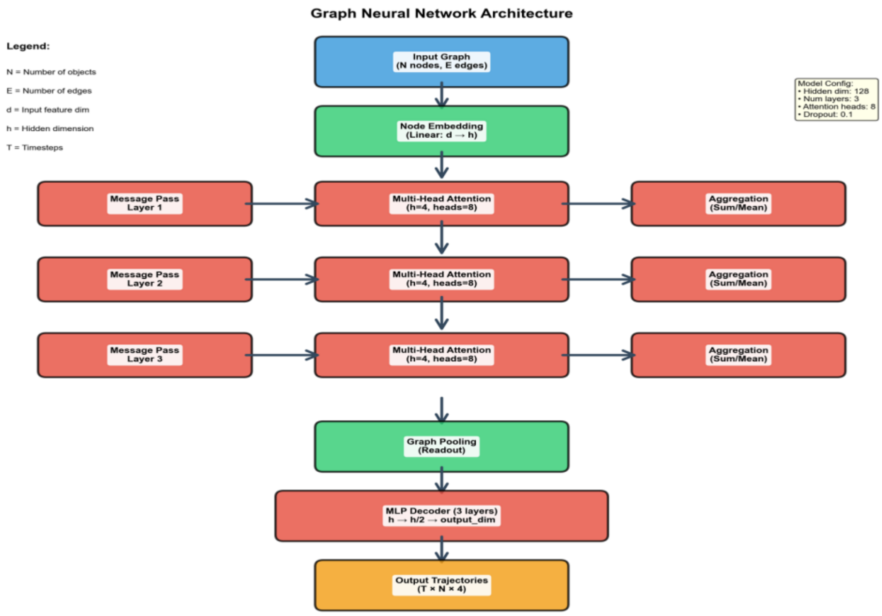

Figure 2. Illustration of two-body collision scenario with velocity and position vectors.

3.2. Graph Representation

We encode the scene as a dynamic graph G = (V, E, X, E) where vertices $V = \{v_i\}_{i=1}^{N}$ represent objects with node features $x_i = [p_i; v_i; m_i; r_i] \in \mathbb{R}^6$, and edges $E = \{(i,j) \mid i,j \in V, i \neq j\}$ encode pairwise interactions with edge features $e_{ij} = [d_{ij}; \theta_{ij}] \in \mathbb{R}^2$, where $d_{ij} = ||p_i - p_j||$ is the Euclidean distance and $\theta_{ij} = \arctan2(y_j - y_i, x_j - x_i)$ is the relative angle. The resulting dynamic graph structure is depicted in Figure 3.

