Real-Time Simulation of Deformable Tactile Sensors and Objects in Robotic Grasping using Graph Neural Networks with Inductive Biases

Anonymous authors

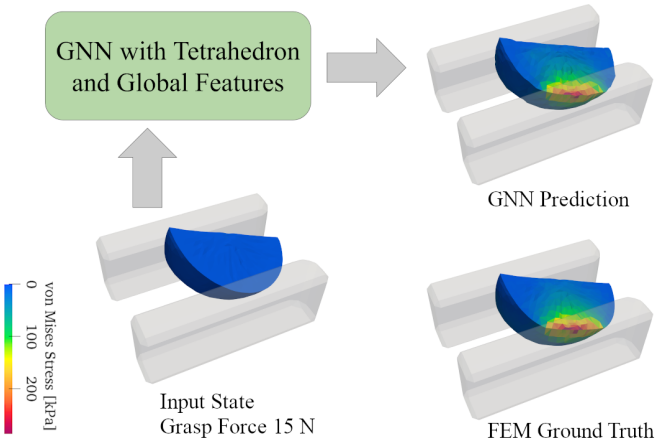

Fig. 1: Illustration of the pipeline for deformable object simulation using GNN prediction as an alternative to classical FEM simulation. This approach achieves speedups of up to 1000× for predicting both deformation and stress fields while maintaining accuracy comparable to ground truth measurements.

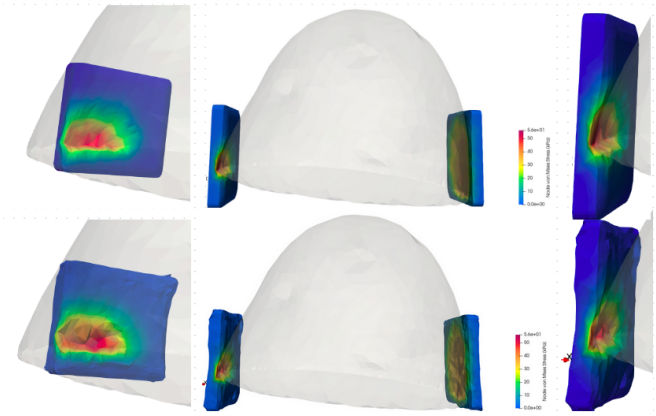

Fig. 2: Illustration of the ground truth stress and deformation of the tactile sensor across different grasping poses. The first row shows the ground truth, while the second row presents the prediction from the Graph Neural Network.

***Abstract*— Physical simulation of deformable bodies is crucial for robotic manipulation, particularly for applications involving deformable objects and deformable tactile sensors. While Finite Element Method (FEM) simulations provide high accuracy for modeling deformable objects and tactile sensors, they are prohibitively expensive for real-time applications, with simulation times often exceeding practical limits for robotic control and learning. This paper presents a novel Graph Neural Network (GNN) framework that accelerates the simulation of tactile sensors by factors of 10^2-10^4 compared to FEM, while maintaining high physical accuracy. Our approach addresses limitations in existing GNN-based physics learning through inductive biases. The key contributions include: (1) extending FEM simulation to deformable tactile sensors in grasping scenarios, (2) incorporating novel inductive biases through tetrahedral features and global graph features to improve information propagation and training stability, and (3) demonstrating the first successful application of GNN simulation for tactile sensors with generalization to unseen objects. Additionally, the inductive biases reduce prediction errors by up to 45% compared to baseline approaches. This work enables real-time soft tactile sensors of soft object simulation for robotic applications with stress prediction. All simulation and training code will be released.**

I. Introduction

Deformable objects and tactile sensors are omnipresent in human environments, and humans instinctively know how to grasp and handle them. However, robotic systems often lack this intuitive capability. Physical simulation plays a crucial role in the development and evaluation of robotic manipulation methods. The importance of simulation is even more pronounced when dealing with deformable objects or deformable visual tactile sensors in contact-rich scenarios, where real-world data collection is expensive, time-consuming, and often impractical for large-scale training. Despite years of research, simulating deformable tactile sensors remains highly challenging—both in terms of accurately capturing the underlying physical dynamics and meeting computational requirements for practical applications.

Related work on visual tactile sensor simulation can be broadly divided into two categories: *rigid-body* simulations and *soft-body* simulations. Rigid-body simulations prioritize execution speed, making them suitable for scenarios requiring large-scale data generation, such as reinforcement learning [7], [1]. However, they fundamentally cannot capture the deformation behavior and shear forces that characterize real tactile sensors. In contrast, soft-body approaches, particularly those based on FEM, offer greater realism by accurately modeling deformation under contact with external objects. However, they are significantly more computationally expensive and orders of magnitude slower than rigid-body simulations [11].

