RIGID BODY DYNAMICS SIMULATION BASED ON GNNS WITH CONSTRAINTS

Anonymous authors

Paper under double-blind review

ABSTRACT

In recent years, the utilization of Graph Neural Network (GNN)-based methods for simulating complex physical systems has opened new avenues for the fields of computational 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 analysis 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. Codes and anonymous links to the datasets are available in the supplementary materials.

025 026 027

028

043

004

010 011

012

013

014

015

016

017

018

019

021

1 INTRODUCTION

029 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 031 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 033 object motion, stress, and interactions for optimization and performance enhancement. Traditional 034 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 035 understanding of the physical properties of the analyzed objects and the physical environment in 036 which they exist Thijssen (2007). Furthermore, such methods require a thorough grasp of relevant 037 physical principles and relationships. With the rise of deep learning, data-driven methods have excelled in various fields by removing the need for detailed prior knowledge of system intricacies, significantly simplifying complex problems Dong et al. (2021). Recently, GNNs have transformed 040 physics simulation, particularly in rigid body dynamics, by extending neural network applications to 041 physical analysis Sanchez-Gonzalez et al. (2020); Pfaff et al. (2021). GNNs analyze objects modeled 042 as graphs, demonstrating effective results in motion state analysis.

However, the current GNN-based methods concerning rigid body dynamic analysis are still limited to 044 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); 046 Bhattoo et al. (2022); Bishnoi et al. (2023); Allen et al. (2023b); Rubanova et al. (2021). This 047 limitation hinders the practical application of such methods in industrial settings. In industrial 048 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, 051 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 052 of objects for analysis. The difficulty in acquiring training data relevant to forces, coupled with the necessity for force relationship analysis, places these methods in a dilemma when applied to rigid

062

063

064

090

092

094

096

098



Figure 1: The comparison between the mechanical body scenarios we utilized and the general geometric body scenarios employed by other methods, results show that the performance of SGNN Han et al. (2022a) and GNS Sanchez-Gonzalez et al. (2020) exhibit significantly higher Mean Squared Error (MSE) in their predictive results for mechanical structures compared to geometric shapes.

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.

068 To address this challenge, we endeavor to draw inspiration from traditional multibody dynamics 069 analysis and incorporate relevant methods and constraints for calculating complex force relationships. It is noteworthy that while GNN-based dynamic analysis methods can assist in avoiding the need 071 for acquiring detailed prior knowledge about the internal complexities of systems, the output results 072 must adhere to the principles and rules of the physical world. Some researchers argue that a purely 073 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 074 traditional multibody dynamics analysis into GNN-based simulators, incorporating force analysis and 075 prior constraints, to address the challenges associated with rigid body dynamics analysis in complex 076 mechanical structures. 077

078 Building upon the aforementioned analysis, we propose a Multibody Dynamics Guided GNN 079 Simulator, dubbed MDGS, to conduct a more thorough analysis of rigid body dynamics scenarios involving complex mechanical structures. Initially, we introduce a graph modeling pattern for efficient analysis of forces in physical systems, tailored for multibody dynamics. Subsequently, we 081 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 083 of additional data beyond conventional GNN-based physical dynamics simulators. Then, the com-084 puted forces are integrated into the succeeding analyses to adapt to complex mechanical scenarios. 085 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 087 model's computed results. The major contributions are as follows.

- We introduce a novel method, referred to as MDGS, to incorporate force analysis into GNNbased simulators for the state prediction 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 multibody dynamics simulation tasks and create new datasets for validation, the results demonstrate the effectiveness of MDGS.
- 099 2 RELATED WORKS

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.

Dynamics Analysis. The framework of traditional mechanical analysis relies on either force-based 116 or energy-based approaches Wyk et al. (2022) for rigid body dynamics. Energy-based methods 117 compute unknown energy using equilibrium equations, which is cumbersome for large structures 118 Finzi et al. (2020). The force-based multibody dynamics model uses two recursive algorithms 119 centered on the Newton-Euler formulation for forward kinematics and inverse dynamics Gonçalves 120 et al. (2023). Some methods use ODEs to approximate optimal solutions for multibody conditions 121 Schubert et al. (2023); Nada & Bayoumi (2023); Zhang et al. (2023). Traditional mechanical analysis 122 requires precise physical parameters and domain knowledge. In contrast, GNN-based methods are 123 data-driven and do not need these inputs. We combine elements of rigid body dynamics with GNNs 124 for improved and simplified predictions.

125 126

127 128

129

115

3 METHODOLOGY

3.1 PRELIMINARY

130 GNN-based simulators. GNN-based simulators predict the dynamical states of physical systems. 131 Consider a physical system comprising M elements, collectively forming N objects. At time t, GNN-132 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. 133 The feature $z_i^{(t)}$ of node *i* in the set $\mathcal{V}^{(t)}$ includes position information $\vec{x}_i^{(t)}$, velocity information 134 135 $\vec{v}_i^{(t)}$, and the properties k_i of the object to which the element belongs, certain methods Allen et al. (2023a); Sanchez-Gonzalez et al. (2020) may retain only a subset of these features, such as velocity or 136 137 position. $\boldsymbol{z}_i^{(t)}$ is defined as the concatenation of the vectors mentioned: $\boldsymbol{z}_i^{(t)} = [\boldsymbol{\vec{x}}_i^{(t)} || \boldsymbol{\vec{v}}_i^{(t)} || \boldsymbol{k}_i]$, where '||' denotes vector concatenation along the first dimension. $\boldsymbol{Z}^{(t)} = \{\boldsymbol{z}_i^{(t)}\}_{i=1}^M$ denotes the set of all 138 139 $z_i^{(t)}$. The edge set $\mathcal{E}^{(t)}$ of graph $G^{(t)}$ characterizes the connectivity relationships between elements. 140 For instance, some approaches Han et al. (2022a); Bhattoo et al. (2022) judge that if the distance 141 between element i and j is less than a certain threshold, the edge (i, j) is connected. GNN-based 142 simulator utilizes the GNN model $g(\cdot)$ to predict the values of $Z^{(t+1)}$ based on $G^{(t)}$, such process 143 can be formalized as $Z^{(t+1)} = q(G^{(t)})$. 144

144 145

> **Kane's Equation.** Kane's equation Kane & Levinson (1985) is a mathematical formulation used 146 in the field of multibody dynamics to describe the motion of interconnected rigid bodies. For a 147 nonholonomic system \mathcal{S} , the number of independently variable motions or deformations is referred to 148 as degrees of freedom. For each degree of freedom, there exists a corresponding generalized velocity. 149 These velocity vectors associated with generalized coordinates are typically denoted by \dot{q} , where q 150 represents the generalized coordinates. Specifically, with respect to the generalized coordinate q_i and 151 its corresponding generalized velocity \dot{q}_i , if \dot{q}_i is independent of the other velocities in the system, 152 meaning that it cannot be expressed or derived from other velocities in the system's description, then 153 \dot{q}_i is referred to as an independent velocity. Furthermore, the partial velocity $u'_{ij} = \frac{\partial \dot{q}_i}{\partial \dot{q}_j}$ denotes the 154 partial derivative of the velocity with respect to coordinate q_i when coordinate q_i is varied, within the 155 context of a multi-coordinate system. With the aforementioned concepts, we could define the general 156 active force according to the γ -th independent velocity of S as follows:

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

158 159

157

where $\vec{F_i}$ denotes the force acting on the *i*-th mass point in the system S, and $\vec{u'_{\gamma,i}}$ represents its partial velocity with respect to the γ -th independent velocity, *n* is the number of mass points. Likewise, we

172

173

174 175

181

182

183

185

162 163

164



 $F^{(t)}$

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.

could define the generalized inertia forces according to the γ -th independent velocity as follows:

$$K_{\gamma}^{*} = \sum_{i=1}^{n} \left(-m_{i} \ddot{\vec{r}}_{i} \right) \cdot \vec{u}_{\gamma,i}^{\prime}, \tag{2}$$

Com

mse

 $\mathbf{z}^{*(t+1)}$

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

$$K_{\gamma}^{*} + K_{\gamma} = 0, \gamma = \{1, 2, ..., d\},$$
(3)

d is the number of independent velocities.