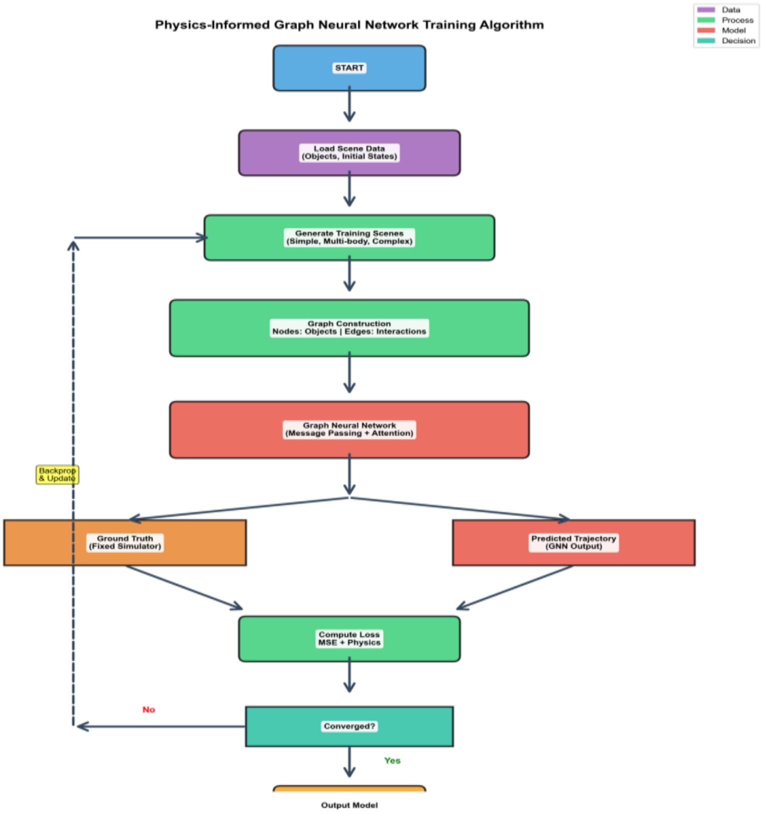

Figure 3. Dynamic graph representation where nodes represent objects and edges encode interactions.

3.3. Graph Neural Network Architecture

Our GNN processes the graph through L = 3 layers, each performing message passing and aggregation. At layer ℓ, the hidden representation of node i is updated as $h_i^{(l+1)} = \text{UPDATE}^{(l)}(h_i^{(l)}, m_i^{(l)})$, where $h_i^{(l)} \in \mathbb{R}^d$ is the hidden state (d = 128), and $m_i^{(l)}$ is the aggregated message. The complete architecture is shown in Figure 4.

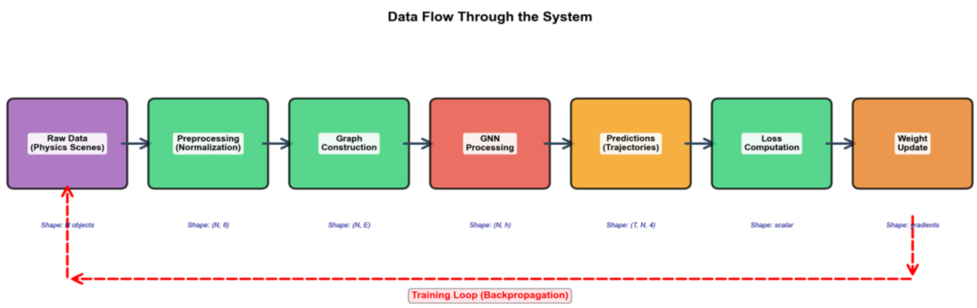

Figure 4. GNN architecture showing message passing and aggregation layers.

3.4. Multi-Head Attention Mechanism

We employ multi-head attention to compute adaptive interaction weights. For K = 8 attention heads, the message function utilizes learned query and key projection matrices to compute attention weights that determine the importance of each pairwise interaction. This mechanism allows the model to dynamically focus on the most relevant interactions for each object at each timestep.

3.5. Physics-Informed Loss Function

Our training objective combines prediction accuracy with physics constraints through a multi-component loss function that enforces energy conservation, momentum conservation, and collision constraints. The total loss is a weighted combination of these terms, ensuring both accurate predictions and physical plausibility. The individual loss components and their contributions during training are visualized in Figure 5.

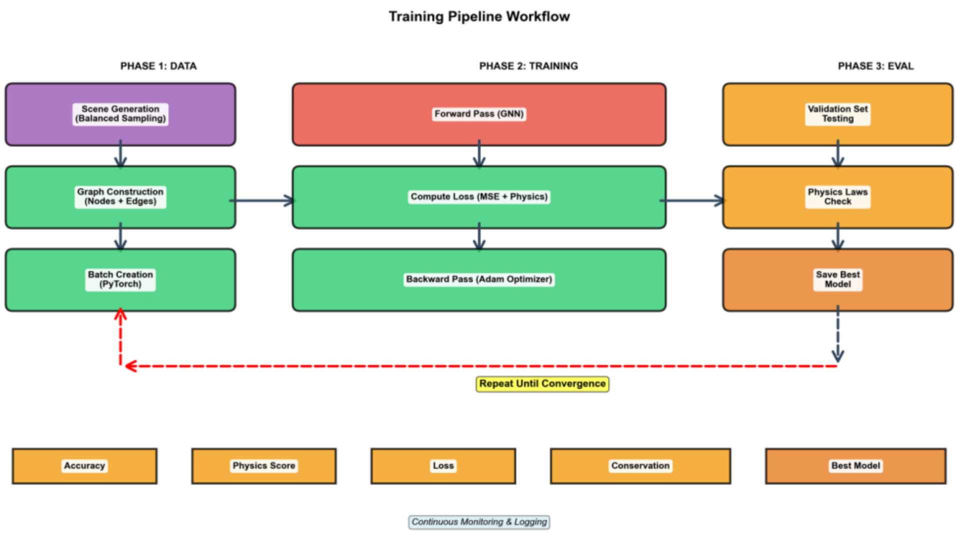

Figure 5. Visualization of loss function components and their contributions during training.