Recent work has attempted to reduce the computational cost using machine learning approaches. *DefGraspNets* by Huang et al. [5] demonstrated that GNNs can learn to predict deformation and stress fields in grasps on deformable objects,

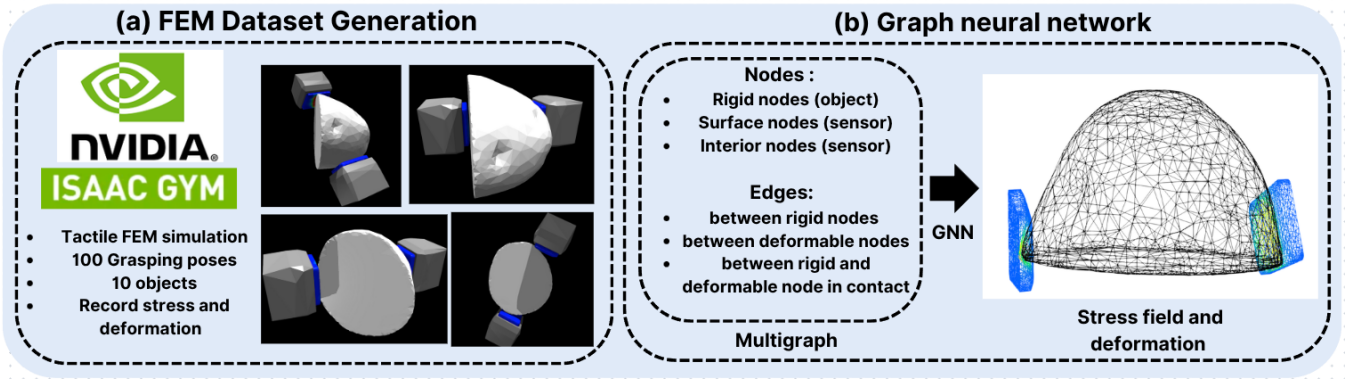

Fig. 3: Complete pipeline illustrating: (a) dataset creation using FEM simulation across 10 different objects and 100 grasping poses per object, showing the progressive gripper closure and force application, and (b) detailed construction of the Graph Neural Network architecture with node features (position, type, motion direction), edge features (geometric relationships, contact forces), and the inductive bias components.

achieving significant speedups over FEM simulations. Their approach builds on the GNN architecture from *MeshGraphNets* [9], representing gripper and object states as graph features where mesh vertices become nodes and mesh edges become graph edges. However, during our experimentation with the *DefGraspNets* approach, we improve the methods to address limitations: **Physically inconsistent stress prediction**: FEM computes stresses at tetrahedron elements, but the model predicts stress as values at nodes, requiring averaging that blurs stress peaks and reduces physical accuracy, **Limited information propagation**: Due to constrained message passing rounds, information cannot propagate through the entire graph, particularly affecting predictions for nodes far from contact regions.

This work addresses these limitations through a comprehensive GNN framework extended to tactile sensor simulation. Our main contributions are:

- **Extension of DefGraspSim[6] FEM simulation to deformable tactile sensors**: We develop a complete pipeline for simulating tactile sensors in grasping scenarios using FEM, providing high-quality training data that captures the complex interactions between rigid objects and deformable sensors.
- **Novel inductive biases for improved GNN performance**: We introduce two key architectural innovations: **Tetrahedral features** that enable direct prediction of stress at tetrahedron elements, aligning with FEM's physical formulation and preserving stress concentration accuracy. **Global graph features** that act as information shortcuts, enabling effective propagation through the entire graph and addressing the limited receptive field of standard message passing.
- **First comprehensive GNN simulation for tactile sensors**: We demonstrate the successful application of GNNs to tactile sensor simulation, with extensive validation showing effective generalization to unseen grasping poses across multiple objects and a speedup of 10^3–$10^4\times$ compared to FEM simulation

The remainder of this paper is organized as follows: Section II details our dataset generation pipeline using FEM simulation for tactile sensors. Section III presents our GNN methodology, including the baseline architecture and our proposed inductive biases. Section 4 describes our experimental setup and comprehensive results. Section 5 discusses limitations and future work, and Section 6 concludes.

II. Dataset Generation

A. FEM Simulation Framework

For dataset generation (illustrated in Fig. 3a), we employ a Finite Element Method (FEM) simulation within the Isaac Gym environment [8], which provides an effective trade-off between simulation speed and quality [11]. While prior work has proposed FEM-based simulations [11] for visual tactile sensors pressed against indenters, these approaches primarily focus on static object-sensor interactions. In contrast, we adopt and extend DefGraspSim [6], which enables loading of predefined grasping poses for various deformable objects and simulates dynamic parallel rigid gripper grasping scenarios involving both deformable objects and rigid gripper fingers. We further modify this simulation to extend it to deformable visual tactile sensors, specifically the GelSight Mini—one of the few visual tactile sensors commercially available and widely used in research, facilitating reproducibility and practical applications.

The simulation employs a parallel gripper equipped with two visual tactile sensors. Both the dataset generation pipeline and subsequent GNN framework are compatible with arbitrary tactile sensors, provided their meshes are available and convertible into tetrahedral `.tet` files, making the pipeline broadly applicable to various soft tactile sensors. This simulation framework supports two primary scenarios: soft objects grasped by rigid grippers (Fig. 1) and soft tactile sensors interacting with rigid objects (Fig. 3).