3.2 GRAPH CONSTRUCTION

Object-Orie Graph

Contact

ited Grap

187 Two commonly used object partitioning approaches 188 in the field of physical simulation are point clouds 189 Gross et al. (2002) and finite elements Pasciak (1995), 190 both of which have been widely applied in GNN-191 based simulators. Point clouds essentially involve 192 decomposing objects into indivisible elements using 193 point cloud construction techniques. On the other hand, the finite element method divides objects into 194 individual polyhedral elements, where the edges and 195 vertices of these elements constitute the graph for 196 GNN-based simulators. In our approach, we inte-197 grate the above two methods by employing the finite



(a) Original (b) Partitioned (c) Graph Con-Body Body struction

Figure 2: Illustration of graph construction.

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 3.2.1.

207 3.2.1 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: non-contact forces and contact forces.

214

206

For non-contact forces, such as gravity, we propose constructing an object-oriented graph $G^{(t)} = \{\mathcal{V}^{(t)}, \widetilde{\mathcal{E}}^{(t)}\}$, where $\mathcal{V}^{(t)}$ represents the centroids corresponding to distinct elements, with attribute

features comprising the input positional and velocity information of these elements at time t. These features consist set $Z^{(t)}$. Additionally, $\mathcal{V}^{(t)}$ includes property about the particular object each element is affiliated to, providing the information concerning the whole object. As for $\tilde{\mathcal{E}}^{(t)}$, as illustrated in Figure 4(a), 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 set of non-contact forces $\tilde{F}^{(t)}$. Formally, this prediction is expressed as:

223 224

238 239

240

241

242 243

244

245

246 247

248

252 253

254 255

256

257

$$\widetilde{F}^{(t)} = g^{\widetilde{F}}(\widetilde{G}^{(t)}),\tag{4}$$

where $g^{\vec{F}}(\cdot)$ denotes a GNN for non-contact force prediction, adopting the GCN described in Appendix B.2.

For contact forces, such as frictional and collision forces, we propose constructing a contact-oriented graph $\bar{G}^{(t)} = \{\mathcal{V}^{(t)}, \bar{\mathcal{E}}^{(t)}\}$. In $\bar{G}^{(t)}, \mathcal{V}^{(t)}$ remains identical to that of $\bar{G}^{(t)}$. The edge set $\bar{\mathcal{E}}^{(t)}$ comprises only those edges determined by the existence of contact between elements. As depicted in Figure 4(b), 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, where *r* is a threshold defined by a hyperparameter. In this manner, we derive a graph $\bar{G}^{(t)}$ specifically illustrating contact relationships.

Subsequently, based on the union of $\overline{G}^{(t)}$ and $\widetilde{G}^{(t)}$, we first obtain the features of each node to ensure that global information can be fully referenced when calculating contact-related forces. Formally, we have:

$$\bar{\mathbf{Z}}^{(t)} = g^{\bar{\mathbf{Z}}}(\bar{G}^{(t)} \cup \tilde{G}^{(t)}), \tag{5}$$

Where $g^{\bar{Z}}$ denotes a GNN for global information gathering. Afterwards, 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:

$$\bar{F}^{(t)} = g^{\bar{F}}(\bar{G}^{(t)}, \bar{Z}^{(t)}), \tag{6}$$

where $g^{\bar{F}}(\cdot)$ denotes a GNN for contact force prediction. In selecting the specific GNN architecture, we opt for the conventional GCN Kipf & Welling (2017). Detailed information regarding network depth and hidden layer dimensions can be found in Appendix B.2. The vector $\bar{f}_i^{(t)}$ will be set to the zero vector if *i* is not an endpoint of any edge in $\bar{\mathcal{E}}^{(t)}$.

The final force set is obtained by combining the contact and non-contact force sets:

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

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

3.3 MAKING PREDICTIONS

we predict $Z^{(t+1)}$ based on $F^{(t)}$ and $G^{(t)}$. The edge set of $G^{(t)}$ equals $\tilde{\mathcal{E}}^t$, as the contact forces are already in $F^{(t)}$, the contact edges $\bar{\mathcal{E}}^{(t)}$ are excluded. For the node set of $G^{(t)}$, we attach



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

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

the calculated force to the node attribute z. For the *i*-th node with attribute z_i , the new attribute z'_i is calculated as follows:

$$\boldsymbol{z}_{i}^{\prime} = [\boldsymbol{z}_{i} || \lambda F_{i}^{(t)}], \tag{8}$$

where λ is a hyperparameter for controlling the magnitude of the influence of $F^{(t)}$. We then calculated the prediction as follows:

269

265

266

$$Z^{(t+1)} = g(G^{(t)}), (9)$$

where $g(\cdot)$ denotes the GNN for state prediction, adopting the same GNN architecture elaborated above. 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**.

277 $Z^{(t+1)}$ will then be utilized for calculating an MSE loss against the ground truth state $Z^{*(t+1)}$ for 278 the backpropagation of the whole framework: 279

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

(10)

281 282

280

283

301

302

303

304

305

3.4 MODEL CONSTRAINTS BASED ON KANE'S EQUATION.

So far, we have incorporated force analysis to assist predictions within GNN-based simulators.
However, due to the absence of ground-truth labels for the forces, these computations lack accuracy.
As discussed in Section 3.1, Kane's equations provide physical constraints for rigid body dynamical
systems. We aim to employ these constraints to refine our model, thereby enhancing its analytical
fidelity to physical reality.

Theoretically, a model capable of accurately predicting rigid body systems adheres to the following principles:

Theorem 3.1. Consider a rigid body dynamics system denoted as S with d independent velocities. At the time t, the system's state is represented as $G^{(t)}$, constructed according to the detailed graph construction procedure outlined in Section 3.2 and 3.2.1. For any 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:

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

where γ denotes the index of independent velocity, $F_{\gamma,i}^{(t)}$ is the predicted force value of element *i* along γ -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 $g(\cdot)$, δ 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. u_{γ} represents the unit velocity value along the direction of the γ -th independent velocity.