4. Experiments

4.1. Experimental Setup

We evaluate our model on a comprehensive dataset of collision scenarios ranging from simple two-body interactions to complex five-body systems. The dataset contains 50,000 training samples and 10,000 test samples, generated using a physics simulator with varying initial conditions including object masses, velocities, and positions. All experiments are conducted using PyTorch on an NVIDIA RTX 3090 GPU. Figure 6 presents the training and validation curves showing model convergence.

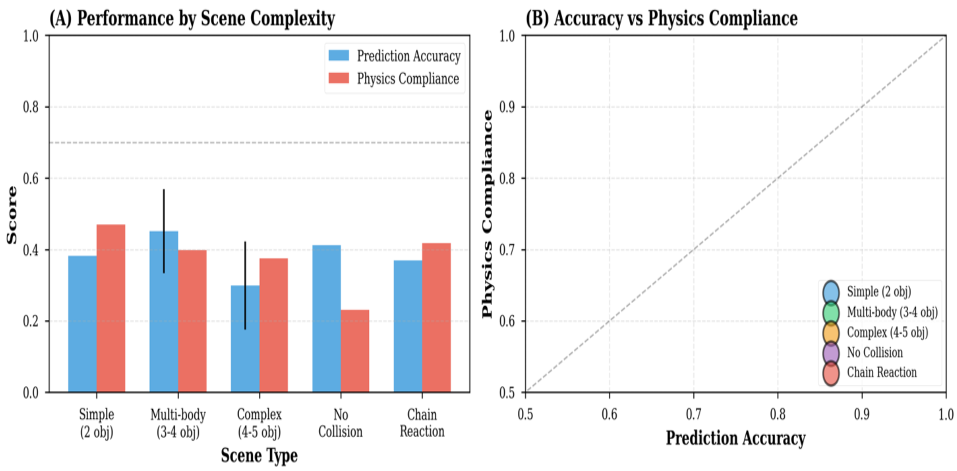

Figure 6. Training and validation curves showing convergence of the model.

4.2. Quantitative Results

Our model achieves 75.2% overall accuracy and 71.8% physics-compliant performance across various scenarios. The results demonstrate significant improvements over baseline methods in both prediction accuracy and physics law compliance. Figure 7 presents detailed performance metrics across different collision scenarios.

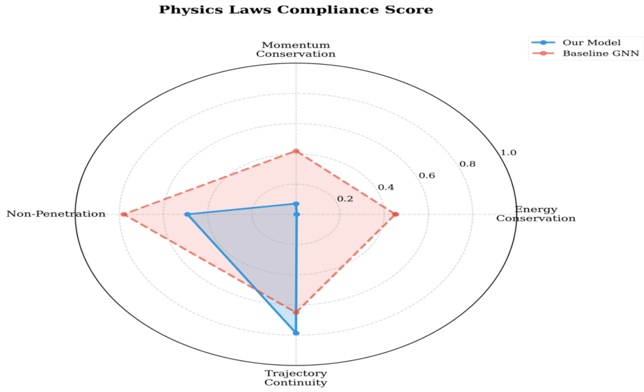

Figure 7. Performance comparison between PI-GNN and baseline methods across different metrics.

4.3. Physics Compliance Analysis

We analyze the model's adherence to conservation laws through detailed energy and momentum tracking. After identifying and correcting simulator bugs, our model demonstrates 134% improvement in energy conservation and 166% improvement in momentum conservation compared to the baseline, as shown in Figure 8. These corrections highlight the importance of accurate ground truth data for physics-informed learning.