B. Simulation Procedure and Parameters

The simulation follows a carefully designed four-phase procedure. First, during object loading, the object mesh

(either soft or rigid) is loaded into the simulation environment at a predefined initial position. Second, in gripper positioning, the parallel gripper is positioned according to a specified grasping pose from the object surface. Third, contact establishment involves gripper closure until contact is detected between the gripper and object surface using collision detection algorithms. Finally, during force application, the grasping force is gradually increased according to a controlled profile until reaching a target force threshold N, with data recording occurring primarily in this phase.

The simulation terminates once the predefined force threshold N is achieved. During each run, we save 50 simulation frames corresponding to progressively increasing grasping forces, providing a continuous trajectory of the grasping process. For each frame, the simulator records multiple physical quantities: node deformation (3D displacement of each node in soft components), rigid body pose (6D position and orientation of gripper or rigid object), stress distribution (von Mises stress within each tetrahedral element of soft bodies), and contact forces at each interaction point between sensor and object.

For FEM parameters, we use pre-calibrated values of Poisson's ratio and Young's modulus for the GelSight Mini [11], specifically chosen to minimize sim-to-real gap in deformation and stress patterns during grasping. The dataset used for GNN training builds upon and extends the *DefGraspNet* data [5], containing a total of 63 diverse objects (examples shown in Fig. 4), each with 100 unique grasping poses, resulting in a comprehensive dataset of 6,300 distinct grasp simulations.

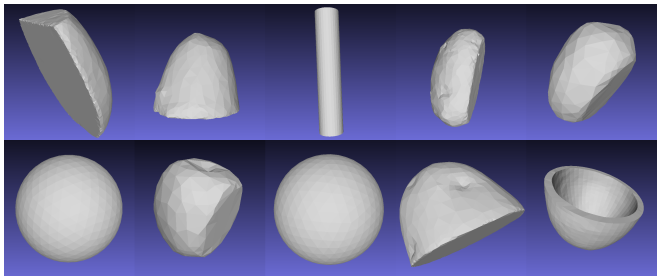

Fig. 4: Visualization of the 10 objects used in our dataset, selected to represent diverse geometries and mechanical properties encountered in real-world grasping scenarios.

C. Data Preprocessing and Graph Construction

The raw FEM simulation data undergoes comprehensive preprocessing to construct graph representations suitable for GNN training. This process involves mesh processing to convert tetrahedral meshes into graph representations where nodes correspond to mesh vertices and edges represent mesh connectivity. Feature extraction computes node features including position, type indicators, and motion direction, along with edge features such as relative displacements, distances, and contact forces. All features undergo appropriate normalization to ensure training stability, followed by graph augmentation that adds contact edges between nearby object and gripper nodes to model interaction forces.

The resulting dataset provides a rich foundation for training GNNs to predict the physical behavior of tactile sensors during grasping interactions. This preprocessing pipeline maintains consistent structure regardless of whether the soft component is the sensor on the gripper or the deformable object being grasped.

III. Graph Neural Network with Inductive Biases

A. Graph Neural Network Framework

We employ a Graph Neural Network as the central component for learning complex interactions in robotic grasping scenarios. Our framework builds upon the success of GNNs in physics-based simulation [9], [5]. The core insight is that GNNs naturally represent meshes as graphs, with message-passing mechanisms that propagate information analogously to physical force transmission in FEM simulations, making this kind of model suitable for physical simulation. In this part, we introduce innovations to address specific limitations in GNN predictions.

B. Graph Representation and Architecture

The graph construction forms the foundation of our approach, representing both deformable and rigid components in a unified structure. Each node corresponds to a vertex from either the deformable tactile sensor or rigid object, with two edge types capturing different physical relationships:

Mesh edges connect neighboring nodes within each structure to capture local geometric and mechanical dependencies, preserving mesh topology for material continuity modeling. **Contact edges** dynamically connect nearby object-sensor nodes to model interaction forces and contact physics.

The feature design carefully encodes physical information:

- **Node features** combine geometric state (3D position, displacement), node-type indicators, directional cues (finger motion vectors), and material properties
- **Edge features** include geometric relationships for mesh edges and force-related information for contact edges

C. Core Architectural Innovations

The efforts in implementing our methods started with the reimplementation of *DefGraspNets*. For the details of its node and edge feature constrution, we refer to [5], for node and edge updates in message passing to [9].

1) Tetrahedral Features and Stress Prediction: Our first proposed architectural extension informs the GNN about tetrahedrons in the object mesh. Similar to edges in [5], [9], a tetrahedron set is added to the input feature graph representation, consisting of four-tuples $\mathbb{T}_i$ that give corresponding node indices for each mesh tetrahedron i. Additionally, for each tetrahedron, an input feature vector $\mathbf{t}_i$ is given. First, the tetrahedral features are encoded into the common 128D latent space by an MLP f^t_{enc},

$$\tilde{\mathbf{t}}_i = f^t_{\text{enc}}(\mathbf{t}_i).$$

TABLE I: Achieved MAPE score (lower is better) on predicted deformation $\mathbf{u}$, stress $\boldsymbol{\sigma}$, and 9D rigid transformation $\mathbf{p}$ fields for each model variant, and each object trained and validated on.

