

# Rigid Body Dynamics Simulation Based on Graph Neural Networks with Constraints

Anonymous Authors<sup>1</sup>

## Abstract

In recent years, the utilization of Graph Neural Network (GNN)-based methods for simulating complex physical systems has significantly advanced the field of numerical science and engineering. Despite their success, current GNN-based methods for rigid body dynamic simulation are constrained to relatively simple scenarios, hindering their practical use in industrial settings where complex mechanical structures and interconnected components prevail. These methods face challenges in handling intricate force relationships within rigid bodies, primarily due to the difficulty in obtaining force-related data for objects in industrial environments. To address this, we propose a novel constraint-guided method that incorporates force computations into GNN-based simulations. The model incorporates computations related to both contact and non-contact forces into the prediction process. Additionally, it imposes physical constraints on the prediction process based on Kane's equations. We have rigorously demonstrated the model's rationality and effectiveness with thorough theoretical demonstration and empirical analysis.

## 1. Introduction

Simulation of rigid body dynamics plays a pivotal role in numerous domains within numerical science and engineering (Landau et al., 2008; Thijssen, 2007; Szabó & Babuška, 2021), including mechanical engineering (Brach, 2007), automotive design (Tong, 2000), biomechanics (Silva et al., 1997), and virtual reality (Sauer & Schömer, 1998), enabling the accurate modeling and analysis of object motion, stress, and interactions for optimization and performance enhancement. Traditional methods for simulating rigid body

systems often rely on intricate physical equations tailored to specific domains (Huston & Passerello, 1979; Mirtich, 1996; Featherstone, 2014). This necessitates a precise understanding of the physical properties of the analyzed objects and the physical environment in which they exist (Thijssen, 2007). Furthermore, such methods require a thorough grasp of relevant physical principles and relationships. With the rise of deep learning, connectionist machine learning approaches have demonstrated outstanding performance across various fields, bypassing the need for acquiring intricate prior knowledge and rules about the internal complexities of systems through a data-driven approach (Dong et al., 2021). Deep learning has simplified many complex problems in different domains and made previously unsolvable problems tractable. In recent years, the emergence of the Graph Neural Network (GNN) based physics simulation methods has opened new avenues for addressing rigid body dynamics analysis problems using deep learning (Sanchez-Gonzalez et al., 2020; Pfaff et al., 2021). Such approaches effectively extend neural network methods from traditional domains such as images, videos, and text into the realm of real-world physical analysis. Leveraging GNNs, such methods analyze the motion states of objects modeled as graphs, achieving commendable results in related methodologies.

However, the current GNN-based methods concerning rigid body dynamic analysis are still limited to analyzing relatively simple and ideal scenarios, often focusing on the analysis of common geometric shapes (Allen et al., 2023a; Bhattoo et al., 2022; Sanchez-Gonzalez et al., 2020; Han et al., 2022a; Bhattoo et al., 2022; Bishnoi et al., 2023; Allen et al., 2023b; Rubanova et al., 2021). This limitation hinders the practical application of such methods in industrial settings. In industrial environments, complex mechanical structures often involve numerous interconnected components, including intricate force relationships within rigid bodies and the resulting mutual influences between objects. Presently, GNN-based methods encounter significant challenges in addressing such problems, primarily because force-related data for objects is often challenging to observe and obtain (Iglberger & Rüde, 2011). As a result, these methods typically rely solely on positional and velocity information of objects for analysis. The difficulty in acquiring training data relevant to forces, coupled with

<sup>1</sup>Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

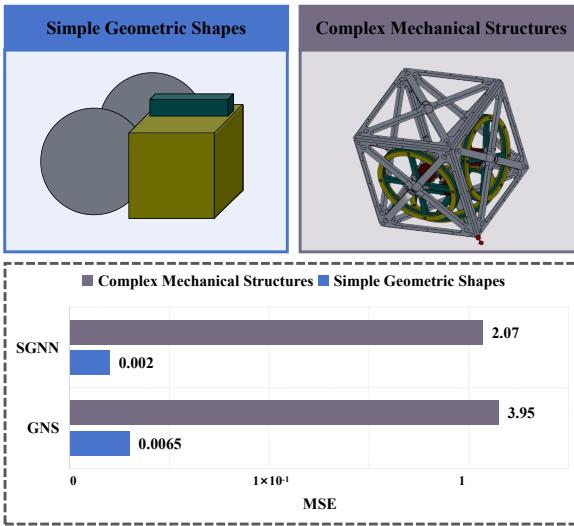


Figure 1. The comparison between the complex mechanical body scenarios we utilized and the general geometric body scenarios employed by other methods, the following illustrates the performance of two GNN-based Physical Dynamics Simulators in different scenes. From the figure, it is evident that both SGNN (Han et al., 2022a) and GNS (Sanchez-Gonzalez et al., 2020) methods exhibit significantly higher Mean Squared Error (MSE) in their predictive results for the dynamic analysis of complex mechanical structures compared to simple geometric shapes.

the necessity for force relationship analysis, places these methods in a dilemma when applied to rigid body dynamics analysis with complex mechanical structures. The comparisons and experiments within Figure 1 validate such a conclusion, demonstrating a significant degradation in the performance of the relevant GNN-based methods for the analysis of complex mechanical structures.

To address this challenge, we endeavor to draw inspiration from traditional multibody dynamics analysis methods and incorporate relevant techniques. It is noteworthy that while GNN-based dynamic analysis methods can assist in circumventing the need for acquiring intricate prior knowledge and rules about the internal complexities of systems, the output results must adhere to the principles and rules of the physical world. Some researchers argue that a purely data-driven learning paradigm may struggle to achieve optimal results for physical problems (Bhattoo et al., 2022; Han et al., 2022a; Bishnoi et al., 2023). Therefore, we aim to integrate the principles of traditional multibody dynamics analysis with GNN-based simulators, incorporating prior constraints, to address the challenges associated with rigid body dynamics analysis in complex mechanical structures.

Building upon the aforementioned analysis, we propose a ***Multibody Dynamics Guided GNN Simulator***, dubbed MDGS, to conduct a more thorough analysis of rigid body dynamics scenarios involving intricate mechanical struc-

tures. Initially, we introduce a graph modeling pattern for efficient analysis of forces in physical systems, tailored for multibody dynamics. Subsequently, we introduce calculations related to forces, encompassing both contact and non-contact forces, into the methodology. These force calculations are performed using neural networks without the introduction of additional data beyond conventional GNN-based physical dynamics simulators. Then, the computed forces are integrated into the succeeding analyses to adapt to complex mechanical scenarios. Furthermore, we imposed constraints on the model output based on the Kane equations tailored for addressing multibody dynamics analysis problems. This ensures the accuracy and rationality of the model's computed results. The major contributions are as follows.

- We introduce a novel method, referred to as MDGS, to incorporate force computation into GNN-based simulations for the analysis of rigid body systems. This approach enhances the treatment of multibody dynamics scenarios, expanding the potential applications of such methods in industrial settings.
- We imposed physical constraints on our method based on Kane's equations, offering a more precise analysis of rigid body dynamics. These constraints are derived from rigorous theoretical analysis and proof.
- We provide the implementations of MDGS for multi-body dynamics simulation tasks and tested on multiple scenarios, the results consistently showcase the outstanding performance of MDGS.

## 2. Related Works

### 2.1. GNN-based Physical Dynamics Simulators

GNN-based physical dynamics simulators (Sanchez-Gonzalez et al., 2020) leverages GNNs (Scarselli et al., 2009; Kipf & Welling, 2017) to simulate predictive representations for graph-modeled physical systems. Such approaches find widespread applications in computing atomic forces (Hu et al., 2021), simulating particle-modeled and mesh-modeled physical systems (Allen et al., 2023b; Rubanova et al., 2021; Li et al., 2019; Sanchez-Gonzalez et al., 2020; Pfaff et al., 2021; Han et al., 2022b). In recent research, some methods (Han et al., 2022a; Bhattoo et al., 2022; Bishnoi et al., 2023) have proposed incorporating prior knowledge from relevant domains into the GNN-based physical dynamics simulator to enhance its performance. This includes integrating concepts such as Subequivariant and Lagrangian equations into the model framework, making the model more closely aligned with real-world physical systems. Additionally, some approaches (Linkerhägner et al., 2023) optimize the model by collecting data in real-

world environments as supplementary information. Our method introduces force analysis between physical entities in rigid body dynamics scenarios into the system and employs Kane's equations (Kane & Levinson, 1985) to impose constraints on the system output, ensuring a more realistic simulation performance.

## 2.2. Dynamics Analysis

The framework of traditional mechanical analysis relies on either force-based or energy-based approaches(Wyk et al., 2022) for rigid body dynamics analysis. Energy-based approaches involve the computation of all the unknown energy based on the equations of equilibrium and hence are cumbersome for large structures(Finzi et al., 2020). The force-based multibody dynamics model solves the forward kinematics and inverse dynamics of the main mechanisms by employing two recursive algorithms centered around the Newton–Euler formulation(Gonçalves et al., 2023). And some methods use ODEs to calculate multibody corresponding conditions to approximate an optimal solution(Schubert et al., 2023; Nada & Bayoumi, 2023; Zhang et al., 2023). We amalgamate select concepts from traditional rigid body dynamics analysis with the newly introduced GNN-based approach to achieve improved and simplified predictions.

## 3. Methodology

### 3.1. Preliminary

#### 3.1.1. GNN-BASED SIMULATORS

GNN-based simulators predict the dynamical states of physical systems. Consider a physical system comprising  $M$  elements, collectively forming  $N$  objects. At time  $t$ , GNN-based physical dynamics simulators model the aforementioned physical system using a graph  $G^{(t)} = \{\mathcal{V}^{(t)}, \mathcal{E}^{(t)}\}$ , where the node set  $\mathcal{V}^{(t)}$  of graph  $G^{(t)}$  represents different elements constituting objects. The feature  $\mathbf{z}_i^{(t)}$  of node  $i$  in the set  $\mathcal{V}^{(t)}$  includes position information  $\vec{x}_i^{(t)}$ , velocity information  $\vec{v}_i^{(t)}$ , and  $\mathbf{k}_i$  representing the properties of the object to which the element belongs. In other words,  $\mathbf{z}_i^{(t)}$  is defined as the concatenation of these vectors:  $\mathbf{z}_i^{(t)} = [\vec{x}_i^{(t)} || \vec{v}_i^{(t)} || \mathbf{k}_i]$ , where ‘||’ denotes vector concatenation along the first dimension.  $\mathbf{Z}^{(t)} = \{\mathbf{z}_i^{(t)}\}_{i=1}^M$  denotes the set of all  $\mathbf{z}_i^{(t)}$ . The edge set  $\mathcal{E}^{(t)}$  of graph  $G^{(t)}$  characterizes the connectivity relationships between elements. For instance, some approaches (Han et al., 2022a; Bhatoo et al., 2022) judge that if the distance between element  $i$  and  $j$  is less than a certain threshold, the edge  $(i, j)$  is connected. Subsequently, GNN-based simulator utilizes the GNN model  $g(\cdot)$  to predict the values of  $\mathbf{Z}^{(t+1)}$  based on  $G^{(t)}$ , such process can

be formalized as follows:

$$\mathbf{Z}^{(t+1)} = g(G^{(t)}). \quad (1)$$

#### 3.1.2. KANE'S EQUATION

Kane's equation (Kane & Levinson, 1985) is a mathematical formulation used in the field of multibody dynamics to describe the motion of interconnected rigid bodies. For a nonholonomic system  $\mathcal{S}$ , the number of independently variable motions or deformations is referred to as degrees of freedom. For each degree of freedom, there exists a corresponding generalized velocity. These velocity vectors associated with generalized coordinates are typically denoted by  $\dot{q}$ , where  $q$  represents the generalized coordinates. Specifically, with respect to the generalized coordinate  $q_i$  and its corresponding generalized velocity  $\dot{q}_i$ , if  $\dot{q}_i$  is independent of the other velocities in the system, meaning that it cannot be expressed or derived from other velocities in the system's description, then  $\dot{q}_i$  is referred to as an independent velocity. Furthermore, the partial velocity  $u_{ij}' = \frac{\partial q_i}{\partial \dot{q}_j}$  denotes the partial derivative of the velocity with respect to coordinate  $q_i$  when coordinate  $q_j$  is varied, within the context of a multi-coordinate system.