306 The proof is provided in **Appendix** A.1. Since Equation 11 in Theorem 3.1 holds for any model 307 that accurately represents the target physical state, we can employ it to constrain the forces we have 308 computed, thereby ensuring that our solution adheres more closely to the physical laws. Furthermore, 309 Theorem 3.1 demonstrates that the constraints proposed in Equation 11 are valid along any inde-310 pendent velocity. Therefore, in practical implementation, we refine our model by adopting multiple 311 constraint equations along different independent velocities. The number of equations the model needs to satisfy can thus be determined based on the distinct independent velocities associated with different 312 systems. 313

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, then at time t, the following inequality holds:*

$$-\tau \sum_{i=1}^{n} (|\vec{R}_{i}^{(t)}|)\omega_{\gamma} - \sum_{i=1}^{n} \frac{2}{5}M_{i}\tau^{2}\dot{\omega}_{i}^{(t)} \leq \Psi^{(t)} \leq \tau \sum_{i=1}^{n} (|\vec{R}_{i}^{(t)}|)\omega_{\gamma} + \sum_{i=1}^{n} \frac{2}{5}M_{i}\tau^{2}\dot{\omega}_{i}^{(t)}, \quad (12)$$

319 320

318

where $\vec{R}_i^{(t)}$ denotes the interval force act on element *i*, τ is the max radius among all elements, ω_{γ} denotes unit angular velocity value along the direction of the γ -th independent velocity, $\dot{\omega}_i^{(t)}$ represents the max angular acceleration of element *i*. 324 Based on Theorem 3.1 and Theorem 3.2, we proposed the following loss function for the upper bound 325 of constrain: 326

327 328 $\mathcal{L}_{sup} = \operatorname{ReLU}\Big(\sum_{i} F_{\gamma,i}^{(t)} u_{\gamma} - \sum_{i} M_{i} \frac{1}{\delta} \big(g(G^{(t)})_{[v_{\gamma},i]}\big)$ $-v_{\gamma}^{(t)}\big)u_{\gamma} - \tau \sum_{i=1}^{n} (|\vec{R}_{i}^{(t)}|)\omega_{\gamma} - \sum_{i=1}^{n} \frac{2}{5}M_{i}\tau^{2}\dot{\omega}_{i}^{(t)}\Big),$ (13)

330 331 332

333

334

335

340

341

342 343 344

345

346 347 348

349 350

351

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:

$$\mathcal{L}_{inf} = \text{ReLU}\Big(-\sum_{i} F_{\gamma,i}^{(t)} u_{\gamma} + \sum_{i} M_{i} \frac{1}{\delta} \big(g(G^{(t)})_{[v_{\gamma},i]} - v_{\gamma}^{(t)}\big) u_{\gamma} - \tau \sum_{i=1}^{n} (|\vec{R}_{i}^{(t)}|) \omega_{\gamma} - \sum_{i=1}^{n} \frac{2}{5} M_{i} \tau^{2} \dot{\omega}_{i}^{(t)}\Big),$$
(14)

The overall loss for constrain is:

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

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

4 EXPERIMENTS

4.1 ENVIRONMENTAL SETUP

Dataset and Systems. To validate the pro-352 posed MDGS, we conducted method verifica-353 tion on a variety of complex mechanical sys-354 tems. Specifically, we utilized the Cubli robot 355 Gajamohan et al. (2012) (a metallic mechanical 356 cube capable of performing various maneuvers 357 using three flywheels), complex rotating bodies 358 (intricate mechanical structures featuring com-359 ponents such as bearings and hinges), vehicles 360 (mechanical cars operating under different con-361 ditions and environments), UR5-Husky robot 362 Wang et al. (2020) (vehicle-type robot equipped with mechanical arms), and 6-Dof Space robot 363 Wang et al. (2022) (robot equipped with mechan-364



plex internal and external forces and motion states.

(a) Each dataset incorpo- (b) Various adjustments rates a variety of com- are applied to the structures and actuators to create different datasets.

Figure 5: The dataset construction principles.

ical arms for executing specific tasks in space environments). We then modified the structures of these mechanical systems and introduced different environmental factors, resulting in the creation 366 of 15 distinct datasets. Figure 5 provides a simple explanation of the datasets we constructed, with 367 detailed descriptions available in Appendix B.4. 368

369

Baselines. We conduct a comparative analysis of our proposed MDGS against various baselines, 370 including GNS Sanchez-Gonzalez et al. (2020), SGNN Han et al. (2022a), LGNS Bhattoo et al. 371 (2022), and HGNS Bishnoi et al. (2023). Specifically, GNS employs a pure GNN for simulating 372 physical systems, while SGNN introduces the concept of subequivariant in the physical system into 373 the GNN-based physical dynamics simulator. Furthermore, LGNS and HGNS leverage Lagrangian 374 equations and Hamiltonian mechanics, respectively, to guide and constrain the GNN model, enhancing 375 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 376 with these baselines, we can effectively analyze the algorithmic performance of our approach in the 377 dynamic analysis of complex mechanical bodies.

Experimental Settings. Our designed rigid body structural constraint module significantly enhances the accuracy of GNN-based simulators for rigid body problems. For a fair comparison, we integrated this module into other baseline models, improving their performance similarly. The effects of this integration are detailed in Section 4.3. We evaluated model performance using MSE between predicted and actual values, conducting each experiment 10 times to ensure statistical significance, as described in Han et al. (2022a). Details on specific hyperparameters and model backbones are provided in Appendix B.2, and computational complexity is discussed in Appendix B.3.

Table 1: Comparative experiment results on complex rotational body datasets and different robot datasets. **Bold** indicates the method with optimal performance. <u>Underline</u> denotes the method with second-best performance. Some standard deviations are marked as 0.00 due to being too small to represent effectively.

	Senario	Cubli Robot			Complex Rotational Body			Space Robot		
Time	Methods	Single	Double	Triple	No	Side	Center	Single	Double	Under
Time		Flywheel	Flywheels	Flywheels	Hinge	Hinges	Hinge	Arm	Arms	Strike
1	GNS	0.89 ± 0.17	0.32 ± 0.26	0.75 ± 0.05	7.77±1.79	$0.80 {\pm} 0.09$	$0.65 {\pm} 0.19$	6.5 ± 0.00	11.2 ± 0.52	31.24 ± 10.31
10	LGNS	2.80 ± 1.08	11.00 ± 1.73	3.12 ± 0.26	0.27 ± 0.01	$0.65 {\pm} 0.22$	0.42 ± 0.22	7.78±1.79	11.1 ± 0.63	8.91 ± 0.98
₹¥×	SGNN	0.54 ± 0.32	17.24 ± 1.12	13.09 ± 0.38	10.65 ± 0.21	8.73 ± 1.06	1.31 ± 0.83	0.41 ± 0.16	$9.42 {\pm} 0.06$	12.61 ± 0.31
SE	HGNS	0.33 ± 0.13	36.51 ± 2.29	6.24 ± 0.27	12.84 ± 0.52	0.50 ± 0.09	2.37 ± 0.98	23.37 ± 0.88	8.12 ± 0.20	72.14 ± 1.77
S	MDGS (Ours)	0.17 ± 0.08	0.24 ± 0.06	$0.31 {\pm} 0.03$	0.15 ± 0.04	0.11 ± 0.08	$0.18{\pm}0.04$	0.39±0.00	$\textbf{3.22} \pm \textbf{0.30}$	$\textbf{4.01} \pm \textbf{0.36}$
3	GNS	0.25 ± 0.11	0.14 ± 0.02	0.22 ± 0.01	5.71±2.37	0.22 ± 0.04	0.17 ± 0.03	2.18 ± 0.21	4.80 ± 0.29	18.10 ± 2.29
10	LGNS	0.77 ± 0.23	7.61 ± 0.92	2.30 ± 1.13	0.07 ± 0.01	$0.18 {\pm} 0.03$	0.11 ± 0.02	5.71 ± 2.36	2.99 ± 0.18	3.84 ± 0.57
- <u>1</u>	SGNN	0.12 ± 0.07	12.20 ± 0.37	8.73 ± 0.79	15.27 ± 1.21	9.54 ± 1.06	0.96 ± 0.21	0.34 ± 0.11	2.63 ± 0.18	9.53 ± 2.09
t= ISE	HGNS	0.09 ± 0.01	26.79 ± 0.44	4.61 ± 0.35	10.13 ± 0.78	0.14 ± 0.01	$0.64 {\pm} 0.39$	6.37 ± 0.24	5.66 ± 0.23	58.02 ± 4.29
5	MDGS (Ours)	$0.05 {\pm} 0.01$	$0.06 {\pm} 0.01$	$0.09{\pm}0.01$	0.04±0.01	$0.03{\pm}0.00$	$0.05{\pm}0.00$	$0.22 {\pm} 0.00$	$2.40 {\pm} 0.33$	$2.51 {\pm} 0.93$



Figure 6: Part of the experimental results on performance evaluation.