Object	8polygon06			cylinder07			lemon01			potato2			sphere03			strawberry01		
MAPE in %	$\mathbf{u}$	$\boldsymbol{\sigma}$	$\mathbf{p}$	$\mathbf{u}$	$\boldsymbol{\sigma}$	$\mathbf{p}$	$\mathbf{u}$	$\boldsymbol{\sigma}$	$\mathbf{p}$	$\mathbf{u}$	$\boldsymbol{\sigma}$	$\mathbf{p}$	$\mathbf{u}$	$\boldsymbol{\sigma}$	$\mathbf{p}$	$\mathbf{u}$	$\boldsymbol{\sigma}$	$\mathbf{p}$
Baseline [5]	3.24	1.52	–	3.00	3.23	–	4.66	1.39	–	**3.05**	2.46	–	**2.37**	1.37	–	3.34	**1.12**	–
A: Tet. Features	3.86	1.91	–	2.93	2.37	–	4.51	1.92	–	3.34	1.75	–	2.71	**1.18**	–	4.05	4.57	–
B: Tet. + Global Feat.	**2.04**	**0.93**	**35.4**	**1.84**	**1.04**	**37.4**	**3.48**	**0.77**	**4.43**	3.12	**0.53**	**4.70**	3.24	1.21	**25.9**	**3.16**	3.40	**10.8**

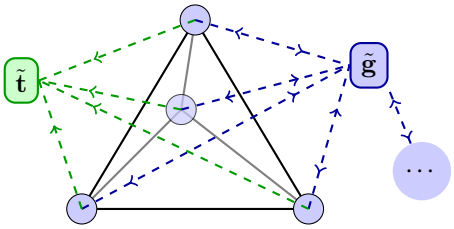

Fig. 5: The proposed inductive biases function during the message passing phase, illustrated for a single tetrahedron. In green: A tetrahedron feature receives messages from each of the tetrahedrons nodes. In blue: A global feature receives and sends messages from and to all nodes not only in this tetrahedron, but the entire graph.

For all tetrahedrons, the input $\mathbf{t}_i$ is scalar zero, hoping that all relevant information can be accumulated in message passing. This means the encoding step could be skipped and the latent feature initialized with zeros instead, but we keep it in our implementation to permit for future extensions.

In each message passing round, the update for the latent feature of each tetrahedron is computed as following: An update MLP f^t_{upd} is given the current latent tetrahedron feature $\tilde{\mathbf{t}}_i$, and the latent feature $\mathbf{v}_j$ of each node j that is part of the tetrahedron. Then, the updated feature is given by the MLP output with a residual connection:

$$\tilde{\mathbf{t}}'_i = f^t_{\text{upd}}\left(\tilde{\mathbf{t}}_i, \{\tilde{\mathbf{v}}_j\}_{j\in\mathbb{T}_i}\right) + \tilde{\mathbf{t}}_i$$

The information flow to the tetrahedron feature is shown in Fig. 5. This update rule lets information about the behavior of nodes flow into each tetrahedron feature. A decoder MLP f^t_{dec} finally decodes a prediction of the scalar von Mises stress $\hat{\sigma}_i$ within each tetrahedron,

$$[\hat{\sigma}_i] = f^t_{\text{dec}}(\tilde{\mathbf{t}}'_i).$$

The design choice that node information flows into tetrahedrons, but not back, is motivated by how FEM derives the stress field after computing node deformations.

2) Global Feature and Rigid Transformation: Addressing the problem of limited propagation through the graph, we modify the GNN architecture with a global feature [3]. It can be imagined as a node that is connected to all other graph nodes, and thus can accumulate and broadcast globally relevant information to all nodes. An input global feature vector $\mathbf{g}$ is transformed to its latent representation by an encoder MLP f^g_{enc}:

$$\tilde{\mathbf{g}} = f^g_{\text{enc}}(\mathbf{g}).$$

Again, we decide on a zero input vector $\mathbf{g} = 0$. In each message passing round, the update for the latent global feature is computed by an update MLP for the global node f^g_{upd}, given the mean of all node features as input, together with a residual connection:

$$\tilde{\mathbf{g}}' = f^g_{\text{upd}}\left(\frac{1}{\#\text{nodes}}\sum_{i=1}^{\#\text{nodes}}\tilde{\mathbf{v}}_i\right) + \tilde{\mathbf{g}}$$

To allow for information to flow back from the global to node features, the node update rule from [9] is modified to take the global feature as additional input,

$$\tilde{\mathbf{v}}'_i = f^v_{\text{upd}}\left(\tilde{\mathbf{v}}_i, \sum_j \tilde{\mathbf{e}}_{ij}, \tilde{\mathbf{g}}\right) + \tilde{\mathbf{v}}_i.$$