With the aforementioned concepts, we could define the general active force according to the  $\gamma$ -th independent velocity of  $\mathcal{S}$  as follows:

$$K_\gamma = \sum_{i=1}^n \vec{F}_i \cdot \vec{u}_{i\gamma}', \quad (2)$$

where  $\vec{F}_i$  denotes the force acting on the  $i$ -th mass point in the system  $\mathcal{S}$ , and  $\vec{u}_{i\gamma}'$  represents its partial velocity with respect to the  $\gamma$ -th independent velocity,  $n$  is the number of mass points. Likewise, we could define the generalized inertia forces according to the  $\gamma$ -th independent velocity as follows:

$$K_\gamma^* = \sum_{i=1}^n \left( -m_i \ddot{r}_i \right) \cdot \vec{u}_{i\gamma}', \quad (3)$$

where  $m_i$  represents mass, while  $\vec{r}_i$  represents the vector radius. The Kane's equation can then be formulated as follows:

$$K_\gamma^* + K_\gamma = 0, \gamma = \{1, 2, \dots, d\}, \quad (4)$$

$d$  is the number of independent velocities.

### 3.2. Method

#### 3.2.1. GRAPH CONSTRUCTION

Two commonly used object partitioning approaches in the field of physical simulation are point clouds (Gross et al.,

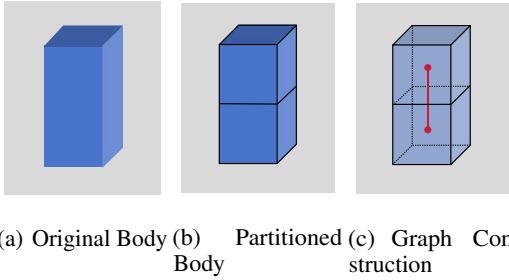


Figure 2. Illustration of the Graph Construction Process.

2002) and finite elements (Pasciak, 1995), both of which have been widely applied in GNN-based simulators. Point clouds essentially involve decomposing objects into indivisible elements using point cloud construction techniques. On the other hand, the finite element method divides objects into individual polyhedral elements, where the edges and vertices of these elements constitute the graph for GNN-based simulators. In our approach, we integrate the above two methods by employing the finite element method to partition rigid bodies into different elements and designate the centroid of each element as a node in the graph. In the specific implementation, we assume uniform density within each element and utilize its geometric center as its centroid. We believe that this approach maximally preserves the kinematic properties of objects while aligning more closely with the commonly used analytical paradigm in multibody dynamics (Kane & Levinson, 1985). Figure 2 provides an illustrative example of our graph construction method. As illustrated in Figure 2(c), the centroids serve as nodes in the graph, and the centroids of adjacent elements are connected by edges. The specific methodology for edge construction will be elaborated in detail in Section 3.2.2.

### 3.2.2. CALCULATING FORCES

In the realm of multibody dynamics, the magnitude of forces acting on an object plays a crucial role in determining its state. However, due to the inherent difficulty in measuring and observing the force conditions of objects, existing GNN-based simulators do not currently incorporate force information into the prediction computation process. To address this issue, we employ neural networks to predict the forces acting on rigid bodies and integrate these predicted values into the model. Specifically, we categorize the forces acting on objects into two components: contact force and non-contact force.

For contact forces, such as frictional and collision forces, we propose the construction of a contact-oriented graph  $\bar{G}^{(t)} = \{\mathcal{V}^{(t)}, \bar{\mathcal{E}}^{(t)}\}$ . In  $\bar{G}^{(t)}$ ,  $\mathcal{V}^{(t)}$  represents the centroids corresponding to distinct elements, with attributes comprising the input positional and velocity information of these elements at time  $t$ . Additionally,  $\mathcal{V}^{(t)}$  includes property

about the particular object each element is affiliated to, providing the information concerning the whole object.  $\bar{\mathcal{E}}^{(t)}$  comprises only those edges determined by the existence of contact between elements. As depicted in Figure 4(a), if the distance between the centroids of two elements, denoted as  $i$  and  $j$ , is less than  $r$ , and they belong to different objects, we consider these elements to be in contact, and the edge  $(i, j)$  is established,  $r$  is a threshold defined by a hyperparameter. In this manner, we derive a graph  $\bar{G}^{(t)}$  specifically illustrating contact relationships. Subsequently, we employ  $\bar{G}^{(t)}$  to predict the force set  $\bar{F}^{(t)}$ , where  $\bar{F}^{(t)} \in \mathbb{R}^{N \times 3}$ . The  $i$ -th vector  $\bar{f}_i^{(t)}$  of  $\bar{F}^{(t)}$  denotes the contact forces acting upon the centroid of the  $i$ -th element. Formally, we have:

$$\bar{F}^{(t)} = g^{\bar{F}}(\bar{G}^{(t)}), \quad (5)$$

where  $g^{\bar{F}}(\cdot)$  denotes a GNN specifically designed for contact force prediction.  $\bar{f}_i^{(t)}$  will be set to the zero vector if  $i$  is not an endpoint of any edge in  $\bar{\mathcal{E}}^{(t)}$ .

For non-contact forces, such as gravity, we propose constructing the object-oriented graph  $\tilde{G}^{(t)} = \{\mathcal{V}^{(t)}, \tilde{\mathcal{E}}^{(t)}\}$ , where  $\mathcal{V}^{(t)}$  remains identical to that of  $\bar{G}^{(t)}$ . As illustrated in Figure 4(b), if two elements  $i$  and  $j$  belong to the same object and share faces between them, then an edge  $(i, j)$  is established within  $\tilde{\mathcal{E}}^{(t)}$ . Subsequently, we predict the non-contact force set  $\tilde{F}^{(t)}$ . Formally, this prediction is expressed as:

$$\tilde{F}^{(t)} = g^{\tilde{F}}(\tilde{G}^{(t)}), \quad (6)$$

where  $g^{\tilde{F}}(\cdot)$  denotes a GNN specifically designed for non-contact force prediction. The final force set is obtained by combining the contact and non-contact force sets:

$$F^{(t)} = \{f_i^{(t)} : f_i^{(t)} = \bar{f}_i^{(t)} + \tilde{f}_i^{(t)}\}, \quad (7)$$

where  $\bar{f}_i^{(t)}$  and  $\tilde{f}_i^{(t)}$  represent the  $i$ -th force vector of  $\bar{F}^{(t)}$  and  $\tilde{F}^{(t)}$  respectively.

### 3.2.3. MAKING PREDICTIONS

Subsequently, we predict  $G^{(t+1)}$  based on both  $G^{(t)}$  and  $F^{(t)}$ . We first attach the calculated force to the attribute  $z$  of each node. For the  $i$ -th node with attribute feature  $z_i$ , the new attribute feature  $z'_i$  is calculated as follows:

$$z'_i = [z_i || \lambda F_i^{(t)}], \quad (8)$$

where  $\lambda$  is a hyperparameter for controlling the magnitude of the influence of  $F^{(t)}$ . We denote the modified  $G^{(t)}$  as  $G'^{(t)}$  and calculate the prediction as follows:

$$Z^{(t+1)} = g(G'^{(t)}), \quad (9)$$

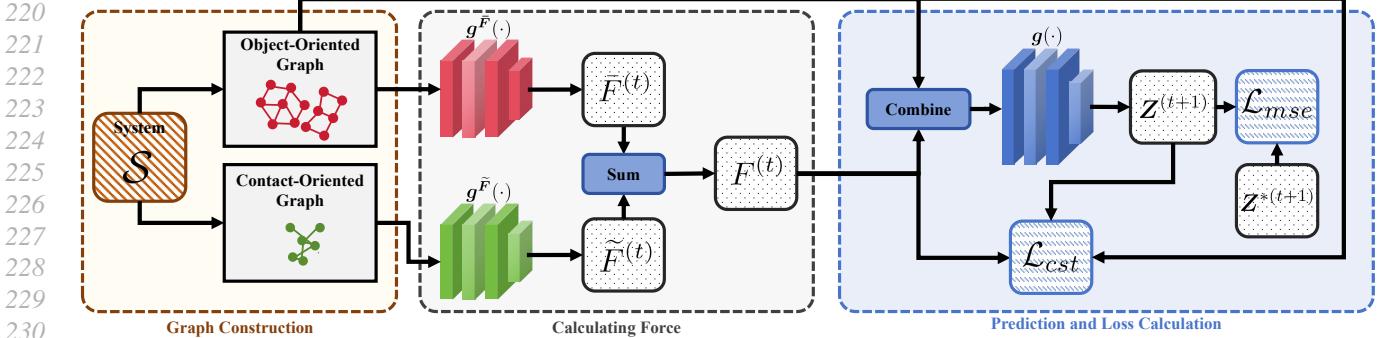
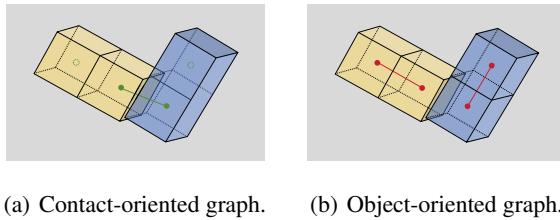


Figure 3. The framework of our proposed method. In the figure, the framework is segmented based on the stages of method execution, dividing it into three parts.



(a) Contact-oriented graph. (b) Object-oriented graph.

Figure 4. Illustration of contact-oriented graph and object-oriented graph.

where  $g(\cdot)$  denotes the GNN for state prediction. Motivated by (Allen et al., 2023a), we introduced a rigid body structural constraint module in our method. This module ensures the invariance of rigid body structures in the predicted results. Specifically, we calculate the relative positions of various nodes on the object based on the initial values. Subsequently, we re-align our predicted results to the original relative positions, ensuring that the rigid body structure remains unchanged. A detailed description of this module can be found in Appendix B.1.

$Z^{(t+1)}$  will then be utilized for calculating an MSE loss against the ground truth state  $Z^{*(t+1)}$  for the backpropagation of the whole framework:

$$\mathcal{L}_{mse} = \text{MSE}(Z^{(t+1)}, Z^{*(t+1)}). \quad (10)$$

#### 3.2.4. MODEL CONSTRAINTS BASED ON KANE'S EQUATION.

So far, we have successfully incorporated force computation to assist in predictions within the GNN-based simulators. However, such computation still faces challenges in ensuring accuracy. As discussed within section 3.1.2, Kane's equations provide a physical constraint for rigid body dynamical systems, and we aim to employ such constraints to refine our model, enhancing the analytical fidelity to physical reality.

Theoretically, a neural network capable of accurately predicting rigid body systems should adhere to the following

principles:

**Theorem 3.1.** Consider a rigid body dynamics system denoted as  $\mathcal{S}$  with  $d$  independent velocities. At time  $t$ , the system's state is represented as  $G^{(t)}$ , constructed according to the detailed graph construction procedure outlined in Section 3.2.1 and 3.2.2. For any Graph Neural Network (GNN) denoted as  $g(\cdot)$ , which demonstrates accurate prediction capabilities for  $F^{(t)}$  and  $Z^{(t+1)}$  based on  $G^{(t)}$ , the following equation must be satisfied:

$$\sum_i F_{\gamma,i}^{(t)} u_\gamma - \sum_i M_i \frac{1}{\delta} (g(G^{(t)})_{[v_\gamma,i]} - v_\gamma^{(t)}) u_\gamma + \Psi^{(t)} = 0, \forall \gamma \in \{1, 2, \dots, d\}, \quad (11)$$

where  $\gamma$  denote the index of independent velocity,  $F_{\gamma,i}^{(t)}$  is the predicted force value of element  $i$  along  $\gamma$ -th independent velocity,  $M_i$  is the mass of element  $i$ ,  $g(G^{(t)})_{[v_\gamma,i]}$  denote the predicted  $v_\gamma^{(t+1)}$  with  $g(\cdot)$ ,  $\delta$  is the time step length,  $\Psi^{(t)}$  is a disturbance term directly proportional to the sum of the radii from different mass points to the center of mass in each element.