4.2 PERFORMANCE EVALUATION

Precision. Building upon the aforementioned experimental setup, we conducted performance testing experiments for both MDGS and other baseline algorithms across a total of 15 datasets. Table 1 and 2 presents the experimental results. It is evident from the results that our proposed MDGS consistently outperforms the majority of comparisons. Figure 6(a) demonstrated rollout-MSE curves on part of the datasets. We provide all the curves in **Appendix B.5**.

Generalization. We compared the generalization performance of various methods using the complete Cubli robot with three flywheels and introduced system attribute variations. The methods trained

Table 2: Comparative experiment results on vehicle datasets and UR5-Husky robot dataset. **Bold** indicates the method with optimal performance. <u>Underline</u> denotes the method with the second-best performance.

anormanee.								
		Senario		Ve	UR5-Husky			
Tir	Time Methods		Original	Fixed Obstacle	Damaged	Movable Obstacle	Original	Object Interaction
	-	GNS	$1.52 {\pm} 0.58$	7.77 ±1.79	$5.81 {\pm} 0.33$	60.15 ± 0.69	28.81 ± 0.33	163.61 ± 16.22
-	2	LGNS	$0.39 {\pm} 0.07$	4.39 ± 0.07	0.41 ± 0.16	88.33 ± 0.89	10.60 ± 1.30	219.07 ± 35.00
-40	×	SGNN	0.45 ± 0.00	1.85 ± 0.00	0.39 ± 0.05	23.11 ± 0.31	82.01±5.58	154.44 ± 11.98
-	ISE	HGNS	0.21 ± 0.03	9.99 ± 0.03	$2.79 {\pm} 0.08$	108.25 ± 2.82	6.97 ± 0.13	172.02 ± 9.95
	3	MDGS (Ours)	$\overline{0.08 \pm 0.01}$	1.31 ± 0.03	$0.21{\pm}0.02$	$12.8 {\pm} 0.24$	6.18±0.26	$132.08 {\pm} 4.83$
	2)	GNS	0.37 ± 0.15	5.71 ± 2.36	$4.06 {\pm} 0.08$	304.99 ± 0.38	19.47 ± 6.67	44.44 ± 4.43
0	2	LGNS	0.07 ± 0.02	3.54 ± 0.12	$0.34 {\pm} 0.10$	114.52 ± 0.14	7.35 ± 1.97	59.40 ± 9.72
=10	X	SGNN	0.25 ± 0.00	1.29 ± 0.01	0.11 ± 0.01	24.14 ± 0.01	53.81 ± 1.35	41.82 ± 3.07
<u></u> Д	ISE	HGNS	0.08 ± 0.01	6.73 ± 0.01	0.77 ± 0.23	159.15 ± 3.89	5.10 ± 0.18	46.57 ± 2.69
	S	MDGS (Ours)	$0.02{\pm}0.00$	$0.96{\pm}0.01$	$0.08 {\pm} 0.00$	$10.13 {\pm} 1.10$	2.16 ± 0.03	35.68 ± 1.28

Table 3: Generalization experiment results on Cubli robot dataset with triple flywheels. Δ indicates the disparity in model performance between the test set with changes and the original test set. **Bold** indicates the method with optimal performance. <u>Underline</u> denotes the method with second-best performance.

Time	Methods	More Intense	Δ	Less Intense	Δ	Shape	Δ	Density	Δ
		Motion	_	Motion	_	Varied	-	Varied	_
	GNS	115.61 ± 18.08	+113.53	5.41 ± 0.27	+3.33	5.01 ± 0.62	+1.06	8.17±3.57	+4.22
10	LGNS	188.71 ± 28.27	+157.41	40.79 ± 13.93	+9.49	21.77±0.83	+13.25	30.72 ± 9.43	+22.20
× 12	SGNN	614.13 ± 31.04	566.36	97.02 ± 49.25	+49.26	57.53±1.02	+21.26	61.92 ± 48.41	+25.65
SE	HGNS	216.76 ± 47.45	+119.74	197.58 ± 61.12	+100.56	40.53 ± 1.02	+23.86	120.01 ± 34.54	+103.34
<u>N</u>	MDGS (Ours)	7.91 ± 2.85	+6.14	4.47 ± 0.38	+2.71	2.13±0.26	+0.05	$2.63 {\pm} 0.58$	+0.55
6	GNS	6.08 ± 0.05	+5.97	0.36 ± 0.14	+0.22	0.31±0.03	+0.09	0.48 ± 0.26	+0.26
10	LGNS	58.33 ± 4.91	+50.72	11.60 ± 0.15	+3.97	6.41±1.32	+4.11	7.25 ± 0.16	+4.95
<u> </u>	SGNN	124.04 ± 11.04	+111.83	26.80 ± 2.47	+1.45	13.11±2.37	+4.38	13.67 ± 0.02	+4.94
≞ ES	HGNS	54.40 ± 7.15	+28.61	50.86 ± 8.32	+24.07	9.76±0.98	+5.15	6.37 ± 0.24	+1.76
E E	MDGS (Ours)	$0.40 {\pm} 0.15$	+0.33	$0.25 {\pm} 0.02$	+0.19	$0.10 {\pm} 0.01$	+0.01	$0.11 {\pm} 0.02$	+0.02

on the original dataset were tested for their accuracy on a modified system, with results shown in Table 3. Our method outperformed others, and notably, except for MDGS, methods incorporating prior physical system knowledge generally fared poorly in these experiments. Figure 6(b) demonstrated rollout-MSE curves on part of the datasets, all the curves are provided in **Appendix B.5**.

446

447

4.3 IN-DEPTH STUDY

452 **Visualization.** To gain a comprehensive understanding of the performance of our proposed MDGS 453 method, we visualized the prediction results in detail. As demonstrated in 7(a), it is evident that 454 the MDGS method significantly enhances the accuracy of posture predictions for the Cubli robot 455 when compared to the SGNN method. This improvement suggests that MDGS offers a more 456 precise modeling approach for the internal flywheels and their intricate effects on rotational inertia, which are crucial for accurate dynamics simulation. Additionally, Figure 7(b) showcases MDGS's 457 strong capabilities in trajectory prediction, highlighting its ability to anticipate movement patterns 458 effectively. Meanwhile, Figure 7(c) further confirms the method's effectiveness when applied to 459 complex robots operating within dynamic spatial environments, emphasizing its versatility and 460 robustness. To enrich the understanding of our results, we also provide videos of the prediction 461 outcomes in the supplementary materials, allowing viewers to see the model's performance in 462 action and further illustrating the advantages of the MDGS method in real-time scenarios. These 463 visualizations collectively reinforce our findings and demonstrate the ultimate goal of achieving 464 enhanced predictive accuracy in robotic dynamics.