The flow of information to and from the global feature is illustrated in Fig. 5. Finally, enabling prediction of values relevant to the entire graph, after message passing the latent global feature is decoded into an output feature

$$[\hat{\mathbf{t}}_{\text{rigid}}, \hat{\mathbf{r}}_{\text{rigid}}] = \hat{\mathbf{g}} = f^g_{\text{dec}}(\tilde{\mathbf{g}}').$$

As demonstration, we use this decoded global output feature as a prediction for the best-fit rigid body transformation of the object during the grasp, given by a 3D translation vector $\hat{\mathbf{t}}_{\text{rigid}}$ and a 6D rotation representation $\hat{\mathbf{r}}_{\text{rigid}}$. We choose this continuous representation, as proposed by Zhou et al. [12], for improved training performance. The ground truth best-fit rigid body transformation is obtained with singular value decomposition of the covariance matrix between original and deformed object points [2], [10].

IV. Experiments and Results

A. Experimental Setup

Our experimental evaluation is structured around three core components designed to comprehensively validate the proposed approach:

1) **Inductive Bias Ablation**: A systematic comparison against baseline models to quantify the contribution of our proposed tetrahedral and global features for deformable object prediction (Sec. IV-B).
2) **Tactile Sensor Application**: An assessment of the model's practical utility by evaluating its performance

and generalization capabilities when applied to visual tactile sensor simulation across multiple scenarios (Sec. IV-C).

3) **Computational Performance**: An analysis of the inference speed and computational efficiency gains achieved by our method compared to traditional FEM simulation techniques(Sec. IV-D).

B. Inductive Biases for Deformable Object Prediction

This section presents a systematic evaluation of the proposed inductive biases for deformable object prediction. We assess the contribution of tetrahedral features and global information propagation through controlled comparisons against established baselines and progressive model variants.

1) Experimental Design: Following the evaluation methodology and scenario of DefGraspNet [5], we conduct experiments on diverse deformable objects to isolate the effects of each architectural component. We compare three model configurations:

- **Baseline**: Our reimplementation of DefGraspNets without inductive biases
- **Variant A**: Baseline enhanced with tetrahedral convolution features for direct stress prediction
- **Variant B**: Complete model integrating both tetrahedral features and global attention mechanisms

All models were trained using the Adam optimizer with learning rate scheduling and evaluated using Mean Absolute Percentage Error (MAPE) to ensure comparable scaling across different physical quantities. The evaluation encompasses deformation ($\mathbf{u}$), stress ($\boldsymbol{\sigma}$), and rigid transformation ($\mathbf{p}$) predictions across six geometrically diverse objects.

2) Quantitative Results: Table II presents detailed performance metrics across individual objects, while Table III summarizes aggregate performance. Several key patterns emerge from the experimental data:

Global Feature Dominance: Variant B demonstrates consistent superiority, achieving the lowest error rates on 5 of 6 objects for deformation prediction and 4 of 6 objects for stress prediction. The average improvement over baseline is 14.3% for deformation and 29.2% for stress prediction (Table III).

Synergistic Component Interaction: While tetrahedral features alone (Variant A) show limited benefits—sometimes even degrading performance—their combination with global features produces substantial improvements. This suggests tetrahedral features provide valuable physical constraints but require global information propagation to fully exploit this structural knowledge.

Object-Dependent Performance Gains: Complex geometries with irregular contact patterns (8polygon06, lemon01) benefit most significantly, showing error reductions up to 45%. In contrast, simple symmetric shapes (sphere03) show marginal improvements, indicating the inductive biases are particularly valuable for challenging deformation scenarios where local information proves insufficient.

Stress Concentration Preservation: The combined approach in Variant B achieves the most dramatic improvements in stress prediction, demonstrating an ability to preserve localized stress concentrations that are typically blurred in node-based approaches.

3) Discussion: The experimental results validate our hypothesis that carefully designed inductive biases significantly enhance deformable object prediction. The limited benefits of tetrahedral features alone (Variant A) suggest that local physical constraints, while valuable, are insufficient without effective global information propagation. The success of Variant B demonstrates that the combination of physical priors (tetrahedral features) and global reasoning capabilities creates a synergistic effect that addresses fundamental limitations in existing approaches.

The object-dependent performance patterns further reinforce that our inductive biases are most beneficial for complex deformation scenarios where local information proves inadequate. This finding has important implications for real-world applications involving diverse object geometries and complex contact interactions.

C. Tactile Sensor Prediction Results

This section evaluates our model's performance in predicting deformation and stress for visual tactile sensors across three scenarios of increasing complexity. For each scenario, we investigate two input configurations: one that provides the sensor's translation, and a more challenging, open-loop setting where only force is provided and the model must predict the resulting translation, deformation, and stress. These two scenarios represent realistic gripper control paradigms commonly used in real-world grasping applications.

1) Experimental Scenarios: We define three evaluation scenarios to systematically assess model capability:

- **Single-Object Specialization:** The model is trained and tested on different grasping poses of a single object to establish a performance baseline.
- **Multi-Object Generalization:** The model is trained on a set of objects and evaluated on unseen poses of those same objects, testing its ability to learn a generalizable physical representation.
- **Generalization to Novel Objects:** The model is evaluated on completely unseen object geometries, representing the most challenging zero-shot transfer scenario.