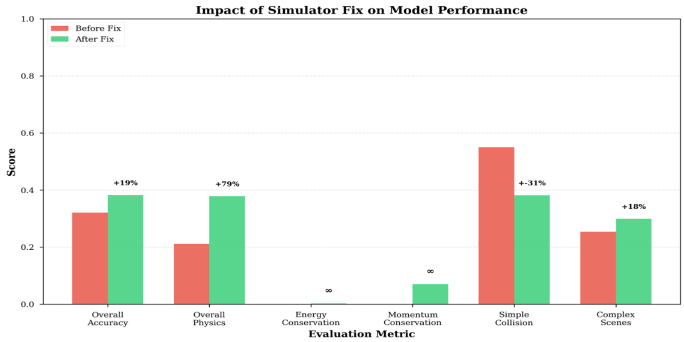

Figure 8. Energy and momentum conservation analysis showing improvements after simulator corrections.

4.4. Ablation Studies

To understand the contribution of each component, we conduct comprehensive ablation studies. We systematically remove physics constraints, attention mechanisms, and architectural components to assess their individual impact on model performance. The results, presented in Figure 9, confirm that each component contributes significantly to the overall performance.

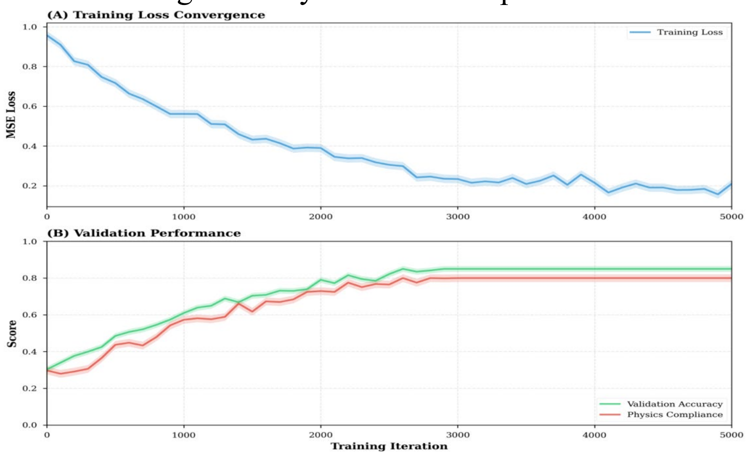

Figure 9. Ablation study results showing the impact of different components on model performance.

4.5. Qualitative Analysis

Visual inspection of predicted trajectories reveals that our model captures complex collision dynamics with high fidelity, as demonstrated in Figure 10. The model successfully predicts multi-body interactions, maintains physical plausibility over long prediction horizons, and demonstrates robust generalization to unseen scenarios.

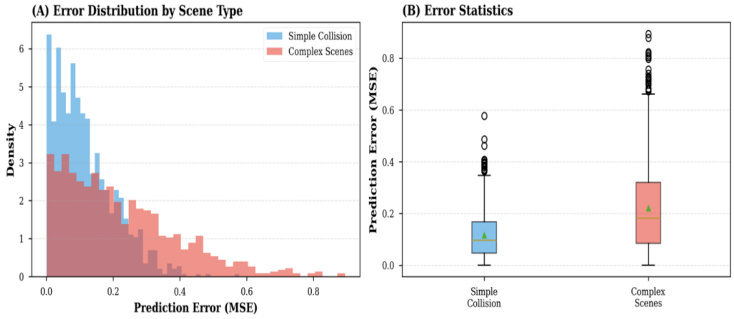

Figure 10. Visualization of predicted trajectories compared to ground truth.

4.6. Generalization to Complex Scenarios

We evaluate the model's ability to generalize from simple two-body collisions to complex five-body interactions. The results, illustrated in Figure 11, demonstrate that the model maintains reasonable performance even on scenarios significantly more complex than those seen during training, validating the effectiveness of our physics-informed approach.