465

466 Structural Constraint Module Analysis. To rigorously assess the effectiveness of the proposed 467 structural constraint module, we conducted a series of validation experiments utilizing a Cubli 468 robot dataset. This dataset provided a suitable platform to evaluate our model's performance under 469 varying conditions. During these experiments, we compared the outcomes of our model with the structural constraint module against those obtained without it. As illustrated in Figure 7(e), the 470 results demonstrate a marked improvement in model performance when the module is integrated, 471 evidenced by significantly lower MSE values. This reduction in error highlights the module's efficacy 472 in enhancing the accuracy of rigid body dynamics analysis. The detailed results, presented in digital 473 form, can be found in the Appendix B.5, offering further insights into the quantitative improvements 474 observed in our experiments. These findings underscore the importance of structural constraints in 475 refining simulation fidelity and advancing the field of rigid body

476

477 **Ablation Study.** We conducted a comprehensive series of ablation experiments aimed at isolating 478 the effects of various constraints within Kane's equations. Specifically, we developed two baseline 479 models, which we refer to as MDGS-NC and MDGS-SC, to evaluate the contributions of these 480 constraints to the overall performance of our method. MDGS-NC is designed to remove all constraints 481 from Kane's equations entirely, allowing us to assess the performance impact of a constraint-free 482 environment. In contrast, MDGS-SC retains only an independent velocity constraint, providing a more limited framework for comparison. The experimental results, which are clearly illustrated in 483 Figure 7(d), demonstrate a significant distinction between the performance of the three models. It 484 is evident from our findings that the constraints inherent in Kane's equations play a critical role 485 in enhancing the performance of our method, thereby validating our approach and suggesting that

437

432



careful consideration of these constraints can lead to more accurate and robust simulations in complex mechanical systems.

5 CONCLUSION

In this paper, we propose a novel approach, MDGS, to incorporate force analysis into GNN-based
 rigid body system simulators, aiming to enhance the accuracy of such methods in complex mechanical
 scenarios. We substantiate the design of MDGS through theoretical foundations and proofs and
 validate its performance through a series of experiments.

540 REFERENCES

550

559

577

578

579 580

581

582

583

584 585

586

587

588

589

590

Kelsey R Allen, Tatiana Lopez Guevara, Yulia Rubanova, Kim Stachenfeld, Alvaro Sanchez-Gonzalez,
Peter Battaglia, and Tobias Pfaff. Graph network simulators can learn discontinuous, rigid contact
dynamics. In *Conference on Robot Learning*, pp. 1157–1167. PMLR, 2023a.

- Kelsey R. Allen, Yulia Rubanova, Tatiana Lopez-Guevara, William Whitney, Alvaro Sanchez-Gonzalez, Peter W. Battaglia, and Tobias Pfaff. Learning rigid dynamics with face interaction graph networks. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023b. URL https://openreview.net/pdf?id=J7Uh781A05p.
- Ravinder Bhattoo, Sayan Ranu, and N. M. Anoop Krishnan. Learning articulated rigid body dynamics
 with lagrangian graph neural network. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle
 Belgrave, K. Cho, and A. Oh (eds.), Advances in Neural Information Processing Systems 35:
 Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans,
 LA, USA, November 28 December 9, 2022, 2022.
- Suresh Bishnoi, Ravinder Bhattoo, Jayadeva Jayadeva, Sayan Ranu, and NM Anoop Krishnan.
 Learning the dynamics of physical systems with hamiltonian graph neural networks. In *ICLR 2023 Workshop on Physics for Machine Learning*, 2023.
- Raymond M Brach. *Mechanical impact dynamics: rigid body collisions*. Brach Engineering, LLC, 2007.
- Shi Dong, Ping Wang, and Khushnood Abbas. A survey on deep learning and its applications.
 Computer Science Review, 40:100379, 2021.
- 565 Roy Featherstone. *Rigid body dynamics algorithms*. Springer, 2014.

Marc Finzi, Ke Alexander Wang, and Andrew Gordon Wilson. Simplifying hamiltonian and lagrangian neural networks via explicit constraints. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual, 2020.

- Mohanarajah Gajamohan, Michael Merz, Igor Thommen, and Raffaello D'Andrea. The cubli: A cube that can jump up and balance. In 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2012, Vilamoura, Algarve, Portugal, October 7-12, 2012, pp. 3722–3727. IEEE, 2012. doi: 10.1109/IROS.2012.6385896. URL https://doi.org/10.1109/IROS.2012.6385896.
 - Fernando Gonçalves, Tiago Ribeiro, A Fernando Ribeiro, Gil Lopes, and Paulo Flores. Multibody model of the human-inspired robot charmie. *Multibody System Dynamics*, pp. 1–28, 2023.
 - Markus H. Gross, Hanspeter Pfister, Matthias Zwicker, Mark Pauly, Marc Stamminger, and Marc Alexa. Point based computer graphics. In Sabine Coquillart and Heinrich Müller (eds.), 23rd Annual Conference of the European Association for Computer Graphics, Eurographics 2002 Tutorials, Saarbrücken, Germany, September 2-6, 2002. Eurographics Association, 2002. doi: 10.2312/EGT.20021060. URL https://doi.org/10.2312/egt.20021060.
 - Jiaqi Han, Wenbing Huang, Hengbo Ma, Jiachen Li, Josh Tenenbaum, and Chuang Gan. Learning physical dynamics with subequivariant graph neural networks. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 December 9, 2022, 2022a.
- Xu Han, Han Gao, Tobias Pfaff, Jian-Xun Wang, and Liping Liu. Predicting physics in mesh reduced space with temporal attention. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022.* OpenReview.net, 2022b. URL
 https://openreview.net/forum?id=XctLdNfCmP.