The proof can be found in Appendix A.1. Theorem 3.1 furnishes us with a specific methodology for constraining the model. From theorem 3.1, it is evident that the constraints proposed in Equation 11 hold along any independent velocity. In practical implementation, we can determine the number of equations the model needs to satisfy based on the distinct independent velocities associated with different systems.

To further constrain the perturbation terms in Theorem 3.1, we propose the following theorem:

**Theorem 3.2.** Given the conditions within Theorem 3.1,

275 then at time  $t$ , the following inequality holds:

$$\begin{aligned} -\tau \sum_{i=1}^n (|\vec{R}_i^{(t)}|) \omega_\gamma - \sum_{i=1}^n \frac{2}{5} M_i \tau^2 \dot{\omega}_i^{(t)} &\leq \Psi^{(t)} \leq \\ \tau \sum_{i=1}^n (|\vec{R}_i^{(t)}|) \omega_\gamma + \sum_{i=1}^n \frac{2}{5} M_i \tau^2 \dot{\omega}_i^{(t)}, \end{aligned} \quad (12)$$

283 where  $\vec{R}_i^{(t)}$  denotes the interval force act on element  $i$ ,  $\tau$  is the max radius  $\tau$  among all elements,  $\omega_\gamma$  denotes unit angular velocity value along the direction of the  $\gamma$ -th independent velocity,  $\dot{\omega}_i^{(t)}$  represents the max angular acceleration of element  $i$ .

288 Based on Theorem 3.1 and Theorem 3.2, we proposed the  
289 following loss function for the upper bound of constrain:

$$\begin{aligned} \mathcal{L}_{sup} = \text{Relu} \left( \sum_i F_{\gamma,i}^{(t)} u_\gamma - \sum_i M_i \frac{1}{\delta} (g(G^{(t)})_{[v_\gamma,i]} \right. \\ \left. - v_\gamma^{(t)}) u_\gamma - \tau \sum_{i=1}^n (|\vec{R}_i^{(t)}|) \omega_\gamma - \sum_{i=1}^n \frac{2}{5} M_i \tau^2 \dot{\omega}_i^{(t)} \right), \quad (13) \end{aligned}$$

297 where  $F_{\gamma,i}^{(t)}$  and  $|\vec{R}_i^{(t)}|$  can be calculated with force prediction, while  $\dot{\omega}_i^{(t)}$  is calculated based on variant within coordinates,  $\text{Relu}(\cdot)$  denote the ReLU function,  $M_i$  is calculated based on the properties of the object, the details can be found in Appendix B.1. Similarly, we can calculate the loss function for the lower bound of constrain:

$$\begin{aligned} \mathcal{L}_{inf} = \text{Relu} \left( - \sum_i F_{\gamma,i}^{(t)} u_\gamma + \sum_i M_i \frac{1}{\delta} (g(G^{(t)})_{[v_\gamma,i]} \right. \\ \left. - v_\gamma^{(t)}) u_\gamma - \tau \sum_{i=1}^n (|\vec{R}_i^{(t)}|) \omega_\gamma - \sum_{i=1}^n \frac{2}{5} M_i \tau^2 \dot{\omega}_i^{(t)} \right), \quad (14) \end{aligned}$$

310 The overall loss for constrain is:

$$\mathcal{L}_{cst} = \mathcal{L}_{sup} + \mathcal{L}_{inf}. \quad (15)$$

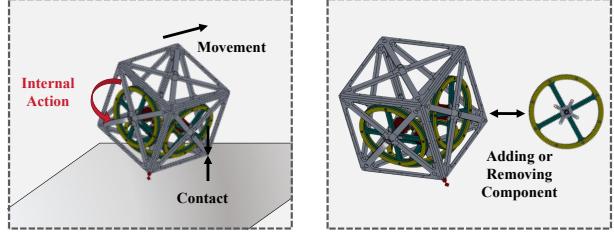
313  $\mathcal{L}_{cst}$  will only be used to update  $g^{\bar{F}}$  and  $g^{\tilde{F}}$ .  $\mathcal{L}_{cst}$  and  $\mathcal{L}_{mse}$  are summed up for the final loss. The overall framework is  
314 illustrated within Figure 3.

## 4. Experiments

### 4.1. Environmental Setup

#### 4.1.1. DATASET AND SYSTEMS

317 To validate our methodology, we conduct comprehensive  
318 testing on diverse datasets derived from four distinct  
319 mechanical systems. Specifically, we utilized the Cubli robot  
320 (Gajamohan et al., 2012) (a metallic mechanical cube capable  
321 of performing various maneuvers using three flywheels),  
322 complex rotating bodies (intricate mechanical structures  
323 featuring components such as bearings and hinges), vehicles  
324 (3D models of cars and trucks), and space robots (Wang et al.,  
325 2022) (robots equipped with mechanical arms for executing  
326 specific tasks in space environments). Utilizing Mujoco (Todorov et al.,  
327 2012) for physics-based modeling, we systematically vary  
328 motion patterns and mechanical components to generate six  
329 datasets comprising 2200 trajectories, with each trajectory  
330 consisting of 120 time steps, enabling a thorough evaluation  
331 of the robustness of our approach. We describe the  
332 construction principles within Figure 5. Specifically, in the  
333 context of the Cubli robot, we have systematically varied  
334 the number of reaction wheels, creating distinct rigid body  
335 systems comprising single, double, and triple flywheels. For  
336 intricate rotational bodies, we have constructed three  
337 structures with no hinge, hinges on both sides, and hinges on the  
338 central position, respectively. All utilized physical systems  
339 are demonstrated within Appendix B.3



(a) Our dataset incorporates interactions between complex mechanical bodies and the environment, the overall movement of mechanical bodies, and the internal actions of components within the mechanical bodies.  
(b) We modify the attributes and motion patterns of mechanical bodies by adding or removing mechanical components within them. Such modifications allow us to test the performances of the algorithms across various scenarios.

Figure 5. The underlying principles behind the construction of our dataset.

(mechanical cars operating under different conditions and environments), and space robots (Wang et al., 2022) (robots equipped with mechanical arms for executing specific tasks in space environments). Utilizing Mujoco (Todorov et al., 2012) for physics-based modeling, we systematically vary motion patterns and mechanical components to generate six datasets comprising 2200 trajectories, with each trajectory consisting of 120 time steps, enabling a thorough evaluation of the robustness of our approach. We describe the construction principles within Figure 5. Specifically, in the context of the Cubli robot, we have systematically varied the number of reaction wheels, creating distinct rigid body systems comprising single, double, and triple flywheels. For intricate rotational bodies, we have constructed three structures with no hinge, hinges on both sides, and hinges on the central position, respectively. All utilized physical systems are demonstrated within Appendix B.3

#### 4.1.2. BASELINES

We conduct a comparative analysis of our proposed MDGS against various baselines, including GNS (Sanchez-Gonzalez et al., 2020), SGNN (Han et al., 2022a), LGNS (Bhattoo et al., 2022), and HGNS (Bishnoi et al., 2023). Specifically, GNS employs a pure Graph Neural Network (GNN) for simulating physical systems, while SGNN introduces the concept of subequivariant in the physical system into the GNN-based physical dynamics simulator. Furthermore, LGNS and HGNS leverage Lagrangian equations and Hamiltonian mechanics, respectively, to guide and constrain the GNN model, enhancing its capability to simulate physical reality. Apart from GNS, the other three methods incorporate varying degrees of prior knowledge from physics to reinforce the models. Through comparisons with these baselines, we can effectively analyze the algorithmic performance of our approach in the dynamic analysis of complex mechanical bodies.

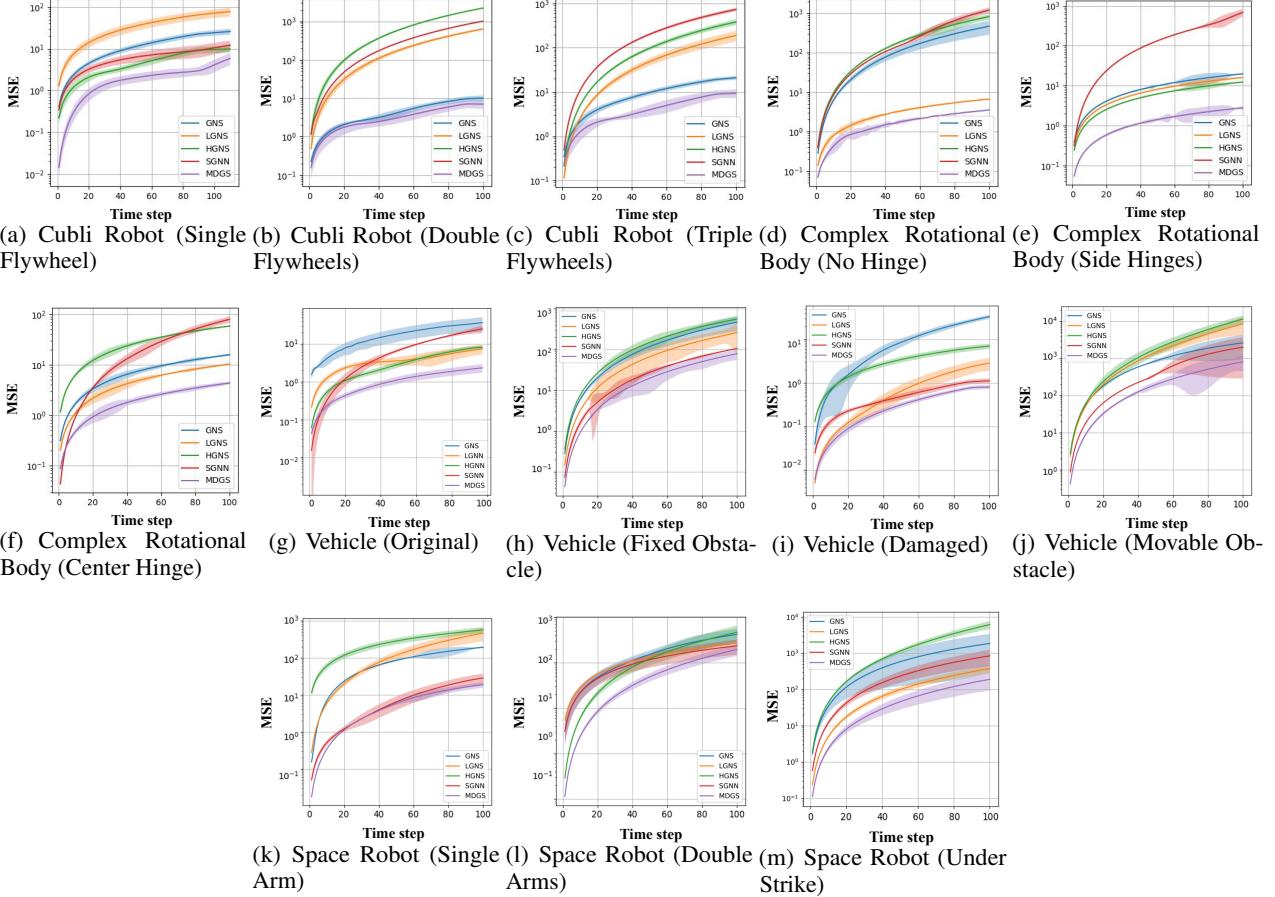


Figure 6. Experimental results on precision evaluation.