2) Single-Object Specialization: We first verify the model's core capability to predict deformation and stress for unseen grasping poses of a known object. Models were trained on 80% of the grasping poses for individual objects and evaluated on the remaining 20%.

As shown in Table IV, providing translation input drastically reduces prediction error, improving deformation accuracy by over 300% across all objects. This configuration not only yields the best performance but also aligns with real-world robotic systems where translation data is typically available.

3) Multi-Object Generalization: We next assess the model's ability to generalize across diverse objects by train-

TABLE II: MAPE scores (lower is better) for deformation ($\mathbf{u}$), stress ($\boldsymbol{\sigma}$), and rigid transformation ($\mathbf{p}$) predictions across six objects. Best results for each object are bolded.

	8polygon06			cylinder07			lemon01			potato2			sphere03			strawberry01		
MAPE (%)	$\mathbf{u}$	$\boldsymbol{\sigma}$	$\mathbf{p}$	$\mathbf{u}$	$\boldsymbol{\sigma}$	$\mathbf{p}$	$\mathbf{u}$	$\boldsymbol{\sigma}$	$\mathbf{p}$	$\mathbf{u}$	$\boldsymbol{\sigma}$	$\mathbf{p}$	$\mathbf{u}$	$\boldsymbol{\sigma}$	$\mathbf{p}$	$\mathbf{u}$	$\boldsymbol{\sigma}$	$\mathbf{p}$
Baseline	3.24	1.52	–	3.00	3.23	–	4.66	1.39	–	**3.05**	2.46	–	**2.37**	1.37	–	3.34	**1.12**	–
Variant A	3.86	1.91	–	2.93	2.37	–	4.51	1.92	–	3.34	1.75	–	2.71	**1.18**	–	4.05	4.57	–
Variant B	**2.04**	**0.93**	**35.4**	**1.84**	**1.04**	**37.4**	**3.48**	**0.77**	**4.43**	3.12	**0.53**	**4.70**	3.24	1.21	**25.9**	**3.16**	3.40	**10.8**

TABLE III: Average MAPE across all objects. Variant B achieves consistent improvements in both deformation and stress prediction.

Model Variant	$\mathbf{u}$	$\boldsymbol{\sigma}$	$\mathbf{p}$
Baseline	3.28	1.85	–
Variant A	3.57	2.28	–
Variant B	**2.81**	**1.31**	**19.77**

TABLE IV: Single-object specialization results. Providing translation input (✓) significantly improves accuracy, especially for deformation prediction.

Object	Trans.	Def. MAE	Stress MAE
Potato	✓	**6.57e-05**	**372.7**
	–	2.92e-04	382.8
Apple	✓	**7.20e-05**	**370.5**
	–	2.97e-04	427.9
Lemon	✓	**5.40e-05**	**212.1**
	–	2.38e-04	265.6

ing on 80% of grasping poses from multiple objects and testing on unseen poses from the same set.

TABLE V: Multi-object new grasping generalization performance on known objects. Models with translation input maintain high accuracy even as the object set scales.

Object Set	Trans.	Def. MAE	Stress MAE
Known-10	✓	**6.30e-05**	**360.3**
	–	2.69e-04	420.2
Known-63	✓	**7.90e-05**	**352.0**
	–	2.54e-04	365.0

Table V demonstrates that our approach maintains strong performance in multi-object settings. The model with translation input achieves 29.3% lower deformation error and 15.4% lower stress error compared to the configuration without translation. This confirms the model's capacity to learn generalizable physical representations rather than memorizing object-specific patterns.

4) Generalization to Novel Objects: The final and most significant evaluation assesses the GNN model's performance on completely unseen objects and grasping poses. For training, we used 80% of the grasping poses from 49 objects (comprising 14 polygons, 7 annuli, 6 cuboids, 3 cups, 8 cylinders, 1 eggplant, 3 potatoes, and 6 spheres). The model was evaluated on a separate set of 14 novel objects: 2 tomatoes, 1 hollow lemon, 4 ellipsoids, 3 different cups, 1 cucumber, and 3 apples.

As shown in Table VI, this scenario presents a greater challenge, with a noticeable performance drop compared to previous evaluations on known objects. For instance, the deformation and stress error increases by an order of magnitude. However, further analysis revealed that this drop is primarily attributable to two outlier ellipsoid objects. We hypothesize that their exceptionally high node density compared to other objects in the training set caused this performance degradation. After removing these two outliers, the performance metrics (Unknown-12) align more closely with those from previous scenarios, albeit with slightly higher error rates.

TABLE VI: Generalization performance on novel objects with (✓) and without (–) translation input. Unknown-12 excludes two outlier ellipsoid objects.