594 595 596	Weihua Hu, Muhammed Shuaibi, Abhishek Das, Siddharth Goyal, Anuroop Sriram, Jure Leskovec, Devi Parikh, and C Lawrence Zitnick. Forcenet: A graph neural network for large-scale quantum calculations. In <i>ICLR 2021 SimDL Workshop</i> , volume 20, 2021.
597 598 599	Ronald L Huston and Chris Passerello. On multi-rigid-body system dynamics. <i>Computers & Structures</i> , 10(3):439–446, 1979.
600 601	Klaus Iglberger and Ulrich Rüde. Large-scale rigid body simulations. <i>Multibody system dynamics</i> , 25:81–95, 2011.
603	Thomas R Kane and David A Levinson. Dynamics, theory and applications. McGraw Hill, 1985.
604 605 606 607	Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings. OpenReview.net, 2017. URL https://openreview.net/forum?id=SJU4ayYgl.
608 609 610	Rubin H Landau, Manuel José Páez Mejía, and Cristian C Bordeianu. A survey of computational physics: introductory computational science. Princeton University Press, 2008.
611 612 613 614	Yunzhu Li, Jiajun Wu, Russ Tedrake, Joshua B. Tenenbaum, and Antonio Torralba. Learning particle dynamics for manipulating rigid bodies, deformable objects, and fluids. In <i>7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019.</i> OpenReview.net, 2019. URL https://openreview.net/forum?id=rJgbSn09Ym.
615 616 617 618 619 620	Jonas Linkerhägner, Niklas Freymuth, Paul Maria Scheikl, Franziska Mathis-Ullrich, and Ger- hard Neumann. Grounding graph network simulators using physical sensor observations. In <i>The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali,</i> <i>Rwanda, May 1-5, 2023.</i> OpenReview.net, 2023. URL https://openreview.net/pdf? id=js2sEd8VEY.
621 622	Brian Vincent Mirtich. Impulse-based dynamic simulation of rigid body systems. University of California, Berkeley, 1996.
623 624	Ayman Nada and Mona Bayoumi. Development of a constraint stabilization method of multibody systems based on fuzzy logic control. <i>Multibody System Dynamics</i> , pp. 1–33, 2023.
626 627 628	Joseph E. Pasciak. The mathematical theory of finite element methods (susanne c. brenner and l. ridgway scott). <i>SIAM Rev.</i> , 37(3):472–473, 1995. doi: 10.1137/1037111. URL https://doi.org/10.1137/1037111.
629 630 631 632	Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter W. Battaglia. Learning mesh- based simulation with graph networks. In 9th International Conference on Learning Repre- sentations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021. OpenReview.net, 2021. URL https://openreview.net/forum?id=roNqYL0_XP.
633 634 635 636	Yulia Rubanova, Alvaro Sanchez-Gonzalez, Tobias Pfaff, and Peter W. Battaglia. Constraint-based graph network simulator. <i>CoRR</i> , abs/2112.09161, 2021. URL https://arxiv.org/abs/2112.09161.
637 638 639 640 641	Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter W. Battaglia. Learning to simulate complex physics with graph networks. In <i>Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event</i> , volume 119 of <i>Proceedings of Machine Learning Research</i> , pp. 8459–8468. PMLR, 2020. URL http://proceedings.mlr.press/v119/sanchez-gonzalez20a.html.
642 643 644 645	Jörg Sauer and Elmar Schömer. A constraint-based approach to rigid body dynamics for virtual reality applications. In <i>Proceedings of the ACM symposium on Virtual reality software and technology</i> , pp. 153–162, 1998.
646 647	Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. <i>IEEE Trans. Neural Networks</i> , 20(1):61–80, 2009. doi: 10.1109/TNN.2008.2005605. URL https://doi.org/10.1109/TNN.2008.2005605.

- Matthias Schubert, Rodrigo T Sato Martín de Almagro, Karin Nachbagauer, Sina Ober-Blöbaum, and Sigrid Leyendecker. Discrete adjoint method for variational integration of constrained odes and its application to optimal control of geometrically exact beam dynamics. *Multibody System Dynamics*, pp. 1–28, 2023.
- MPT Silva, JAC Ambrosio, and MS Pereira. Biomechanical model with joint resistance for impact simulation. *Multibody System Dynamics*, 1(1):65–84, 1997.
- Barna Szabó and Ivo Babuška. Finite element analysis: Method, verification and validation. 2021.
- Jos Thijssen. *Computational physics*. Cambridge university press, 2007.

696 697

699 700

- Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2012, Vilamoura, Algarve, Portugal, October 7-12, 2012, pp. 5026–5033. IEEE, 2012. doi: 10.1109/IROS.2012.
 6386109. URL https://doi.org/10.1109/IROS.2012.6386109.
- Y Yi Tong. Vehicle dynamic simulations based on flexible and rigid multibody models. Technical report, SAE Technical Paper, 2000.
- Cong Wang, Qifeng Zhang, Qiyan Tian, Shuo Li, Xiaohui Wang, David Lane, Yvan Petillot, and Sen Wang. Learning mobile manipulation through deep reinforcement learning. *Sensors*, 20(3), 2020.
 ISSN 1424-8220. doi: 10.3390/s20030939. URL https://www.mdpi.com/1424-8220/20/3/939.
- Shengjie Wang, Yuxue Cao, Xiang Zheng, and Tao Zhang. Collision-free trajectory planning for a 6-dof free-floating space robot via hierarchical decoupling optimization. *IEEE Robotics Autom. Lett.*, 7(2):4953–4960, 2022. doi: 10.1109/LRA.2022.3152698. URL https://doi.org/10.1109/LRA.2022.3152698.
- Karl Van Wyk, Mandy Xie, Anqi Li, Muhammad Asif Rana, Buck Babich, Bryan Peele, Qian
 Wan, Iretiayo Akinola, Balakumar Sundaralingam, Dieter Fox, Byron Boots, and Nathan D.
 Ratliff. Geometric fabrics: Generalizing classical mechanics to capture the physics of behavior. *IEEE Robotics Autom. Lett.*, 7(2):3202–3209, 2022. doi: 10.1109/LRA.2022.3143311. URL
 https://doi.org/10.1109/LRA.2022.3143311.
- Kizhe Zhang, Xiaoting Rui, Jianshu Zhang, Junjie Gu, and Lina Zhang. Statics analysis based on the
 reduced multibody system transfer matrix method. *Multibody System Dynamics*, pp. 1–25, 2023.

13

PROOFS А

THE PROOF OF THEOREM 3.1 A.1

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

Lemma A.1. By partitioning each rigid body within system S into an element set $C = \{c_i\}_{i=0}^n$ following the procedure in Section 3.2, the motion states of all constructed elements within C, as well as the external forces $\{\vec{F}_i\}_{i=0}^n$ acting on these elements, conform to Kane's equation.

Proof. According to the definition of generalized active force Kane & Levinson (1985), S's generalized active force K_{γ} according to the γ -th independent velocity can be expressed as follows:

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

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

 $=\sum_{n}^{n}\sum_{i,l}^{\widetilde{m}_{i}}\vec{F}_{i,l}^{\prime}\cdot\vec{u}_{i,\gamma}+\sum_{n}^{n}\sum_{i,l}^{\widetilde{m}_{i}}\vec{F}_{i,l}^{\prime}\cdot\left(\vec{\omega}_{i,\gamma}\times\vec{r}_{i,l}\right)$

$$\sum_{i=1}^{n} l=1 \qquad i=1 l=1 = \sum_{i=1}^{n} \left(\sum_{l=1}^{\tilde{m}_{i}} \vec{F}_{i,l}' \right) \cdot \vec{u}_{i,\gamma} + \sum_{i=1}^{n} \left(\sum_{l=1}^{\tilde{m}_{i}} (\vec{r}_{i,l} \times \vec{F}_{i,l}') \right) \cdot \vec{\omega}_{i,\gamma}$$

$$\sum_{i=1}^{n} \vec{P}_{i,l} \vec{F}_{i,l} + \sum_{i=1}^{n} \vec{P}_{i,l} \vec{F}_{i,l}$$

$$(1)$$

 $K_{\gamma} = \sum_{i=1}^{n} \sum_{l=1}^{m_i} \vec{F}'_{i,l} \cdot (\vec{u}_{i,\gamma} + \vec{\omega}_{i,\gamma} \times \vec{r}_{i,l})$

$$=\sum_{i=1}^{n}\vec{R}_{i}\cdot\vec{u}_{i,\gamma}+\sum_{i=1}^{n}\vec{M}_{i}\cdot\vec{\omega}_{i,\gamma},$$
(17)