#### 4.1.3. EXPERIMENTAL SETTINGS

Due to the widely applicable enhancement provided by our designed rigid body structural constraint module on the accuracy of the GNN-based simulator in predicting rigid body problems, to ensure fairness in comparisons, we incorporated this module into other baselines as well, elevating their performance equivalently. Simultaneously, in Section 4.3.1, we presented the impact of adding or removing the rigid body structural constraint module on other baselines. Our experiments utilized MSE between predicted and ground truth values as the metric for evaluating the performance of each model. To demonstrate statistical significance, we followed the methodology outlined in (Han et al., 2022a), conducting each experiment 10 times and averaging the results. Specific hyperparameters and the backbones are detailed in Appendix B.2.

## 4.2. Performance Evaluation

### 4.2.1. PRECISION

Building upon the aforementioned experimental setup, we conducted performance testing experiments for both MDGS

and other baseline algorithms across a total of six datasets. Figure 6 presents the experimental results. It is evident from the results that our proposed MDGS consistently outperforms the majority of comparisons. Figure 8 demonstrated the visualized results. We provide results represented in digital form in the Appendix B.4.

### 4.2.2. GENERALIZATION

We also conducted comparisons on the generalization performance of various methods. Using the complete Cubli Robot with three flywheels, we introduced variations to different attributes of the system. Subsequently, we examined whether the methods trained on the original Cubli Robot dataset could accurately predict the changed system and observed variations in algorithm prediction errors on the new test set. The test results are presented in the Figure ???. It is evident that our method achieves the best performance. Additionally, an interesting observation is that, apart from MDGS, other methods incorporating prior knowledge of the physical system perform poorly in generalization experiments.

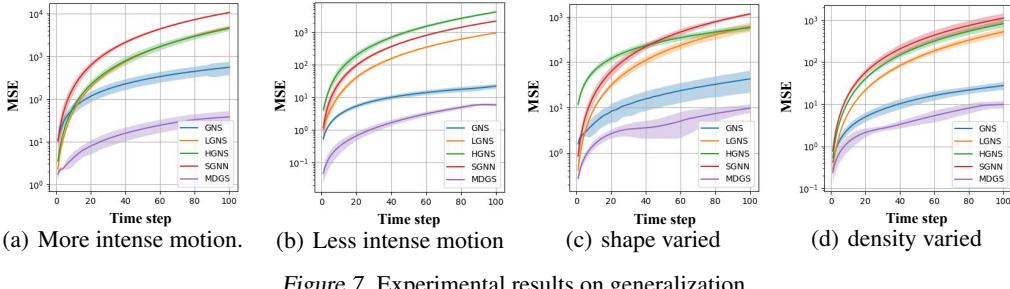


Figure 7. Experimental results on generalization.

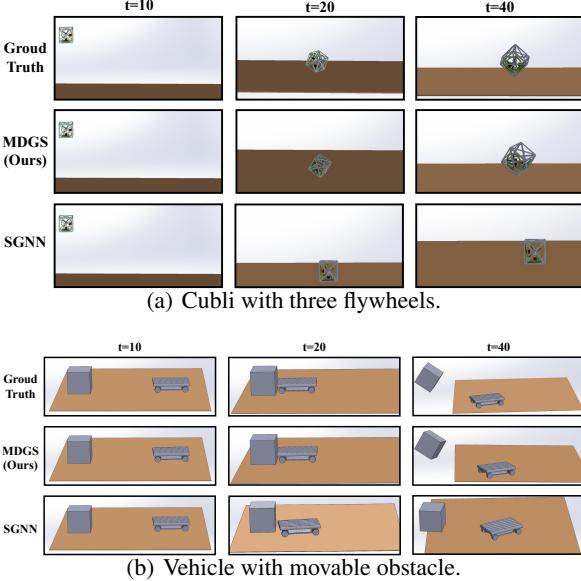


Figure 8. Visualized experimental results.

### 4.3. In-Depth Study

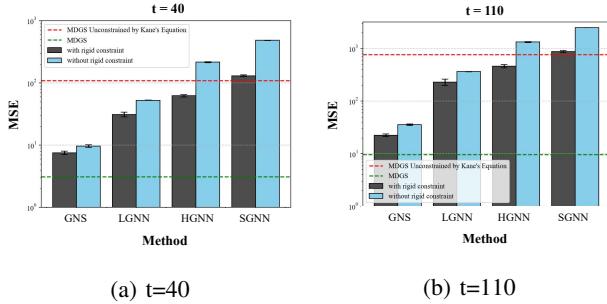


Figure 9. Experimental results of baselines with and without structural constraint module.

#### 4.3.1. STRUCTURAL CONSTRAINT MODULE ANALYSIS

Validation experiments were conducted to analyze the capabilities of the proposed structural constraint module. A comparative study was performed on the dataset of the Cubli robot with all three flywheels, evaluating the performance of methods with and without the structural constraint module. In Figure 9, the performance of the respective models at

different time steps is illustrated. It can be observed that all methods exhibit improved performance, i.e., lower MSE, after incorporating the structural constraint module. This experimental result demonstrates the effectiveness of the proposed structural constraint module in enhancing the accuracy of model predictions in rigid body dynamics analysis, providing additional constraint information about the model structure for further predictions. Simultaneously, the experiment validates the necessity of integrating the structural constraint module when comparing our approach with other baselines. We provide results represented in digital form in the [Appendix B.4](#).

#### 4.3.2. ABLATION STUDY

For a more in-depth analysis of our method, we conducted a series of ablation experiments. Specifically, we modified the constraints of Kane's equations to create two baseline models, namely MDGS-NC and MDGS-SC. MDGS-NC removes all constraints from Kane's equations, while MDGS-SC retains only an independent velocity constraint. The experimental results are depicted in Figure 10. It is evident that the constraints of Kane's equations indeed contribute to the improved performance of our method. Table 3 further illustrates the comparison between MDGS-NC, MDGS-SC, and other baselines.

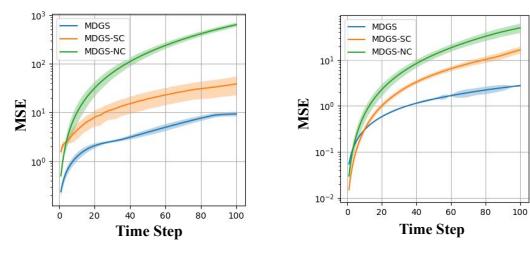


Figure 10. Ablation experiments results.

## 5. Conclusion

In this paper, we propose a novel approach, MDGS, to incorporate force analysis into GNN-based rigid body system

440 simulators, aiming to enhance the accuracy of such methods  
 441 in complex mechanical scenarios. We substantiate the de-  
 442 sign of MDGS through theoretical foundations and proofs  
 443 and validate its performance through a series of experiments.  
 444 We contend that MDGS can provide assistance to some ex-  
 445 tent in the application of GNN-based rigid body system  
 446 simulators in industrial settings.

## 447 Impact Statements

450 This paper presents work whose goal is to advance the field  
 451 of Machine Learning. There are many potential societal  
 452 consequences of our work, none of which we feel must be  
 453 specifically highlighted here.

## 454 References

455 Allen, K. R., Guevara, T. L., Rubanova, Y., Stachenfeld,  
 456 K., Sanchez-Gonzalez, A., Battaglia, P., and Pfaff, T.  
 457 Graph network simulators can learn discontinuous, rigid  
 458 contact dynamics. In *Conference on Robot Learning*, pp.  
 459 1157–1167. PMLR, 2023a.

460 Allen, K. R., Rubanova, Y., Lopez-Guevara, T., Whitney,  
 461 W., Sanchez-Gonzalez, A., Battaglia, P. W., and Pfaff,  
 462 T. Learning rigid dynamics with face interaction graph  
 463 networks. In *The Eleventh International Conference on  
 464 Learning Representations, ICLR 2023, Kigali, Rwanda,  
 465 May 1-5, 2023*. OpenReview.net, 2023b. URL <https://openreview.net/pdf?id=J7Uh781A05p>.

466 Bhattoo, R., Ranu, S., and Krishnan, N. M. A. Learning  
 467 articulated rigid body dynamics with lagrangian graph  
 468 neural network. In Koyejo, S., Mohamed, S., Agarwal,  
 469 A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances  
 470 in Neural Information Processing Systems 35: Annual  
 471 Conference on Neural Information Processing Systems  
 472 2022, NeurIPS 2022, New Orleans, LA, USA, November  
 473 28 - December 9, 2022*, 2022.

474 Bishnoi, S., Bhattoo, R., Jayadeva, J., Ranu, S., and Krish-  
 475 nan, N. A. Learning the dynamics of physical systems  
 476 with hamiltonian graph neural networks. In *ICLR 2023  
 477 Workshop on Physics for Machine Learning*, 2023.

478 Brach, R. M. *Mechanical impact dynamics: rigid body  
 479 collisions*. Brach Engineering, LLC, 2007.

480 Dong, S., Wang, P., and Abbas, K. A survey on deep  
 481 learning and its applications. *Computer Science Review*,  
 482 40:100379, 2021.

483 Featherstone, R. *Rigid body dynamics algorithms*. Springer,  
 484 2014.

485 Finzi, M., Wang, K. A., and Wilson, A. G. Simplifying  
 486 hamiltonian and lagrangian neural networks via explicit  
 487 constraints. In Larochelle, H., Ranzato, M., Hadsell, R.,  
 488 Balcan, M., and Lin, H. (eds.), *Advances in Neural In-  
 489 formation Processing Systems 33: Annual Conference on  
 490 Neural Information Processing Systems 2020, NeurIPS  
 491 2020, December 6-12, 2020, virtual*, 2020.

492 Gajamohan, M., Merz, M., Thommen, I., and D’Andrea,  
 493 R. The cubli: A cube that can jump up and balance. In  
 494 *2012 IEEE/RSJ International Conference on Intelligent  
 495 Robots and Systems, IROS 2012, Vilamoura, Algarve,  
 496 Portugal, October 7-12, 2012*, pp. 3722–3727. IEEE,  
 497 2012. doi: 10.1109/IROS.2012.6385896. URL <https://doi.org/10.1109/IROS.2012.6385896>.

498 Gonçalves, F., Ribeiro, T., Ribeiro, A. F., Lopes, G., and  
 499 Flores, P. Multibody model of the human-inspired robot  
 500 charmie. *Multibody System Dynamics*, pp. 1–28, 2023.

501 Gross, M. H., Pfister, H., Zwicker, M., Pauly, M., Stam-  
 502 mingger, M., and Alexa, M. Point based computer graph-  
 503 ics. In Coquillart, S. and Müller, H. (eds.), *23rd Annual  
 504 Conference of the European Association for Computer  
 505 Graphics, Eurographics 2002 - Tutorials, Saarbrücken,  
 506 Germany, September 2-6, 2002*. Eurographics Associa-  
 507 tion, 2002. doi: 10.2312/EGT.20021060. URL <https://doi.org/10.2312/egt.20021060>.

508 Han, J., Huang, W., Ma, H., Li, J., Tenenbaum, J., and Gan,  
 509 C. Learning physical dynamics with subequivariant graph  
 510 neural networks. In Koyejo, S., Mohamed, S., Agarwal,  
 511 A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances  
 512 in Neural Information Processing Systems 35: Annual  
 513 Conference on Neural Information Processing Systems  
 514 2022, NeurIPS 2022, New Orleans, LA, USA, November  
 515 28 - December 9, 2022*, 2022a.

516 Han, X., Gao, H., Pfaff, T., Wang, J., and Liu, L. Predict-  
 517 ing physics in mesh-reduced space with temporal atten-  
 518 tion. In *The Tenth International Conference on Learn-  
 519 ing Representations, ICLR 2022, Virtual Event, April  
 520 25-29, 2022*. OpenReview.net, 2022b. URL <https://openreview.net/forum?id=XctLdNfCmP>.