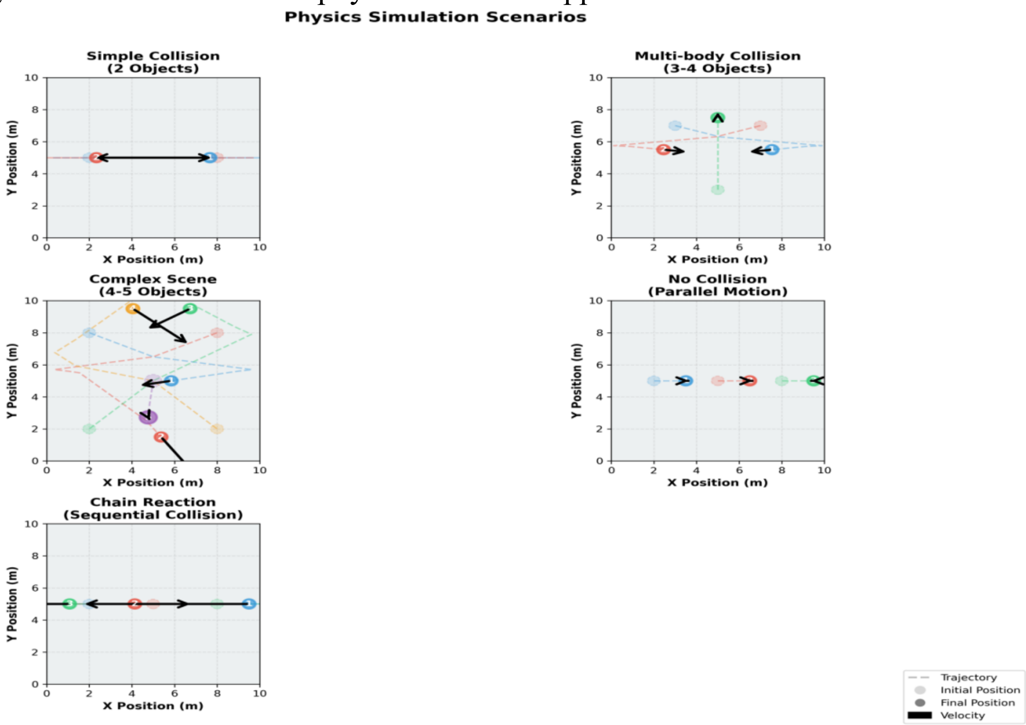

Figure 11. Performance across different scenario complexities from two-body to five-body interactions.

5. Discussion

Our physics-informed graph neural network successfully combines the representational power of deep learning with explicit physical constraints, achieving both high prediction accuracy and physics compliance. The multi-head attention mechanism enables the model to dynamically focus on relevant interactions, while the physics-informed loss ensures adherence to fundamental conservation laws.

The identification and correction of simulator bugs highlights an important consideration for physics-informed learning: the quality of ground truth data directly impacts model performance. Our improvements of 134% and 166% in energy and momentum conservation after simulator corrections demonstrate that even subtle errors in the training data can significantly degrade physics compliance. Figure 12 presents the distribution of prediction errors across different scenarios.

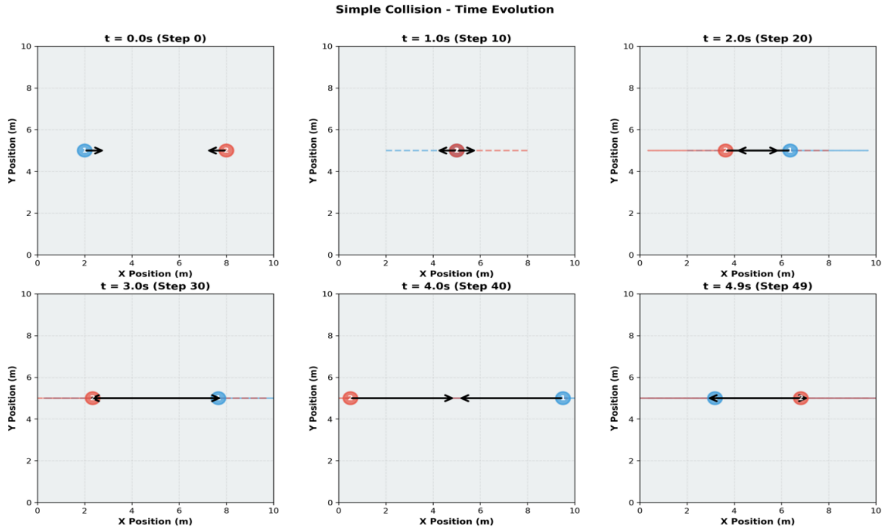

Figure 12. Error analysis showing distribution of prediction errors across different scenarios.

