Inductive Biases for Predicting Deformation and Stress in Deformable Object Grasps with Graph Neural Networks

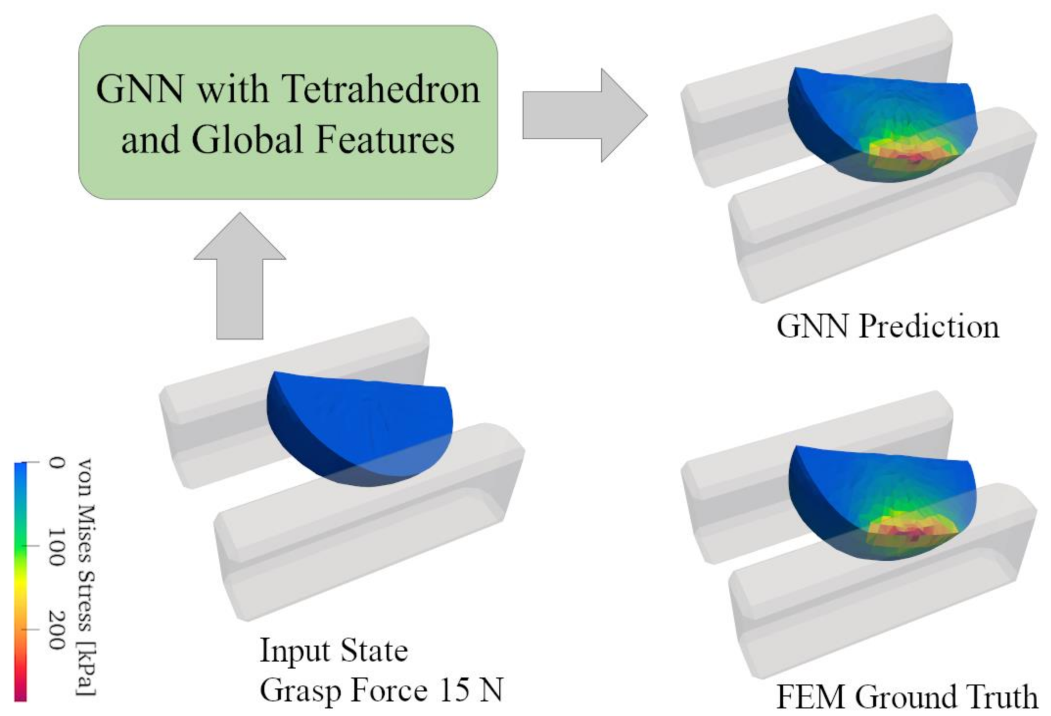
Frederik Heller, Alap Kshirsagar, Tim Schneider, Guillaume Duret and Jan Peters

RoMaDO@ IROS HANGZHOU 2025

TECHNISCHE UNIVERSITÄT DARMSTADT · CENTRALE LYON · IAS · LIRIS

Introduction

- An important task in deformable object manipulation is to predict object deformation and stress
- FEM is "gold standard" given object and gripper meshes
- Recent advances [1] use Graph Neural Networks (GNNs) to learn these fields with good accuracy



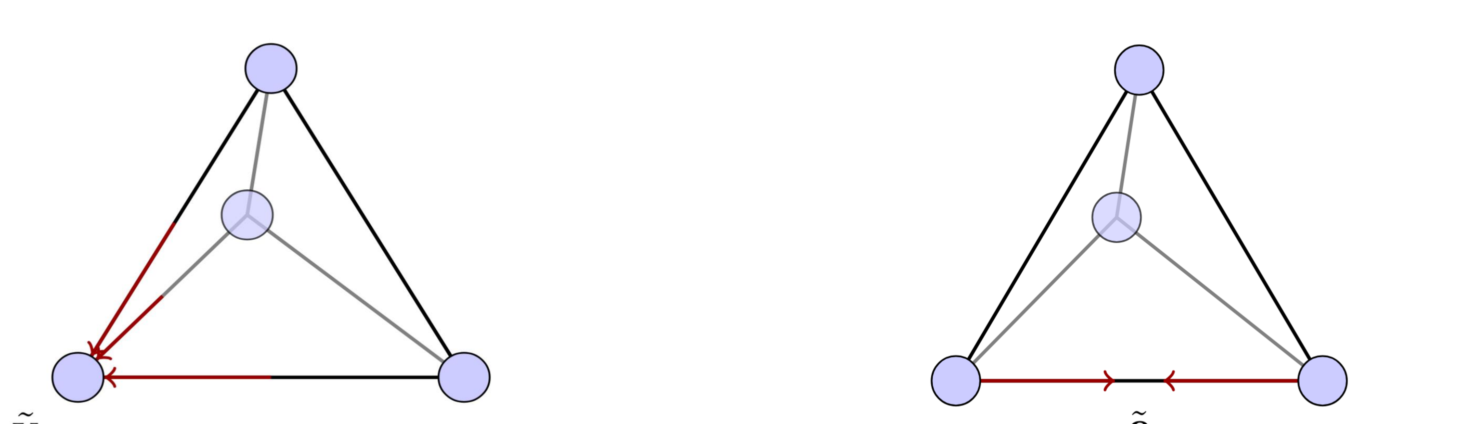
Baseline

Huang et al., 2023: ***DefGraspNets*** [1]

- Dataset generated with FEM simulator *DefGraspSim* [2]
- Object and gripper before grasp expressed as input graph
- Mesh vertices are graph nodes, mesh edges graph edges
- Node and edge features encode relevant information

Encode-Process-Decode GNN architecture [3]:

- Multi-Layer Perceptrons (MLPs) encode features to common 128D latent space
- Message passing propagates information:



$$\tilde{\mathbf{v}}_i' = f^v_{\text{upd}}\left(\tilde{\mathbf{v}}_i, \sum_j \tilde{\mathbf{e}}^M_{ij}, \sum_j \tilde{\mathbf{e}}^W_{ij}\right) + \tilde{\mathbf{v}}_j \qquad \tilde{\mathbf{e}}^{M\prime}_{ij} = f^M_{\text{upd}}\left(\tilde{\mathbf{e}}^M_{ij}, \tilde{\mathbf{v}}_i, \tilde{\mathbf{v}}_j\right)$$

- Decode deformation and stress prediction at each node

Baseline Limitations

1. Released codebase incomplete, e.g. preprocessing for *DefGraspSim* data missing
2. Values can only be predicted at graph nodes, but FEM computes stress values per tetrahedron
3. Slow propagation through network hurts edge cases

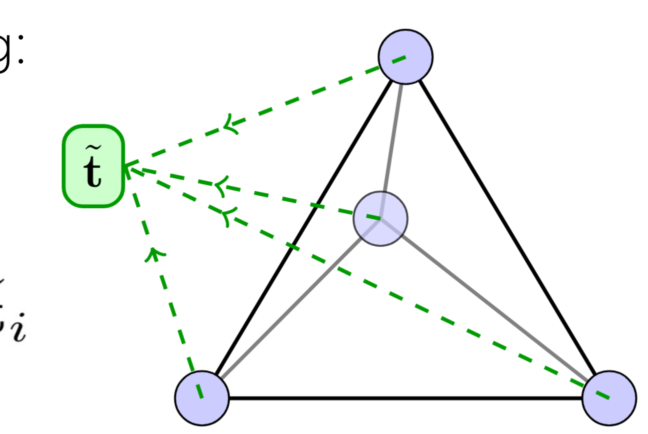
Our Contributions

PyTorch reimplementation of the *DefGraspNets* baseline to a working state to support further research

Tetrahedron features

- Novel architectural extension to the GNN
- GNN is informed about mesh tetrahedrons by giving a tetrahedron set as input, similar to the edge set
- Input feature per tetrahedron, MLP encodes to latent
- Participates in message passing:

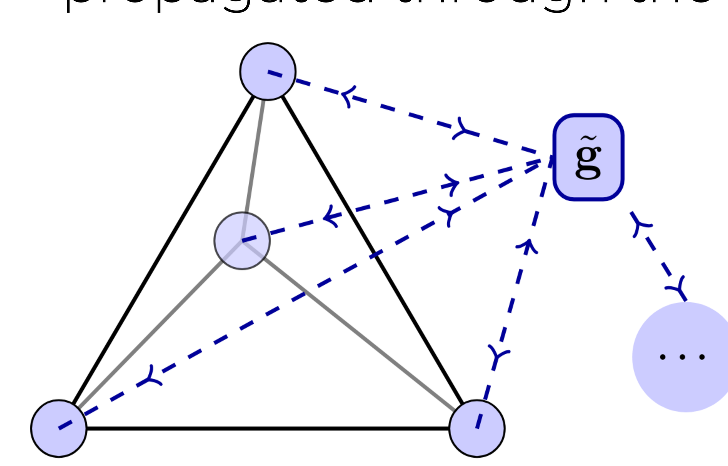
$$\tilde{\mathbf{t}}_i' = f^t_{\text{upd}}\left(\tilde{\mathbf{t}}_i, \{\tilde{\mathbf{v}}_j\}_{j \in \mathbb{T}_i}\right) + \tilde{\mathbf{t}}_i$$



- Decode stress prediction at tetrahedron!

Global feature

- Architectural extension to the GNN
- In message passing, global feature receives information from all nodes in graph, and sends to all nodes
- Act as shortcut for globally relevant information to be propagated through the graph



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

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

- Enables decoding prediction of global value

Evaluation and Results

- *DefGraspSim*-generated dataset of six different objects
- Per object 80 training, 20 test grasps; 50 frames each

MAPE in %	u	$\boldsymbol{\sigma}$	p
Baseline [1]	3.28	1.85	–
A: Tet. Features	3.57	2.28	–
B: Tet. + Global Feat.	**2.81**	**1.31**	**19.77**

Object	8polygon06			cylinder07			lemon01			potato2			sphere03			strawberry01		
MAPE in %	u	$\boldsymbol{\sigma}$	p	u	$\boldsymbol{\sigma}$	p	u	$\boldsymbol{\sigma}$	p	u	$\boldsymbol{\sigma}$	p	u	$\boldsymbol{\sigma}$	p	u	$\boldsymbol{\sigma}$	p
Baseline [1]	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**

Conclusion and Outlook

- Global feature significantly improves performance
- Tetrahedron features allow learning stress as tetra-hedron value, more in line with the FEM model
- Codebase enables further research
  - o Encode material properties in tetrahedrons
  - o Objects with nonisotropic material

Code and paper:

Acknowledgment

We thank Dr. Isabella Huang for providing object geometries and grasp poses from the work *DefGraspNets* [1] for the evaluation of our methods.
This work was supported by the French Research Agency, l'Agence Nationale de Recherche (ANR), and the German Federal Ministry of Education and Research (BMBF) through the project Aristotle (ANR-21-FAI1-0009-01).
We thank Hessisches Ministerium für Wissenschaft & Kunst for the DFKI grant and "The Adaptive Mind" grant.

References

[1] 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.
[2] 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.
[3] T. Pfaff, M. Fortunato, A. Sanchez-Gonzalez, and P. Battaglia, "Learning mesh-based simulation with graph networks," in International conference on learning representations, 2020.