521 Hu, W., Shuaibi, M., Das, A., Goyal, S., Sriram, A.,  
 522 Leskovec, J., Parikh, D., and Zitnick, C. L. Forcenet:  
 523 A graph neural network for large-scale quantum calcula-  
 524 tions. In *ICLR 2021 SimDL Workshop*, volume 20, 2021.

525 Huston, R. L. and Passerello, C. On multi-rigid-body sys-  
 526 tem dynamics. *Computers & Structures*, 10(3):439–446,  
 527 1979.

528 Iglberger, K. and Rüde, U. Large-scale rigid body simula-  
 529 tions. *Multibody system dynamics*, 25:81–95, 2011.

530 Kane, T. R. and Levinson, D. A. *Dynamics, theory and  
 531 applications*. McGraw Hill, 1985.

- 495 Kipf, T. N. and Welling, M. Semi-supervised classification  
 496 with graph convolutional networks. In *5th International*  
 497 *Conference on Learning Representations, ICLR 2017,*  
 498 *Toulon, France, April 24-26, 2017, Conference Track*  
 499 *Proceedings*. OpenReview.net, 2017. URL <https://openreview.net/forum?id=SJU4ayYgl>.
- 500
- 501 Landau, R. H., Mejía, M. J. P., and Bordeianu, C. C. *A*  
 502 *survey of computational physics: introductory computa-*  
 503 *tional science*. Princeton University Press, 2008.
- 504
- 505 Li, Y., Wu, J., Tedrake, R., Tenenbaum, J. B., and Torralba,  
 506 A. Learning particle dynamics for manipulating rigid  
 507 bodies, deformable objects, and fluids. In *7th Interna-*  
 508 *tional Conference on Learning Representations, ICLR*  
 509 *2019, New Orleans, LA, USA, May 6-9, 2019*. OpenRe-  
 510 view.net, 2019. URL <https://openreview.net/forum?id=rJgbSn09Ym>.
- 511
- 512 Linkerhägner, J., Freymuth, N., Scheikl, P. M., Mathis-  
 513 Ullrich, F., and Neumann, G. Grounding graph network  
 514 simulators using physical sensor observations. In *The*  
 515 *Eleventh International Conference on Learning Repre-*  
 516 *sentations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*.  
 517 OpenReview.net, 2023. URL <https://openreview.net/pdf?id=jsZsEd8VEY>.
- 518
- 519 Mirtich, B. V. *Impulse-based dynamic simulation of rigid*  
 520 *body systems*. University of California, Berkeley, 1996.
- 521
- 522 Nada, A. and Bayoumi, M. Development of a constraint  
 523 stabilization method of multibody systems based on fuzzy  
 524 logic control. *Multibody System Dynamics*, pp. 1–33,  
 525 2023.
- 526
- 527 Pasciak, J. E. The mathematical theory of finite element  
 528 methods (susanne c. brenner and l. ridgway scott). *SIAM*  
 529 *Rev.*, 37(3):472–473, 1995. doi: 10.1137/1037111. URL  
 530 <https://doi.org/10.1137/1037111>.
- 531
- 532 Pfaff, T., Fortunato, M., Sanchez-Gonzalez, A., and  
 533 Battaglia, P. W. Learning mesh-based simulation  
 534 with graph networks. In *9th International Confer-*ence on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021**. OpenReview.net,  
 535 2021. URL [https://openreview.net/forum?id=roNqYL0\\_XP](https://openreview.net/forum?id=roNqYL0_XP).
- 536
- 537 Rubanova, Y., Sanchez-Gonzalez, A., Pfaff, T., and  
 538 Battaglia, P. W. Constraint-based graph network sim-  
 539 ulator. *CoRR*, abs/2112.09161, 2021. URL <https://arxiv.org/abs/2112.09161>.
- 540
- 541 Sanchez-Gonzalez, A., Godwin, J., Pfaff, T., Ying, R.,  
 542 Leskovec, J., and Battaglia, P. W. Learning to sim-  
 543 ulate complex physics with graph networks. In *Pro-*ceedings of the 37th International Conference on Ma-**
- 544
- chine Learning, ICML 2020, 13-18 July 2020, Virtual Event, volume 119 of *Proceedings of Machine Learning Research*, pp. 8459–8468. PMLR, 2020. URL <http://proceedings.mlr.press/v119/sanchez-gonzalez20a.html>.
- 545 Sauer, J. and Schömer, E. A constraint-based approach to  
 546 rigid body dynamics for virtual reality applications. In *Proceedings of the ACM symposium on Virtual reality*  
 547 *software and technology*, pp. 153–162, 1998.
- 548
- 549 Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., and  
 550 Monfardini, G. The graph neural network model. *IEEE*  
 551 *Trans. Neural Networks*, 20(1):61–80, 2009. doi: 10.  
 552 1109/TNN.2008.2005605. URL <https://doi.org/10.1109/TNN.2008.2005605>.
- 553
- Schubert, M., Sato Martín de Almagro, R. T., Nachbagauer,  
 554 K., Ober-Blöbaum, S., and Leyendecker, S. Discrete  
 555 adjoint method for variational integration of constrained  
 556 odes and its application to optimal control of geometri-  
 557 cally exact beam dynamics. *Multibody System Dynamics*,  
 558 pp. 1–28, 2023.
- 559
- Silva, M., Ambrosio, J., and Pereira, M. Biomechanical  
 560 model with joint resistance for impact simulation. *Muti-*  
 561 *body System Dynamics*, 1(1):65–84, 1997.
- 562
- Szabó, B. and Babuška, I. Finite element analysis: Method,  
 563 verification and validation. 2021.
- 564
- Thijssen, J. *Computational physics*. Cambridge university  
 565 press, 2007.
- 566
- Todorov, E., Erez, T., and Tassa, Y. Mujoco: A physics  
 567 engine for model-based control. In *2012 IEEE/RSJ Inter-*national Conference on Intelligent Robots and Sys-**  
 568 *tems, IROS 2012, Vilamoura, Algarve, Portugal, Oc-*  
 569 *tober 7-12, 2012*, pp. 5026–5033. IEEE, 2012. doi:  
 570 10.1109/IROS.2012.6386109. URL <https://doi.org/10.1109/IROS.2012.6386109>.
- 571
- Tong, Y. Y. Vehicle dynamic simulations based on flexi-  
 572 ble and rigid multibody models. Technical report, SAE  
 573 Technical Paper, 2000.
- 574
- 575 Wang, S., Cao, Y., Zheng, X., and Zhang, T. Collision-  
 576 free trajectory planning for a 6-dof free-floating space  
 577 robot via hierarchical decoupling optimization. *IEEE*  
 578 *Robotics Autom. Lett.*, 7(2):4953–4960, 2022. doi: 10.  
 579 1109/LRA.2022.3152698. URL <https://doi.org/10.1109/LRA.2022.3152698>.
- 580
- 581 Wyk, K. V., Xie, M., Li, A., Rana, M. A., Babich, B., Peele,  
 582 B., Wan, Q., Akinola, I., Sundaralingam, B., Fox, D.,  
 583 Boots, B., and Ratliff, N. D. Geometric fabrics: Gen-  
 584 eralizing classical mechanics to capture the physics of  
 585 behavior. *IEEE Robotics Autom. Lett.*, 7(2):3202–3209,
- 586

550      2022. doi: 10.1109/LRA.2022.3143311. URL <https://doi.org/10.1109/LRA.2022.3143311>.

551  
552  
553      Zhang, X., Rui, X., Zhang, J., Gu, J., and Zhang, L. Statics  
554      analysis based on the reduced multibody system transfer  
555      matrix method. *Multibody System Dynamics*, pp. 1–25,  
556      2023.

557

558

559

560

561

562

563

564

565

566

567

568

569

570

571

572

573

574

575

576

577

578

579

580

581

582

583

584

585

586

587

588

589

590

591

592

593

594

595

596

597

598

599

600

601

602

603

604

605  
 606 A. Proofs  
 607

 608 A.1. The Proof of Theorem 3.1  
 609

 610 We first present the following lemma to assist in establishing the rationality of the theorem.  
 611

 612 **Lemma A.1.** *By partitioning each rigid body within system  $\mathcal{S}$  into an element set  $C = \{c_i\}_{i=0}^n$  following the procedure in  
 613 Section 3.2.1, the motion states of all constructed elements within  $C$ , as well as the external forces  $\{\vec{F}_i\}_{i=0}^n$  acting on these  
 614 elements, conform to Kane's equation.*  
 615

 616 *Proof.* According to the definition of generalized active force (Kane & Levinson, 1985),  $\mathcal{S}$ 's generalized active force  $K_\gamma$   
 617 according to the  $\gamma$ -th independent velocity can be expressed as follows:  
 618

619 
$$K_\gamma = \sum_{k=1}^m \vec{F}_k \cdot \vec{u}_{k,\gamma}, \quad (16)$$

 620 where  $m$  represents the number of point masses existing within  $\mathcal{S}$ ,  $\vec{F}_k$  denotes the external force vector acting on point mass  
 621  $k$ ,  $\vec{u}_{k,\gamma}$  denotes the partial speed corresponding to the  $\gamma$ -th independent speed. The  $i$ -th element can be regarded as a rigid  
 622 that consists of multiple point masses. Therefore, we have:  
 623

624 
$$\begin{aligned} K_\gamma &= \sum_{i=1}^n \sum_{l=1}^{\tilde{m}_i} \vec{F}'_{i,l} \cdot (\vec{u}_{i,\gamma} + \vec{\omega}_{i,\gamma} \times \vec{r}_{i,l}) = \sum_{i=1}^n \sum_{l=1}^{\tilde{m}_i} \vec{F}'_{i,l} \cdot \vec{u}_{i,\gamma} + \sum_{i=1}^n \sum_{l=1}^{\tilde{m}_i} \vec{F}'_{i,l} \cdot (\vec{\omega}_{i,\gamma} \times \vec{r}_{i,l}) \\ &= \sum_{i=1}^n \left( \sum_{l=1}^{\tilde{m}_i} \vec{F}'_{i,l} \right) \cdot \vec{u}_{i,\gamma} + \sum_{i=1}^n \left( \sum_{l=1}^{\tilde{m}_i} (\vec{r}_{i,l} \times \vec{F}'_{i,l}) \right) \cdot \vec{\omega}_{i,\gamma} = \sum_{i=1}^n \vec{R}_i \cdot \vec{u}_{i,\gamma} + \sum_{i=1}^n \vec{M}_i \cdot \vec{\omega}_{i,\gamma}, \end{aligned} \quad (17)$$

 625 where  $\vec{F}'_{i,l}$  denote the external force acting on  $l$ -th point mass among the  $\tilde{m}_i$  point masses that consist element  $c_i$ ,  $\vec{u}_{i,\gamma}$  denote  
 626 the partial velocity according to the  $\gamma$ -th independent velocity acting on the mass center of the element  $c_i$ ,  $\vec{\omega}_{i,\gamma}$  denote the  
 627 partial angular velocity of element  $c_i$  according to the  $\gamma$ -th independent velocity,  $\vec{r}_{i,l}$  denotes the radius vector of the  $l$ -th  
 628 point mass towards the mass center of the element  $c_i$ .  $\vec{R}_i$  and  $\vec{M}_i$  represent the forces and moments acting on individual  
 629 elements. Such forces and moments can be calculated based on the properties of external forces  $\{\vec{F}'_i\}_{i=0}^n$ . Consequently, we  
 630 can derive that  $K_\gamma$  can be expressed as aggregating the external forces and moments acting on these elements.  
 631

 632 Furthermore, we can represent  $\mathcal{S}$ 's generalized inertia force as follows:  
 633

634 
$$K_\gamma^* = \sum_{k=1}^m (-m_k \vec{a}_k) \cdot \vec{u}_{k,\gamma}, \quad (18)$$