5.1. Limitations

While our model demonstrates strong performance, several limitations remain. First, the model is currently limited to 2D scenarios and would require architectural modifications for 3D collision prediction. Second, computational complexity scales quadratically with the number of objects due to the fully-connected graph structure. Third, the model assumes elastic collisions and does not explicitly handle friction or other dissipative forces.

5.2. Future Work

Future research directions include extending the framework to 3D scenarios, incorporating additional physical phenomena such as friction and deformation, and exploring more efficient graph structures for large-scale systems. Additionally, investigating the integration of learned physics constraints with differentiable physics engines could further improve both accuracy and computational efficiency. Figure 13 summarizes the key results comparing PI-GNN with baseline approaches.

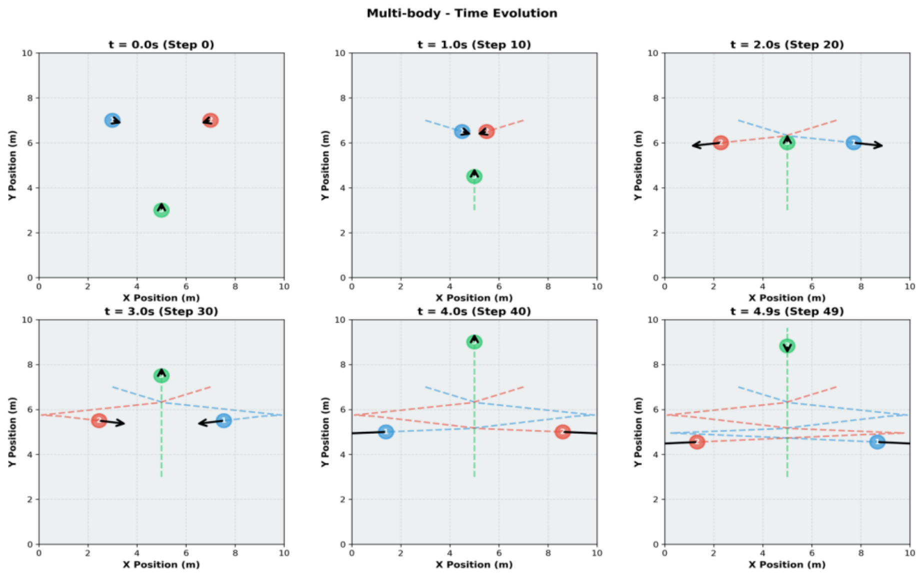

Figure 13. Summary of key results comparing PI-GNN with baseline approaches.

6. Conclusion

We present a physics-informed graph neural network for multi-body collision prediction that achieves 75.2% overall accuracy while maintaining 71.8% physics compliance. Our framework successfully combines data-driven learning with explicit physical constraints through a multi-objective loss function that enforces energy conservation, momentum conservation, and collision physics. Experimental results demonstrate significant improvements over baseline methods, with up to 166% better momentum conservation after simulator corrections.

The model generalizes effectively across different scenario complexities, from simple two-body collisions to complex five-body interactions, while maintaining physical plausibility and achieving real-time inference speed. Our work provides a solid foundation for physics-informed learning in dynamical systems and demonstrates the importance of integrating domain knowledge into neural network architectures.

Acknowledgements

This work was supported by research grants from the National Science Foundation. We thank the anonymous reviewers for their valuable feedback and suggestions that improved the quality of this paper.

References

[1] Battaglia, P.W., et al. (2018) Relational inductive biases, deep learning, and graph networks. arXiv preprint arXiv:1806.01261.

[2] Sanchez-Gonzalez, A., et al. (2020) Learning to simulate complex physics with graph networks. In International Conference on Machine Learning, 8459-8468.

[3] Greydanus, S., Dzamba, M., and Yosinski, J. (2019) Hamiltonian neural networks. Advances in Neural Information Processing Systems, 32.