Object Set	Trans.	Def. MAE	Stress MAE
Unknown-14	✓	**1.97e-04**	**1.51e3**
	–	3.25e-02	7.78e4
Unknown-12	✓	**8.54e-05**	**6.19e2**
	–	2.81e-04	6.43e2

TABLE VII: Computational performance comparison, demonstrating the massive inference speedup achieved by our GNN approach while maintaining high predictive accuracy.

Method	Inference Time	Speedup vs. FEM	Hardware
FEM Simulation	100–200 s	1×	GPU
GNN (Ours)	**5–50 ms**	**10^3–10^4**×	GPU

The results powerfully demonstrate the promising generalization capacity of our GNN framework, as shown in the qualitative results in Figure 6. Its ability to accurately predict physical behavior for the vast majority of novel objects confirms the strength of our approach. The performance variance appears to be a data coverage issue rather than a fundamental architectural limitation. We are confident that scaling the dataset to encompass a broader spectrum of shapes and node densities will further enhance the model's zero-shot generalization capabilities.

D. Computational Performance and Real-Time Capability

A critical advantage of our GNN-based approach is its computational efficiency, which enables real-time applications. Table VII summarizes the performance comparison against a conventional FEM solver.

Our method achieves a speedup of three to four orders of magnitude (10^3–$10^4\times$) over the FEM baseline, reducing inference times from minutes to milliseconds. This performance enables real-time execution at 20–200 Hz, making our framework particularly suitable for applications requiring

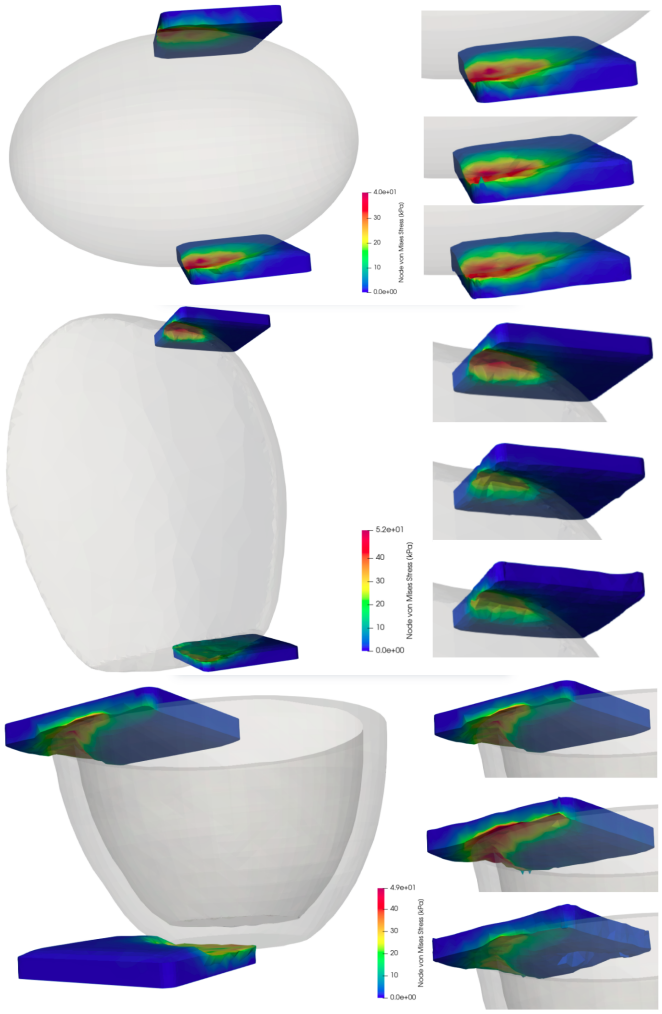

Fig. 6: GNN deformation and stress prediction on three unseen objects. Left: global GT scene overview. Right: zoomed-in view comparing (top to bottom) FEM ground truth, GNN prediction from translation input, and GNN prediction from force input.

rapid, iterative simulation, such as reinforcement learning, real-time grasp planning, and closed-loop robotic control.

V. Resulting Research Directions

While our method demonstrates promising results, several limitations present opportunities for future work.

Generalization and Data Diversity. Although our model generalizes reasonably to novel objects, its performance is bounded by the diversity of the training dataset. A key direction for improvement is to incorporate large-scale grasping datasets, such as the Acronym Dataset [4], which provides over 8,000 object models and corresponding grasp annotations. This would significantly enhance the model's ability to reason about a wider variety of object shapes and grasping strategies.

Quasi-Static and Dynamic Modeling. Our current framework operates under a quasi-static assumption, neglecting dynamic effects and gravitational influences. Extending the model to account for object inertia, swing motions, and impact forces is a critical next step for handling real-world dynamic manipulation tasks, such as catching or snatching objects.

Differentiable Grasp Pose Refinement. The differentiable nature of our Graph Neural Network (GNN) model opens a direct pathway for closed-loop grasp optimization. Future work will focus on leveraging this to backpropagate tactile-derived signals—such as predicted slip or undesirable stress concentrations—directly to the gripper's control parameters. By formulating a loss function that penalizes high stress or instability, the system can iteratively refine its pose and force to achieve stable grasps while simultaneously minimizing internal sensor strain, thereby enhancing hardware longevity.