 635 where  $m_k$  denote the mass of the  $k$ -th point mass,  $\vec{a}'_k$  represents its acceleration. Equation 18 can be reformulated as:  
 636

637 
$$\begin{aligned} K_\gamma^* &= \sum_{i=1}^n \sum_{l=1}^{\tilde{m}_i} (-m_{i,l} \vec{a}_{i,l}) \cdot (\vec{u}_{i,\gamma} + \vec{\omega}_{i,\gamma} \times \vec{r}_{i,l}) = \sum_{i=1}^n \sum_{l=1}^{\tilde{m}_i} (-m_{i,l} \vec{a}_{i,l}) \cdot \vec{u}_{i,\gamma} + \sum_{i=1}^n \sum_{l=1}^{\tilde{m}_i} (-\vec{r}_{i,l} \times m_{i,l} \vec{a}_{i,l}) \cdot \vec{\omega}_{i,\gamma} \\ &= -\sum_{i=1}^n M_i \vec{a}_i \cdot \vec{u}_{i,\gamma} - \sum_{i=1}^n \sum_{l=1}^{\tilde{m}_i} (\vec{r}_{i,l} \times m_{i,l} \vec{a}_{i,l}) \cdot \vec{\omega}_{i,\gamma} = -\sum_{i=1}^n M_i \vec{a}_i \cdot \vec{u}_{i,\gamma} - \sum_{i=1}^n \vec{L}_i^* \cdot \vec{\omega}_{i,\gamma}, \end{aligned} \quad (19)$$

 638 where  $m_{i,l}$  denotes the mass of  $l$ -th point mass that consists element  $c_i$ ,  $\vec{a}_{i,l}$  denotes the corresponding acceleration.  $M_i$  and  
 639  $\vec{a}_i$  denotes the mass and the acceleration of  $c_i$ ,  $\vec{L}_i^*$  can be determined based on the rotational state of  $c_i$ . Therefore, we can  
 640 derive that  $K_\gamma^*$  can be expressed as the aggregation of the motion states of elements within  $C$ . As demonstrated afore,  $K_\gamma$   
 641 can be expressed as aggregating the external forces and moments acting on  $C$ . As both  $K_\gamma^*$  and  $K_\gamma$  within Kane's equation  
 642 can be calculated with the given factors, we can conclude that the lemma holds.  $\square$   
 643

 644 Lemma A.1 demonstrates that the motion of element set  $C$  and external forces  $\{\vec{F}_i\}_{i=0}^n$  conform to the Kane's equation.  
 645 Next, we need to prove that equation 11 holds. According to the theorem, model  $g(\cdot)$  is strictly accurate. Furthermore, from  
 646  
 647

660 Equation 17, we have:

$$661 \quad \vec{R}_i = \sum_{l=1}^{\tilde{m}_i} \vec{F}_{i,l}, \quad (20)$$

662 which can be regarded as the resultant force acting on an element. Therefore, at time  $t$ , we have:

$$663 \quad \vec{R}_i^{(t)} = [F_{1,i}^{(t)}, F_{2,i}^{(t)}, \dots, F_{d,i}^{(t)}]^\top, \quad (21)$$

664 where  $F_{\gamma,i}^{(t)}$  denote the force value along coordinate  $q_\gamma$  at time  $t$ . Based on the statements in the theorem, due to the  
665 independence of generalized velocities from each other, we could take the generalized velocity corresponding to each  
666 generalized coordinate as independent velocity, the partial velocity  $\vec{u}'_{i,\gamma}$  can be represented as follows:  
667

$$668 \quad \vec{u}'_{i,\gamma} = \frac{\partial \vec{r}_i}{\partial q_\gamma} = \frac{\partial [q_{1,i}, \dots, q_{d,i}]^\top}{\partial q_\gamma}, \quad (22)$$

669 where  $q_\gamma$  is the generalized coordinate corresponding to the  $\gamma$ -th independent velocity.  $\vec{u}'_{i,\gamma}$  is a one-hot vector in such  
670 circumstances. Therefore, we have:

$$671 \quad \vec{R}_i^{(t)} \cdot \vec{u}'_{i,\gamma} = [F_{1,i}^{(t)}, F_{2,i}^{(t)}, \dots, F_{d,i}^{(t)}]^\top \cdot \frac{\partial [q_{1,i}, \dots, q_{d,i}]^\top}{\partial q_\gamma} = F_{\gamma,i}^{(t)} u_\gamma, \quad (23)$$

672 where  $u_\gamma$  represents the unit velocity value along the direction of the  $\gamma$ -th independent velocity. Therefore, at time  $t$ , we  
673 have:

$$674 \quad K_\gamma^{(t)} = \sum_{i=1}^n \vec{r}_i \cdot \vec{u}'_{i,\gamma} + \sum_{i=1}^n \vec{M}_i^{(t)} \cdot \vec{\omega}_{i,\gamma}^{(t)} = \sum_{i=1}^n F_{\gamma,i}^{(t)} + \sum_{i=1}^n \vec{M}_i^{(t)} \cdot \vec{\omega}_{i,\gamma}^{(t)} \\ 675 \quad = \sum_{i=1}^n F_{\gamma,i}^{(t)} + \sum_{i=1}^n \sum_{l=1}^{\tilde{m}_i} (\vec{r}_{i,l} \times \vec{F}_{i,l}^{(t)}) \cdot \vec{\omega}_{i,\gamma}^{(t)}. \quad (24)$$

676 Similarly, we can also derive the following result:

$$677 \quad K_\gamma^{*(t)} = - \sum_{i=1}^n M_i \vec{a}_i^{(t)} \cdot \vec{u}'_{i,\gamma} - \sum_{i=1}^n \vec{L}_i^{*(t)} \cdot \vec{\omega}_{i,\gamma}^{(t)} = - \sum_{i=1}^n M_i a_{\gamma,i}^{(t)} u_\gamma - \sum_{i=1}^n \sum_{l=1}^{\tilde{m}_i} (\vec{r}_{i,l} \times m_{i,l} \vec{a}_{i,l}^{(t)}) \cdot \vec{\omega}_{i,\gamma}^{(t)}, \quad (25)$$

678 where  $a_{\gamma,i}^{(t)}$  denotes the accelerate speed value along coordinate  $q_\gamma$  at time  $t$ . According to Kane's equation, we have:

$$679 \quad K_\gamma^{(t)} + K_\gamma^{*(t)} = 0. \quad (26)$$

680 With Equation 24, 25, and 26, we have:

$$681 \quad \sum_{i=1}^n F_{\gamma,i}^{(t)} u_\gamma + \sum_{i=1}^n \sum_{l=1}^{\tilde{m}_i} (\vec{r}_{i,l} \times \vec{F}_{i,l}^{(t)}) \cdot \vec{\omega}_{i,\gamma}^{(t)} - \sum_{i=1}^n M_i a_{\gamma,i}^{(t)} u_\gamma - \sum_{i=1}^n \sum_{l=1}^{\tilde{m}_i} (\vec{r}_{i,l} \times m_{i,l} \vec{a}_{i,l}^{(t)}) \cdot \vec{\omega}_{i,\gamma}^{(t)} = 0. \quad (27)$$

682 As the prediction model  $g(\cdot)$  has assumed to be able to accurately predict the state of  $\mathcal{S}$  in the Theorem 3.1, we have:

$$683 \quad \sum_{i=1}^n F_{\gamma,i}^{(t)} u_\gamma - \sum_i M_i \frac{1}{\delta} (g(G^{(t)})_{[v_\gamma, i]} - v_\gamma^{(t)}) u_\gamma + \Psi^{(t)} = 0. \quad (28)$$

684  $\Psi^{(t)}$  is regarded as a disturbance term, formally:

$$685 \quad \Psi^{(t)} = \sum_{i=1}^n \sum_{l=1}^{\tilde{m}_i} (\vec{r}_{i,l} \times \vec{F}_{i,l}^{(t)}) \cdot \vec{\omega}_{i,\gamma}^{(t)} - \sum_{i=1}^n \sum_{l=1}^{\tilde{m}_i} (\vec{r}_{i,l} \times m_{i,l} \vec{a}_{i,l}^{(t)}) \cdot \vec{\omega}_{i,\gamma}^{(t)} \quad (29)$$

686 From Equation 29, it can be easily conclude that:

$$687 \quad \Psi^{(t)} \propto \sum_{i=1}^n \sum_{l=1}^{\tilde{m}_i} |\vec{r}_{i,l}|. \quad (30)$$

688 So far, we have demonstrated Equation 11, along with the property of  $\Psi^{(t)}$ .

715 **A.2. The Proof of Theorem 3.2**

716 Based on Equation 29, we have:

717

$$\Psi^{(t)} = \sum_{i=1}^n \sum_{l=1}^{\tilde{m}_i} (\mathbf{r}_{i,l} \times \vec{\mathbf{F}}_{i,l}^{(t)}) \cdot \boldsymbol{\omega}_{i,\gamma}^{(t)} - \sum_{i=1}^n \sum_{l=1}^{\tilde{m}_i} (\mathbf{r}_{i,l} \times m_{i,l} \vec{\mathbf{a}}_{i,l}^{(t)}) \cdot \vec{\boldsymbol{\omega}}_{i,\gamma}^{(t)}. \quad (31)$$

718 Within Equation 31, the first term of  $\Psi$  can be derived as follows:

719

$$\begin{aligned} & \sum_{i=1}^n \sum_{l=1}^{\tilde{m}_i} (\mathbf{r}_{i,l} \times \vec{\mathbf{F}}_{i,l}^{(t)}) \cdot \vec{\boldsymbol{\omega}}_{i,\gamma}^{(t)} \leq \sum_{i=1}^n \sum_{l=1}^{\tilde{m}_i} (|\mathbf{r}_{i,l}| |\vec{\mathbf{F}}_{i,l}^{(t)}| \sin(\theta_{i,l}^{RF})) \omega_\gamma \\ & \leq \sum_{i=1}^n \sum_{l=1}^{\tilde{m}_i} (|\mathbf{r}_{i,\arg \max_l(|\mathbf{r}_{i,l}|)}| |\vec{\mathbf{F}}_{i,l}^{(t)}| \sin(\theta_{i,l}^{RF})) \omega_\gamma = \sum_{i=1}^n |\mathbf{r}_{i,\arg \max_l(|\mathbf{r}_{i,l}|)}| \sum_{l=1}^{\tilde{m}_i} (|\vec{\mathbf{F}}_{i,l}^{(t)}| \sin(\theta_{i,l}^{RF})) \omega_\gamma \\ & \leq \mathbf{r}_{\arg \max_i |\mathbf{r}_{i,\arg \max_l(|\mathbf{r}_{i,l}|)}|, \arg \max_l(|\mathbf{r}_{i,l}|)} \sum_{i=1}^n \sum_{l=1}^{\tilde{m}_i} (|\vec{\mathbf{F}}_{i,l}^{(t)}| \sin(\theta_{i,l}^{RF})) \omega_\gamma, \end{aligned} \quad (32)$$

720 where  $\mathbf{r}_{\arg \max_i |\mathbf{r}_{i,\arg \max_l(|\mathbf{r}_{i,l}|)}|, \arg \max_l(|\mathbf{r}_{i,l}|)}$  is the max radius  $\tau$  among all elements,  $\theta_{i,l}^{RF}$  denotes the angle between  $\mathbf{r}_{i,l}$  and  $\mathbf{F}_{i,l}^{(t)}$ . The above expression can be further derived as follows:

721

$$\tau \sum_{i=1}^n \sum_{l=1}^{\tilde{m}_i} (|\vec{\mathbf{F}}_{i,l}^{(t)}| \sin(\theta_{i,l}^{RF})) \omega_\gamma = \tau \sum_{i=1}^n (|\vec{\mathbf{R}}_i^{(t)}| \sin(\theta_{i,l}^{RF})) \leq \tau \sum_{i=1}^n (|\vec{\mathbf{R}}_i^{(t)}|) \omega_\gamma, \quad (33)$$

