The Journal of Open Source Software

Graphorge: An open-source forge of graph neural networks

Bernardo P. Ferreira [1], **Guillaume Broggi** [1,2], **and Miguel A. Bessa** [1¶]

1 School of Engineering, Brown University, United States of America **2** Aerospace Structures and Materials Department, Faculty of Aerospace Engineering, Delft University of Technology, The Netherlands ¶ Corresponding author

DOI: 10.21105/joss.09453

Software

- Review
- Repository
- Archive

Editor: Julia Romanowska

Reviewers:

- @MSamane
- @marjanalbooyeh

Submitted: 17 September 2025
Published: 24 February 2026

Summary

Graphorge is an open-source Python package built on PyTorch that streamlines the development and evaluation of graph neural networks. It provides a complete workflow encompassing data pre-processing, dataset management, model training, prediction, and result post-processing. Graphorge includes a fully implemented, highly customizable example architecture to demonstrate practical use, and its code is thoroughly documented and commented, making it especially accessible to researchers new to implementing graph neural networks. Although originally developed for computational mechanics, Graphorge's core functionality is built around graph neural networks, which have been successfully applied across a wide range of scientific and engineering domains.

Figure 1: Logo of Graphorge.

Statement of need

Graph neural networks have emerged as a powerful modeling tool for data with relational or geometric structure. Existing graph neural network libraries tend to fall into two categories. On the one hand, many research-specific implementations are minimally documented and tightly tailored to particular benchmarks. While this may enable results reproducibility, code comprehension is often limited, as well as its customization and generalization to different applications. On the other hand, general purpose frameworks such as PyTorch Geometric (Fey & Lenssen, 2019) and Deep Graph Library (Wang et al., 2020) offer robust, high-performance platforms to implement graph neural networks, usually providing multiple backend support and integration of state-of-the-art scientific contributions.

Graphorge is not intended as an alternative to general-purpose frameworks. Instead, it is designed as a practical and educational tool for researchers and students who aim to understand and be able to implement a full Graph Neural Network pipeline. While only including a single, highly customizable architecture (Battaglia et al., 2018) to demonstrate practical use, every single module in Graphorge is extensively documented and commented, aiming to maximize

code comprehension. Moreover, rather than abstracting the workflow behind opaque interfaces, Graphorge provides deliberately fully functional, modular scripts for each stage – from pre-processing and dataset handling to model training and post-processing – making it an ideal extensible, starting point for those looking to implement or customize graph neural networks in a research environment.

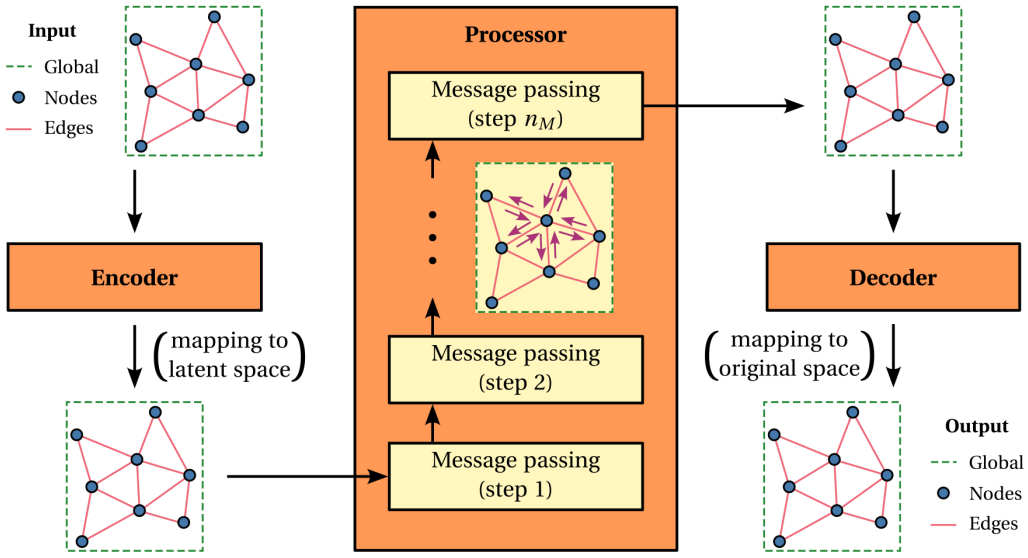

Figure 2: Example of Graph Neural Network model with an encoder-processor-decoder architecture.

Graphorge is built on the graph neural network conceptual framework established by Battaglia and coworkers (Battaglia et al., 2018). In particular, it is originally inspired by the contribution of Sanchez-Gonzalez and coworkers (Sanchez-Gonzalez et al., 2020), namely the supporting code made available through Google DeepMind. In a similar scope, it is worth mentioning meshgraphnets by Pfaff and coworkers (Pfaff et al., 2021), gns by Kumar and Vantassel (Kumar & Vantassel, 2023), and physicsnemo by NVIDIA.

Acknowledgements

Bernardo P. Ferreira and Miguel A. Bessa would like to acknowledge that this effort was undertaken in part with the support from the Department of the Navy, Office of Naval Research, award number N00014-21-2670. We would also like to acknowledge the support from Dassault Systèmes.

References

Battaglia, P. W., Hamrick, J. B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V., Malinowski, M., Tacchetti, A., Raposo, D., Santoro, A., Faulkner, R., Gulcehre, C., Song, F., Ballard, A., Gilmer, J., Dahl, G., Vaswani, A., Allen, K., Nash, C., Langston, V., … Pascanu, R. (2018). *Relational inductive biases, deep learning, and graph networks*. https://arxiv.org/abs/1806.01261

Fey, M., & Lenssen, J. E. (2019). *Fast graph representation learning with PyTorch geometric*. https://arxiv.org/abs/1903.02428

Kumar, K., & Vantassel, J. (2023). GNS: A generalizable graph neural network-based simulator for particulate and fluid modeling. *Journal of Open Source Software*, *8*(88), 5025. https://doi.org/10.21105/joss.05025

Pfaff, T., Fortunato, M., Sanchez-Gonzalez, A., & Battaglia, P. W. (2021). *Learning mesh-based simulation with graph networks*. https://arxiv.org/abs/2010.03409

Sanchez-Gonzalez, A., Godwin, J., Pfaff, T., Ying, R., Leskovec, J., & Battaglia, P. W. (2020). *Learning to simulate complex physics with graph networks*. https://arxiv.org/abs/2002.09405

Wang, M., Zheng, D., Ye, Z., Gan, Q., Li, M., Song, X., Zhou, J., Ma, C., Yu, L., Gai, Y., Xiao, T., He, T., Karypis, G., Li, J., & Zhang, Z. (2020). *Deep graph library: A graph-centric, highly-performant package for graph neural networks*. https://arxiv.org/abs/1909.01315