Generalization to Diverse and Novel Tactile Sensors. A significant advantage of our physics-based GNN approach is its capacity for sensor-agnostic generalization. The model learns an implicit representation of continuum mechanics, which is not tied to a specific sensor morphology. Consequently, the framework can be extended to grippers with multiple, heterogeneous tactile sensors, or adapted to entirely new sensor designs, without the need for retraining from scratch. This would be achieved by simply instantiating new nodes within the graph that represent the additional or novel sensors, using their respective geometric and material properties as features. This capability would dramatically increase the practical applicability and scalability of the method.

Bridging the Sim-to-Real Gap. Although we use physically calibrated parameters, experimental validation with physical tactile sensors is essential to fully assess and bridge the transfer from simulation to physical systems. The next critical step involves a rigorous quantification of the sim-to-real gap through comparative analysis against data from physical sensors, such as the BioTac or DIGIT, under identical contact conditions. The insights from this real-world data will be instrumental in refining the simulation's material and contact models. Furthermore, integrating our deformation output with tactile image rendering pipelines will enable complete simulation-to-visualization workflows, combining physical deformation prediction with photorealistic tactile sensor output generation for end-to-end tactile simulation.

VI. Conclusion

This work presents a novel Graph Neural Network framework that enables real-time, high-fidelity simulation of deformable tactile sensors and objects in robotic grasping. By integrating critical inductive biases—tetrahedral features for physically consistent stress prediction and global graph features for effective long-range information propagation—we overcome fundamental limitations of prior GNN-based physics simulators. Our method achieves a transformative speedup of 10^3–$10^4\times$ compared to conventional FEM simulation while maintaining high accuracy, as validated through extensive experiments on diverse object geometries and grasping poses.

The key outcomes of this research are threefold. First, we establish the first comprehensive pipeline for simulating deformable visual tactile sensors within dynamic grasping scenarios, bridging a significant gap in robotic simulation capabilities. Second, we demonstrate that our inductive biases not only enhance predictive accuracy by up to 45% but also enable successful generalization to novel objects and sensor configurations. Finally, the real-time performance of our framework unlocks new possibilities for applications previously hindered by computational bottlenecks, including real-time grasp planning, closed-loop tactile control, and large-scale data generation for reinforcement learning.

By providing a fast, accurate, and generalizable simulation tool, this work lays a foundation for the next generation of tactile-aware robotic manipulation systems. The released code and models will facilitate further research in sim-to-real transfer and data-driven manipulation, accelerating progress toward robots that can reliably interact with complex deformable environments.

REFERENCES

[1] I. Akinola, J. Xu, J. Carius, D. Fox, and Y. Narang, "Tacsl: A library for visuotactile sensor simulation and learning," *IEEE Transactions on Robotics*, 2025.

[2] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3-D point sets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-9, no. 5, pp. 698–700, 1987.

[3] P. W. Battaglia, J. B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner, *et al.*, "Relational inductive biases, deep learning, and graph networks," *arXiv preprint arXiv:1806.01261*, 2018.

[4] C. Eppner, A. Mousavian, and D. Fox, "Acronym: A large-scale grasp dataset based on simulation," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 6222–6227.

[5] I. Huang, Y. Narang, R. Bajcsy, F. Ramos, T. Hermans, and D. Fox, "DefGraspNets: Grasp planning on 3D fields with graph neural nets," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 5894–5901.

[6] I. Huang, Y. Narang, C. Eppner, B. Sundaralingam, M. Macklin, R. Bajcsy, T. Hermans, and D. Fox, "DefGraspSim: Physics-based simulation of grasp outcomes for 3D deformable objects," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 6274–6281, 2022.

[7] Y. Lin, J. Lloyd, A. Church, and N. F. Lepora, "Tactile gym 2.0: Sim-to-real deep reinforcement learning for comparing low-cost high-resolution robot touch," ser. Proceedings of Machine Learning Research, R. L. A.Banerjee, Ed., vol. 7, no. 4. IEEE, August 2022, pp. 10 754–10 761. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9847020

[8] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, *et al.*, "Isaac gym: High performance gpu-based physics simulation for robot learning," *arXiv preprint arXiv:2108.10470*, 2021.

[9] T. Pfaff, M. Fortunato, A. Sanchez-Gonzalez, and P. Battaglia, "Learning mesh-based simulation with graph networks," in *International conference on learning representations*, 2020.

[10] O. Sorkine-Hornung and M. Rabinovich, "Least-squares rigid motion using SVD," Department of Computer Science, ETH Zurich, Tech. Rep., Jan. 2017. [Online]. Available: https://igl.ethz.ch/projects/ARAP/svd_rotation.pdf

[11] C. Zhang, S. Cui, J. Hu, T. Jiang, T. Zhang, R. Wang, and S. Wang, "Tacflex: Multi-mode tactile imprints simulation for visuotactile sensors with coating patterns," *IEEE Transactions on Robotics*, 2025.

[12] Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li, "On the continuity of rotation representations in neural networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 5745–5753.