722 where  $\theta_i^{RF}$  denote the angle between  $\vec{\mathbf{R}}_i^{(t)}$  and  $\mathbf{r}_{i,l}$ . Using the same method, we can also determine the lower bound of  $\sum_{i=1}^n \sum_{l=1}^{\tilde{m}_i} (\mathbf{r}_{i,l} \times \vec{\mathbf{F}}_{i,l}^{(t)}) \cdot \vec{\boldsymbol{\omega}}_{i,\gamma}^{(t)}$ , and thus we can obtain:

723

$$-\tau \sum_{i=1}^n (|\vec{\mathbf{R}}_i^{(t)}|) \omega_\gamma \leq \sum_{i=1}^n \sum_{l=1}^{\tilde{m}_i} (\mathbf{r}_{i,l} \times \vec{\mathbf{F}}_{i,l}^{(t)}) \cdot \vec{\boldsymbol{\omega}}_{i,\gamma}^{(t)} \leq \tau \sum_{i=1}^n (|\vec{\mathbf{R}}_i^{(t)}|) \omega_\gamma. \quad (34)$$

724 Meanwhile, the second term of Equation 31 can be derived as follows:

725

$$\sum_{i=1}^n \sum_{l=1}^{\tilde{m}_i} (\mathbf{r}_{i,l} \times m_{i,l} \vec{\mathbf{a}}_{i,l}^{(t)}) \cdot \vec{\boldsymbol{\omega}}_{i,\gamma}^{(t)} \leq \sum_{i=1}^n I_i \dot{\omega}_i^{(t)}, \quad (35)$$

726 where  $I_i$  denotes the rotational inertia of element  $c_i$ . However, calculating the rotational inertia of irregular objects can be quite challenging, and this situation may arise with the elements we have divided. Therefore, we further derive Expressing 35 as follows:

727

$$\sum_{i=1}^n I_i \dot{\omega}_i^{(t)} \leq \sum_{i=1}^n \frac{2}{5} m_i \tau^2 \dot{\omega}_i^{(t)}, \quad (36)$$

728 where  $\frac{2}{5} m_i \tau^2$  be the rotational inertia of a sphere that contains element  $c_i$ . Based on Equation 35 and 36, we have:

729

$$-\sum_{i=1}^n \frac{2}{5} m_i \tau^2 \dot{\omega}_i^{(t)} \leq \sum_{i=1}^n \sum_{l=1}^{\tilde{m}_i} (\mathbf{r}_{i,l} \times m_{i,l} \vec{\mathbf{a}}_{i,l}^{(t)}) \cdot \vec{\boldsymbol{\omega}}_{i,\gamma}^{(t)} \leq \sum_{i=1}^n \frac{2}{5} m_i \tau^2 \dot{\omega}_i^{(t)}, \quad (37)$$

730 Therefore, we have:

731

$$-\tau \sum_{i=1}^n (|\vec{\mathbf{R}}_i^{(t)}|) \omega_\gamma - \sum_{i=1}^n \frac{2}{5} m_i \tau^2 \dot{\omega}_i^{(t)} \leq \Psi \leq \tau \sum_{i=1}^n (|\vec{\mathbf{R}}_i^{(t)}|) \omega_\gamma + \sum_{i=1}^n \frac{2}{5} m_i \tau^2 \dot{\omega}_i^{(t)}, \quad (38)$$

732 The theorem is proved.

733

770 **B. Experimental Details**

771 **B.1. Implementations**

773 **B.1.1. EDGE CONSTRUCTION FOR CONTACT-ORIENTED GRAPH.**

774  
 775 In this section, we introduce the construction of edges in our CONTACT-ORIENTED GRAPH. Many physical scenarios are  
 776 complex, involving multiple object interactions (including collision and friction) when the distance between them is minimal.  
 777 The interactions between particles belonging to different objects and those within the same object are often distinct. Hence,  
 778 we propose the concept of a contact graph. We continuously observe the distances between particles. When the distance  
 779 between particles falls below a certain threshold, and they do not belong to the same object, we determine that contact has  
 780 occurred. A new edge is established between these contacting particles to facilitate message passing, thus forming a new,  
 781 contact-related graph. This newly constructed graph is utilized to predict and generate the contact forces experienced by  
 782 the objects. Additionally, we leverage the pooling of particle-level information from the objects themselves to facilitate  
 783 interactions between different objects.

784 **B.1.2. THE SPECIFIC CALCULATION METHOD FOR EACH COMPONENT WITHIN  $\mathcal{L}_{cst}$ .**

785 For the prediction of  $F_{\gamma,i}^{(t)}$ , it is essential to first clarify the number of independent velocities in our system. The rigid body  
 786 system under study is a six-degree-of-freedom system, encompassing x, y, and z coordinates, as well as angular velocities  
 787 along the x, y, and z axes. Nevertheless, given the complexity of acquiring rotational inertia related to angular velocity and  
 788 the challenging nature of obtaining torque data, we limit our calculations to determining the magnitudes of force components  
 789 associated with independent velocities along the x, y, and z directions. These force components can be directly obtained  
 790 from the corresponding elements of the force vector  $f_i$  calculated in Section 3.2.2. In this way,  $|\vec{R}_i^{(t)}|$  can also be directly  
 791 obtained by calculating the magnitude of the force vector  $f_i$ .  $M_i$  can be obtained by calculating the volume of each element  
 792 and multiplying it by the density.

793 We utilize the most commonly known laws of physics for solving angles issues. The angular velocity is derived by  
 794 dividing the velocity difference of the current particle relative to the central particle by the distance between the current  
 795 and the central particle. The formula is thus expressed as  $\omega = \frac{\vec{v}}{r}$ , where  $\vec{v} = v_i(\text{vec, present}) - v_0(\text{vec, center}) =$   
 796  $\sqrt{(v_{i,x} - v_{0,x})^2 + (v_{i,y} - v_{0,y})^2 + (v_{i,z} - v_{0,z})^2}$ ,  $r = \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2 + (z_i - z_0)^2}$ . Similarly, for the ve-  
 797 locity component of each axis,  $\omega_{x,y,z} = \frac{v_{x,y,z}}{r_{x,y,z}}$ , where  $\vec{v}_x = v_{i,x} - v_{0,x}$ ,  $\vec{v}_y = v_{i,y} - v_{0,y}$ ,  $\vec{v}_z = v_{i,z} - v_{0,z}$ ;  
 798  $r_x = x_i - x_0$ ,  $r_y = y_i - y_0$ ,  $r_z = z_i - z_0$ . We will select the angular velocity component along the largest axis to  
 799 use as our angular velocity,  $\omega_{max} = \text{Max}(\omega_x, \omega_y, \omega_z)$ .

800 **B.1.3. STRUCTURAL CONSTRAINT MODULE**

801 In this study, we introduce a novel algorithm aimed at restoring deformed objects to their original, regular rigid body  
 802 shapes. This algorithm is based on the geometric transformation of three-dimensional point sets, achieving the restoration  
 803 of the object's shape by precisely calculating the optimal rotation and translation matrices. Specifically, the algorithm  
 804 selects elements from the current and reference states of the object for processing. It calculates the centroid of each set  
 805 and centralizes the data, shifting the coordinates of both sets to be centered around the origin. This step eliminates the  
 806 influence of positional deviations. Subsequently, it computes the covariance matrix of the two centralized data sets and  
 807 employs Singular Value Decomposition (SVD) to extract the rotation matrix. To ensure the correct rotational direction of the  
 808 object, we introduce a correction matrix to adjust the rotation matrix. This rotation matrix not only reflects the directional  
 809 differences between the two point sets but also ensures that the transformation adheres to the right-hand rule. Finally, by  
 810 calculating the translation vector, the adjusted reference point set is moved to a position as close as possible to the current  
 811 point set. Our experimental results indicate that this method can effectively restore deformed objects back to their original  
 812 shapes.

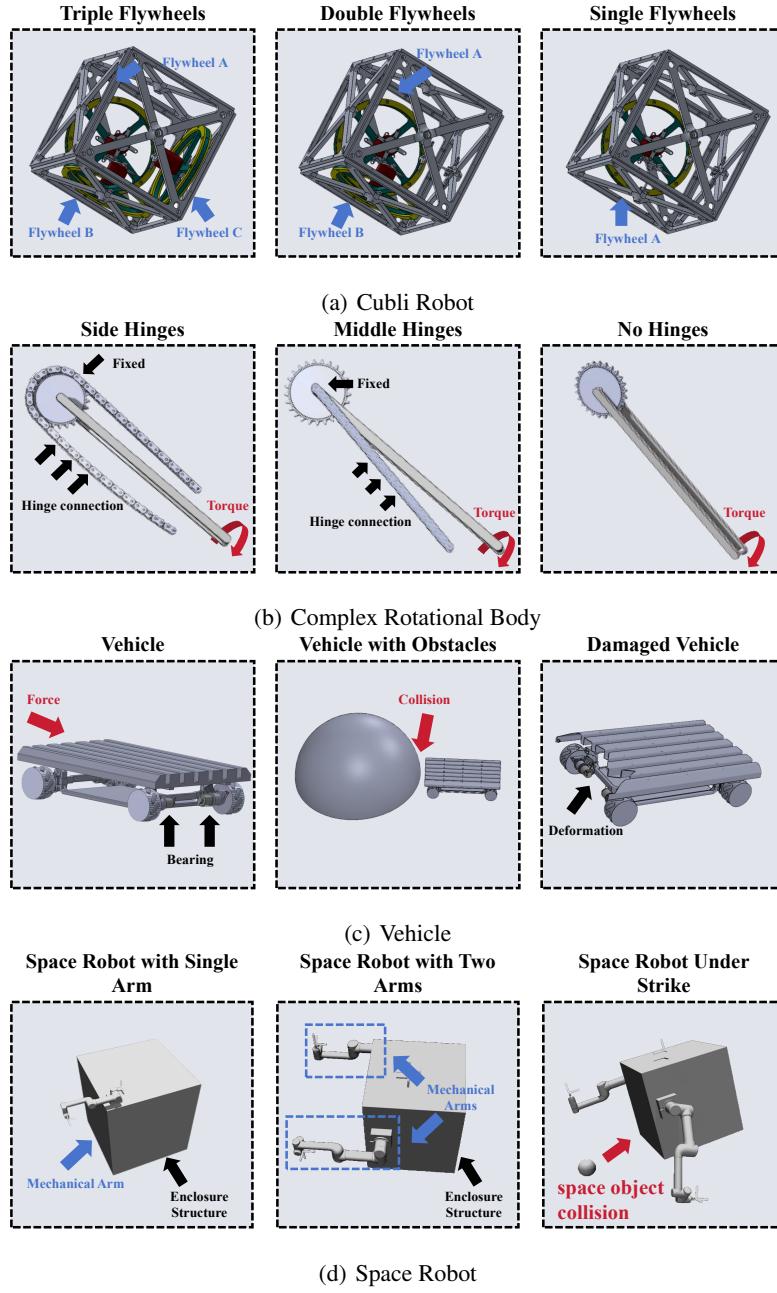
813 **B.2. Hyperparameters and Settings**

814 We adopt GCN as the backbone for all methods. Specifically, for LGNS and HGNS, we employ a hidden dimension of  
 815 200 for the node update function and 300 for message computation, with each hidden dimension comprising 3 layers and  
 816 utilizing Squareplus as the activation function. Due to its multi-stage hierarchical modeling, we use an Adam optimizer with  
 817 an initial learning rate of 0.001 and beta values of (0.9, 0.999), along with a patience-based Plateau scheduler having 3 cycles

825 and a decay factor of 0.8. However, we still make very minor modifications based on the aforementioned hyperparameters.  
 826 For SGNN, GNS, and MDGS, we employ SiLu or ReLU as the activation function and uniformly use a hidden dimension of  
 827 200, with early stopping after 10 epochs. Given the larger size of each system, we use a batch size of 1 for each dataset.  
 828 Furthermore, we inject noise during training to achieve better long-term prediction at test time, with the noise ratio set to  
 829 4e-4 in Cubli, and 0.06 times the standard deviation in the other dataset. The cut-off radius  $r$  for both datasets is set to 0.001.  
 830 On these two datasets, we only use the state information of the last frame  $t$  as input to predict the information of frame  $t + 1$ .  
 831 Experiments are conducted on a single NVIDIA RTX A6000 GPU.  
 832

### B.3. Physical Systems Used in The Experiments

833 Figure 11 shows the detailed structures of all physical systems utilized in the experiments.  
 834



878 *Figure 11.* The physical systems used in all twelve datasets.  
 879

**B.4. Further Results**

Tables 1, 2, and 3 present the digital representation of the conducted experiments, to more clearly display the results.