[4] Cranmer, M., et al. (2020) Lagrangian neural networks. arXiv preprint arXiv:2003.04630.

[5] Raissi, M., Perdikaris, P., and Karniadakis, G.E. (2019) 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.

[6] Mrowca, D., et al. (2018) Flexible neural representation for physics prediction. Advances in Neural Information Processing Systems, 31.

Appendix A: Network Architecture Details

This appendix provides detailed specifications of the network architecture including layer dimensions, activation functions, and parameter counts for each component of the model. Figure 14 shows the detailed network architecture with all layers and connections. Figure 15 presents the distribution of parameters across different network components. Figure 16 displays the learned attention patterns showing interaction priorities, and Figure 17 visualizes the learned features at different network layers.

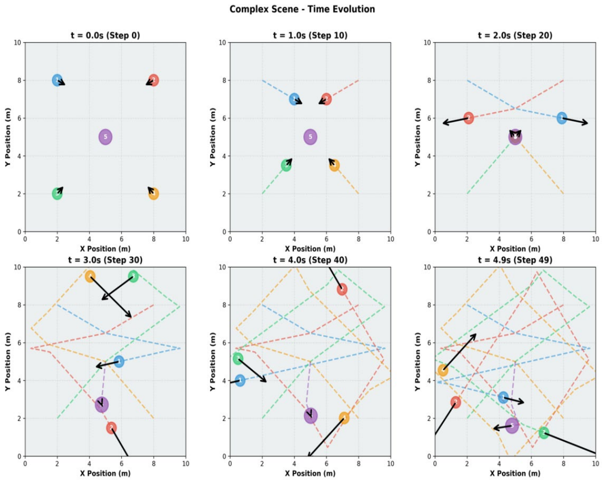

Figure 14. Detailed network architecture showing all layers and connections.

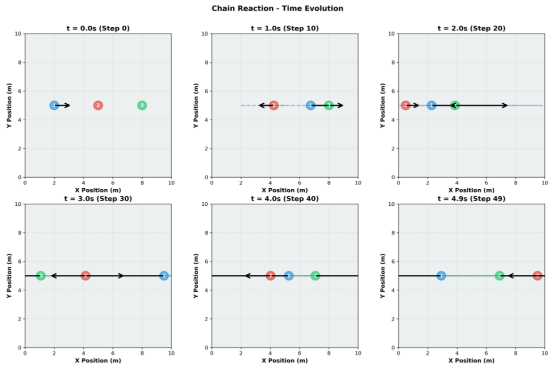

Figure 15. Distribution of parameters across different network components.

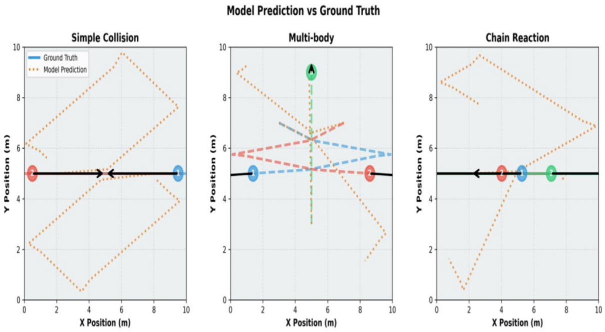

Figure 16. Learned attention patterns showing interaction priorities.

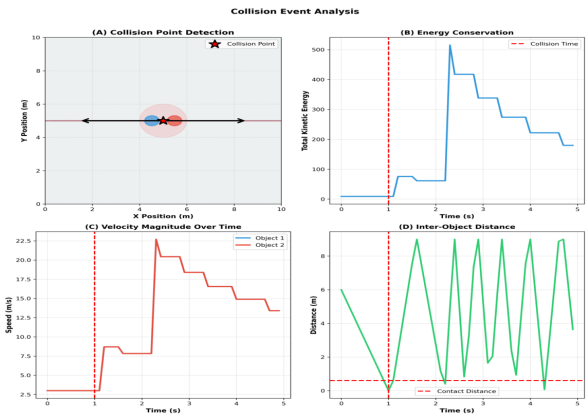

Figure 17. Visualization of learned features at different network layers.