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

Furthermore, we can represent S's generalized inertia force as follows:

$$K_{\gamma}^{*} = \sum_{k=1}^{m} (-m_{k} \vec{a}_{k}) \cdot \vec{u}_{k,\gamma}, \qquad (18)$$

where m_k denote the mass of the k-th point mass, a'_k represents its acceleration. Equation 18 can be reformulated as:

759
760
$$K_{\gamma}^{*} = \sum_{i=1}^{n} \sum_{l=1}^{\widetilde{m}_{i}} (-m_{i,l} \vec{a}_{i,l}) \cdot (\vec{u}_{i,\gamma} + \vec{\omega}_{i,\gamma} \times \vec{r}_{i,l})$$

~

$$=\sum_{i=1}^{n}\sum_{l=1}^{m_{i}}(-m_{i,l}\vec{a}_{i,l})\cdot\vec{u}_{i,\gamma}+\sum_{i=1}^{n}\sum_{l=1}^{m_{i}}(-\vec{r}_{i,l}\times m_{i,l}\vec{a}_{i,l})\cdot\vec{\omega}_{i,\gamma}$$
$$=-\sum_{i=1}^{n}M_{i}\vec{a}_{i}\cdot\vec{u}_{i,\gamma}-\sum_{i=1}^{n}\sum_{l=1}^{\widetilde{m}_{i}}(\vec{r}_{i,l}\times m_{i,l}\vec{a}_{i,l})\cdot\vec{\omega}_{i,\gamma}$$

$$\begin{array}{l}
766 \\
767 \\
768 \\
769 \\
\end{array} = -\sum_{i=1}^{n} M_{i} \vec{a}_{i} \cdot \vec{u}_{i,\gamma} - \sum_{i=1}^{n} \vec{L}_{i}^{*} \cdot \vec{\omega}_{i,\gamma}, \tag{19}$$

$$= -\sum_{i=1}^{M} M_i \boldsymbol{u}_i \cdot \boldsymbol{u}_{i,\gamma} - \sum_{i=1}^{J} \boldsymbol{L}_i \cdot \boldsymbol{\omega}_{i,\gamma}, \tag{1}$$

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 \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 derive that K^*_{γ} can be expressed as the aggregation of the motion states of elements within C. As demonstrated afore, K_{γ} can be expressed as aggregating the external forces and moments acting on C. As both K_{γ}^* and K_{γ} within Kane's equation can be calculated with the given factors, we can conclude that the lemma holds.

Lemma A.1 demonstrates that the motion of element set C and external forces $\{F_i\}_{i=0}^n$ conform to the Kane's equation. Next, we need to prove that equation 11 holds. According to the theorem, model $g(\cdot)$ is strictly accurate. Furthermore, from Equation 17, we have:

$$\vec{R}_i = \sum_{l=1}^{\tilde{m}_i} \vec{F}_{i,l},\tag{20}$$

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

$$\vec{R}_{i}^{(t)} = [F_{1,i}^{(t)}, F_{2,i}^{(t)}, ..., F_{d,i}^{(t)}]^{\mathsf{T}},$$
(21)

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

$$\vec{u}_{i,\gamma}' = \frac{\partial \vec{r}_i}{\partial q_\gamma} = \frac{\partial [q_{1,i}, \dots, q_{d,i}]^{\mathsf{T}}}{\partial q_\gamma},\tag{22}$$

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

$$\vec{R}_{i}^{(t)} \cdot \vec{u}_{i,\gamma}^{\prime(t)} = [F_{1,i}^{(t)}, F_{2,i}^{(t)}, ..., F_{d,i}^{(t)}]^{\mathsf{T}} \cdot \frac{\partial [q_{1,i}, ..., q_{d,i}]^{\mathsf{T}}}{\partial q_{\gamma}} = F_{\gamma,i}^{(t)} u_{\gamma},$$
(23)

where u_{γ} represents the unit velocity value along the direction of the γ -th independent velocity. Therefore, at time t, we have:

$$K_{\gamma}^{(t)} = \sum_{i=1}^{n} \vec{r}_{i} \cdot \vec{u}_{i,\gamma}^{\prime(t)} + \sum_{i=1}^{n} \vec{M}_{i}^{(t)} \cdot \vec{\omega}_{i,\gamma}^{(t)}$$
$$= \sum_{i=1}^{n} F_{\gamma,i}^{(t)} + \sum_{i=1}^{n} \vec{M}_{i}^{(t)} \cdot \vec{\omega}_{i,\gamma}^{(t)}$$
$$= \sum_{i=1}^{n} F_{\gamma,i}^{(t)} + \sum_{i=1}^{n} \sum_{l=1}^{\tilde{m}_{i}} (\vec{r}_{i,l} \times \vec{F}_{i,l}^{(t)}) \cdot \vec{\omega}_{i,\gamma}^{(t)}.$$
(24)

Similarly, we can also derive the following result:

$$K_{\gamma}^{*(t)} = -\sum_{i=1}^{n} M_{i} \vec{a}_{i}^{(t)} \cdot \vec{u}_{i,\gamma}^{\prime(t)} - \sum_{i=1}^{n} \vec{L}_{i}^{*(t)} \cdot \vec{\omega}_{i,\gamma}^{(t)}$$

$$= -\sum_{i=1}^{n} M_{i} a_{\gamma,i}^{(t)} u_{\gamma} - \sum_{i=1}^{n} \sum_{l=1}^{\tilde{m}_{i}} (\vec{r}_{i,l} \times m_{i,l} \vec{a}_{i,l}^{(t)}) \cdot \vec{\omega}_{i,\gamma},$$
(25)

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

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

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

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

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

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

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

$$\Psi^{(t)} = \sum_{i=1}^{n} \sum_{l=1}^{\widetilde{m}_{i}} (\vec{r}_{i,l} \times F_{i,l}^{(t)}) \cdot \boldsymbol{\omega}_{i,\gamma}^{(t)} - \sum_{i=1}^{n} \sum_{l=1}^{\widetilde{m}_{i}} (\vec{r}_{i,l} \times m_{i,l} \boldsymbol{a}_{i,l}^{(t)}) \cdot \boldsymbol{\omega}_{i,\gamma}^{(t)}$$
(29)

From Equation 29, it can be easily conclude that:

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

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

A.2 THE PROOF OF THEOREM 3.2

Based on Equation 29, we have:

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

Within Equation 31, the first term of Ψ can be derived as follows:

where $r_{\arg \max_i | r_{i,\arg \max_l(|r_{i,l}|)}|,\arg \max_l(|r_{i,l}|)}$ is the max radius τ among all elements, $\theta_{i,l}^{rF}$ denotes the angle between $r_{i,l}$ and $F_{i,l}^{(t)}$. The above expression can be further derived as follows:

$$\tau \sum_{i=1}^{n} \sum_{l=1}^{m_i} (|\vec{F}_{i,l}^{(t)}| \sin(\theta_{i,l}^{rF})) \omega_{\gamma} = \tau \sum_{i=1}^{n} (|\vec{R}_i^{(t)}| \sin(\theta_{i,l}^{RF})) \le \tau \sum_{i=1}^{n} (|\vec{R}_i^{(t)}|) \omega_{\gamma},$$
(33)

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

$$-\tau \sum_{i=1}^{n} (|\vec{R}_{i}^{(t)}|) \omega_{\gamma} \leq \sum_{i=1}^{n} \sum_{l=1}^{\widetilde{m}_{i}} (r_{i,l} \times \vec{F}_{i,l}^{(t)}) \cdot \vec{\omega}_{i,\gamma}^{(t)} \leq \tau \sum_{i=1}^{n} (|\vec{R}_{i}^{(t)}|) \omega_{\gamma}.$$
(34)

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

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

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:

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

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:

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

Therefore, we have:

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

The theorem is proved.