**Table 1.** Comparative experiment results. **Bold** indicates the method with optimal performance. Underline denotes the method with second-best performance. Some standard deviations are marked as 0.00 due to being too small to effectively represent.

Scenario		Cubli Robot			Complex Rotational Body		
Time	Methods	Single Flywheel	Double Flywheels	Triple Flywheels	No Hinge	Side Hinges	Center Hinge
$t=20$ (MSE $\times 10^3$ )	GNS	<u>4.47</u> $\pm 0.39$	<u>2.08</u> $\pm 0.17$	<u>3.95</u> $\pm 0.74$	<u>1.42</u> $\pm 0.63$	<u>4.09</u> $\pm 0.82$	<u>3.31</u> $\pm 0.96$
	LGNS	<u>14.05</u> $\pm 0.43$	<u>31.30</u> $\pm 6.80$	<u>8.52</u> $\pm 1.05$	<u>20.81</u> $\pm 3.14$	<u>3.33</u> $\pm 0.86$	<u>2.16</u> $\pm 1.15$
	SGNN	<u>3.22</u> $\pm 0.39$	<u>47.76</u> $\pm 3.55$	<u>36.27</u> $\pm 2.76$	<u>29.23</u> $\pm 5.29$	<u>23.86</u> $\pm 0.38$	<u>3.49</u> $\pm 0.47$
	HGNS	<u>2.08</u> $\pm 0.69$	<u>97.02</u> $\pm 7.14$	<u>16.67</u> $\pm 0.81$	<u>33.61</u> $\pm 6.33$	<u>2.55</u> $\pm 0.37$	<u>12.18</u> $\pm 2.05$
	MDGS (Ours)	<b>0.84</b> $\pm 0.28$	<b>1.76</b> $\pm 0.63$	<b>2.08</b> $\pm 0.24$	<b>0.84</b> $\pm 0.04$	<b>0.58</b> $\pm 0.03$	<b>0.92</b> $\pm 0.15$
$t=40$ (MSE $\times 10^3$ )	GNS	0.89 $\pm 0.17$	<u>0.32</u> $\pm 0.26$	<u>0.75</u> $\pm 0.05$	<u>0.27</u> $\pm 0.01$	<u>0.80</u> $\pm 0.09$	<u>0.65</u> $\pm 0.19$
	LGNS	2.80 $\pm 1.08$	<u>11.00</u> $\pm 1.73$	<u>3.12</u> $\pm 0.26$	<u>7.77</u> $\pm 1.79$	<u>0.65</u> $\pm 0.22$	<u>0.42</u> $\pm 0.22$
	SGNN	0.54 $\pm 0.32$	<u>17.24</u> $\pm 1.12$	<u>13.09</u> $\pm 0.38$	<u>10.65</u> $\pm 0.21$	<u>8.73</u> $\pm 1.06$	<u>1.31</u> $\pm 0.83$
	HGNS	<u>0.33</u> $\pm 0.13$	<u>36.51</u> $\pm 2.29$	<u>6.24</u> $\pm 0.27$	<u>12.84</u> $\pm 0.52$	<u>0.50</u> $\pm 0.09$	<u>2.37</u> $\pm 0.98$
	MDGS (Ours)	<b>0.17</b> $\pm 0.08$	<b>0.24</b> $\pm 0.06$	<b>0.31</b> $\pm 0.03$	<b>0.15</b> $\pm 0.04$	<b>0.11</b> $\pm 0.08$	<b>0.18</b> $\pm 0.04$
$t=110$ (MSE $\times 10^3$ )	GNS	0.25 $\pm 0.11$	<u>0.14</u> $\pm 0.02$	<u>0.22</u> $\pm 0.01$	<u>0.07</u> $\pm 0.01$	<u>0.22</u> $\pm 0.04$	<u>0.17</u> $\pm 0.03$
	LGNS	0.77 $\pm 0.23$	<u>7.61</u> $\pm 0.92$	<u>2.30</u> $\pm 1.13$	<u>5.71</u> $\pm 2.37$	<u>0.18</u> $\pm 0.03$	<u>0.11</u> $\pm 0.02$
	SGNN	0.12 $\pm 0.07$	<u>12.20</u> $\pm 0.37$	<u>8.73</u> $\pm 0.79$	<u>15.27</u> $\pm 1.21$	<u>9.54</u> $\pm 1.06$	<u>0.96</u> $\pm 0.21$
	HGNS	<u>0.09</u> $\pm 0.01$	<u>26.79</u> $\pm 0.44$	<u>4.61</u> $\pm 0.35$	<u>10.13</u> $\pm 0.78$	<u>0.14</u> $\pm 0.01$	<u>0.64</u> $\pm 0.39$
	MDGS (Ours)	<b>0.05</b> $\pm 0.01$	<b>0.06</b> $\pm 0.01$	<b>0.09</b> $\pm 0.01$	<b>0.04</b> $\pm 0.01$	<b>0.03</b> $\pm 0.00$	<b>0.05</b> $\pm 0.00$

**Table 2.** Generalization experiment results.  $\Delta$  indicates the disparity in model performance between the test set with changes and the original test set. **Bold** indicates the method with optimal performance. Underline denotes the method with second-best performance.

Time	Methods	More Intense Motion	$\Delta$	Less Intense Motion	$\Delta$	Shape Varied	$\Delta$	Density Varied	$\Delta$
$t=20$ (MSE $\times 10^3$ )	GNS	<u>115.61</u> $\pm 18.08$	<u>+113.53</u>	5.41 $\pm 0.27$	<u>+3.33</u>	<u>5.01</u> $\pm 0.62$	<u>+1.06</u>	<u>8.17</u> $\pm 3.57$	<u>+4.22</u>
	LGNS	188.71 $\pm 28.27$	<u>+157.41</u>	40.79 $\pm 13.93$	<u>+9.49</u>	21.77 $\pm 0.83$	<u>+13.25</u>	30.72 $\pm 9.43$	<u>+22.20</u>
	SGNN	614.13 $\pm 31.04$	566.36	97.02 $\pm 49.25$	<u>+49.26</u>	57.53 $\pm 1.02$	<u>+21.26</u>	61.92 $\pm 48.41$	<u>+25.65</u>
	HGNS	216.76 $\pm 47.45$	<u>+119.74</u>	197.58 $\pm 61.12$	<u>+100.56</u>	40.53 $\pm 1.02$	<u>+23.86</u>	120.01 $\pm 34.54$	<u>+103.34</u>
	MDGS (Ours)	<b>7.91</b> $\pm 2.85$	<b>+6.14</b>	<b>4.47</b> $\pm 0.38$	<b>+2.71</b>	<b>2.13</b> $\pm 0.26$	<b>+0.05</b>	<b>2.63</b> $\pm 0.58$	<b>+0.55</b>
$t=40$ (MSE $\times 10^3$ )	GNS	22.51 $\pm 3.15$	<u>+22.19</u>	1.00 $\pm 0.15$	<u>+0.68</u>	1.03 $\pm 0.13$	<u>+0.28</u>	1.55 $\pm 0.72$	<u>+0.79</u>
	LGNS	73.64 $\pm 2.51$	<u>+62.63</u>	15.62 $\pm 3.44$	<u>+4.61</u>	8.22 $\pm 1.21$	<u>+5.10</u>	11.05 $\pm 1.98$	<u>+7.93</u>
	SGNN	211.42 $\pm 19.05$	<u>+194.18</u>	36.51 $\pm 6.29$	<u>+19.26</u>	20.61 $\pm 3.53$	<u>+7.52</u>	22.02 $\pm 3.08$	<u>+8.93</u>
	HGNS	78.16 $\pm 16.71$	<u>+41.66</u>	70.53 $\pm 21.02$	<u>+33.54</u>	14.68 $\pm 2.55$	<u>+8.44</u>	23.37 $\pm 2.88$	<u>+17.13</u>
	MDGS (Ours)	<b>1.57</b> $\pm 0.56$	<b>+1.32</b>	<b>0.90</b> $\pm 0.19$	<b>+0.64</b>	<b>0.33</b> $\pm 0.03$	<b>+0.02</b>	<b>0.36</b> $\pm 0.15$	<b>+0.05</b>
$t=110$ (MSE $\times 10^2$ )	GNS	6.08 $\pm 0.05$	<u>+5.97</u>	0.36 $\pm 0.14$	<u>+0.22</u>	0.31 $\pm 0.03$	<u>+0.09</u>	0.48 $\pm 0.26$	<u>+0.26</u>
	LGNS	58.33 $\pm 4.91$	<u>+50.72</u>	11.60 $\pm 0.15$	<u>+3.97</u>	6.41 $\pm 1.32$	<u>+4.11</u>	7.25 $\pm 0.16$	<u>+4.95</u>
	SGNN	124.04 $\pm 11.04$	<u>+111.83</u>	26.80 $\pm 2.47$	<u>+1.45</u>	13.11 $\pm 2.37$	<u>+4.38</u>	13.67 $\pm 0.02$	<u>+4.94</u>
	HGNS	54.40 $\pm 7.15$	<u>+28.61</u>	50.86 $\pm 8.32$	<u>+24.07</u>	9.76 $\pm 0.98$	<u>+5.15</u>	6.37 $\pm 0.24$	<u>+1.76</u>
	MDGS (Ours)	<b>0.40</b> $\pm 0.15$	<b>+0.33</b>	<b>0.25</b> $\pm 0.02$	<b>+0.19</b>	<b>0.10</b> $\pm 0.01$	<b>+0.01</b>	<b>0.11</b> $\pm 0.02$	<b>+0.02</b>

**Table 3.** Ablation experiments compared with other baselines.

Methods	Cubli Robot		Complex Rotational Body	
	t=40	t=110	t=40	t=110
GNS	7.52 $\pm 0.49$	22.36 $\pm 1.21$	8.01 $\pm 0.69$	21.70 $\pm 3.71$
LGNS	31.2 $\pm 2.66$	230.09 $\pm 31.94$	6.52 $\pm 0.87$	17.59 $\pm 2.52$
SGNN	130.95 $\pm 3.77$	873.55 $\pm 39.42$	87.38 $\pm 7.26$	954.46 $\pm 60.82$
HGNS	62.44 $\pm 2.68$	461.5 $\pm 35.41$	4.98 $\pm 0.19$	13.53 $\pm 1.07$
MDGS-NC	110.84 $\pm 12.47$	763.36 $\pm 1.46$	8.40 $\pm 0.33$	60.15 $\pm 7.05$
MDGS-SC	15.24 $\pm 5.88$	43.47 $\pm 18.94$	3.32 $\pm 0.36$	20.85 $\pm 3.27$
MDGS	3.11 $\pm 0.32$	9.59 $\pm 1.27$	1.15 $\pm 0.31$	3.04 $\pm 0.64$