B EXPERIMENTAL DETAILS

B.1 IMPLEMENTATIONS

B.1.1 EDGE CONSTRUCTION FOR CONTACT-ORIENTED GRAPH.

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



B.1.2 The Specific Calculation Method for Each Component Within \mathcal{L}_{cst} .

964

965

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 system under study is a six-degree-of-freedom system, encompassing x, y, and z coordinates, as well as angular velocities along the x, y, and z axes. Nevertheless, given the complexity of acquiring rotational inertia related to angular velocity and the challenging nature of obtaining torque data, we limit our calculations to determining the magnitudes of force components associated with independent velocities along the x, y, and z directions. These force components can be directly obtained from the corresponding elements of the force vector f_i calculated in Section



Figure 10: The rollout-MSE curves from the precision evaluation experiments conducted on Cubli robot datasets.



Figure 11: The rollout-MSE curves from the precision evaluation experiments conducted on complex rotational body datasets.

999 3.2.1. In this way, $|\vec{R}_i^{(t)}|$ can also be directly obtained by calculating the magnitude of the force vector f_i . M_i can be obtained by calculating the volume of each element and multiplying it by the density.

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

1011

1013

982

983

984 985

986

987

988

989

990

991

992

993

994 995

996

997 998

1012 B.1.3 STRUCTURAL CONSTRAINT MODULE

In this study, we introduce a novel algorithm aimed at restoring deformed objects to their original, reg-1014 ular rigid body shapes. This algorithm is based on the geometric transformation of three-dimensional 1015 point sets, achieving the restoration of the object's shape by precisely calculating the optimal rotation 1016 and translation matrices. Specifically, the algorithm selects elements from the current and reference 1017 states of the object for processing. It calculates the centroid of each set and centralizes the data, shift-1018 ing the coordinates of both sets to be centered around the origin. This step eliminates the influence of 1019 positional deviations. Subsequently, it computes the covariance matrix of the two centralized data 1020 sets and employs Singular Value Decomposition (SVD) to extract the rotation matrix. To ensure 1021 the correct rotational direction of the object, we introduce a correction matrix to adjust the rotation matrix. This rotation matrix not only reflects the directional differences between the two point sets but also ensures that the transformation adheres to the right-hand rule. Finally, by calculating the 1023 translation vector, the adjusted reference point set is moved to a position as close as possible to the 1024 current point set. Our experimental results indicate that this method can effectively restore deformed 1025 objects back to their original shapes.



Figure 12: The rollout-MSE curves from the precision evaluation experiments conducted on space robot datasets.



Figure 13: The rollout-MSE curves from the precision evaluation experiments conducted on UR5-HUSKY robot datasets.

1055

1049

1039

1040

1041

1043

1044

1054 B.2 HYPERPARAMETERS AND SETTINGS

 $g^F(\cdot), g^F(\cdot)$, and $g(\cdot)$ utilize identical GCN architectures, employing ReLU as the activation function, 1056 and consist of three layers with a hidden dimension of 200. Due to its multi-stage hierarchical 1057 modeling, we use an Adam optimizer with an initial learning rate of 0.001 and beta values of (0.9, 1058 0.999), along with a patience-based Plateau scheduler having 3 cycles and a decay factor of 0.8. 1059 Furthermore, we inject noise during training to achieve better long-term prediction at test time, with the noise ratio set to 4e-4 in Cubli, and 0.06 times the standard deviation in the other dataset. 1061 The cut-off radius r for both datasets is set to 0.001. On these two datasets, we only use the state 1062 information of the last frame t as input to predict the information of frame t + 1. Experiments are 1063 conducted on a single NVIDIA RTX A6000 GPU.

1064

B.3 COMPUTATIONAL COMPLEXITY

1067 Our algorithm exhibits a computational complexity akin to other GNN-based methods, specifically 1068 $\mathcal{O}(N \times d \times l)$. Here, N and d denote the number of nodes and degrees in the graph, respectively, 1069 while l represents the number of layers. The complexity is primarily influenced by the node count. 1070 Our method constructs graphs with fewer nodes compared to other GNN-based approaches because it only utilizes centroids and does not require meshes or point clouds to represent regular shapes. In 1071 contrast, when examining traditional numerical simulation methods such as finite element analysis, 1072 their complexity is expressed as $\mathcal{O}(N' \times d' \times k')$, where N' is the number of nodes, d' the average 1073 degree, and k' the number of functions computed. Like mesh-based methods, finite element analysis 1074 involves a larger N', resulting in greater computational complexity. 1075

1076

1078

1077 B.4 DATASET DETAILS

1079 Utilizing Mujoco Todorov et al. (2012) for physics-based modeling, we systematically vary motion patterns and mechanical components to generate 15 datasets comprising 2200 trajectories, with



Figure 14: The rollout-MSE curves from the generalization evaluation experiments conducted onCubli robot datasets.



Figure 15: The complete visualization results of experiments upon space robot dataset.

each trajectory consisting of 110 time steps, enabling a thorough evaluation of the robustness of our approach. Figure 8 shows the detailed structures of all physical systems utilized in the experiments.

B.5 FURTHER RESULTS

1121B.5.1COMPARATIVE RESULTS

To more clearly display the results, we provide all the rollout-MSE curves here within Figure 13, 10, 1124 11, 15, 13, and 14. From these results, it is evident that our method performs exceptionally well in most cases, significantly outperforming other comparative methods. This outcome robustly demonstrates the effectiveness of our proposed MDGS method and validates the necessity of incorporating physical knowledge into GNN-based simulators for accurate decision-making. By embedding physical knowledge into the GNN-based simulator, our approach can more accurately simulate and predict the behavior of complex systems, thereby enhancing overall performance.

We believe that the outstanding performance of MDGS across various scenarios is largely attributed
to its successful integration of a force analysis framework into the model, along with the effective
constraints based on principles of multibody dynamics. Specifically, the MDGS method considers
the interactions among different components of the physical system during the modeling process,
ensuring that the simulation results closely align with actual physical phenomena. Furthermore,

through the constraints imposed by multibody dynamics, our method is better equipped to handle
 multiple objects and their interactions within complex systems, thus maintaining high accuracy and
 robustness across diverse application scenarios.

1138 B.5.2 ABLATION STUDY RESULTS

Table 4: Ablation experiments compared with other baselines.

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

Furthermore, Table 4 provides a more detailed comparison of ablation experiments. Results within the table present a numerical comparison of the methods used in our ablation experiments with other methods. These results further validate the rationale and necessity of our methodological framework.

1153 B.5.3 VISUALIZATION RESULTS

Figure 15 provides a clearer illustration of the action flow prediction results for the space robot. From the figure, we can observe that due to the complexity of the research subject, other baselines exhibit larger errors and tend to make significant misjudgments regarding the overall position. We believe this is because other methods lack stronger physical constraints, leading to greater inaccuracies when handling predictions for complex mechanical systems.