
ICML 2023 Topological Deep Learning Challenge: Design and Results

Mathilde Papillon^{*1} Mustafa Hajj^{*12} Audun Myers² Florian Frantzen² Ghada Zamzmi² Helen Jenne²
Johan Mathe² Josef Hoppe² Michael Schaub² Theodore Papamarkou² Aldo Guzmán-Sáenz³
Bastian Rieck³ Henry Kvinge³ Jan Meissner³ Karthikeyan Natesan Ramamurthy³ Michael Scholkemper³
Neal Livesay³ Paul Rosen³ Robin Walters³ Shreyas N. Samaga³ Soham Mukherjee³ Sophia Sanborn³
Tamal Dey³ Tegan Emerson³ Tim Doster³ Tolga Birdal³ Vincent Grande³ Abdelwahed Khamis⁴
Abraham Rabinowitz⁴ Aiden Brent⁴ Alessandro Salatiello⁴ Alexander Nikitin⁴ Ali Zia⁴ Claudio Battiloro⁴
Dmitrii Gavrilov⁴ Georg Bökman⁴ German Magai⁴ Gleb Bazhenov⁴ Guillermo Bernardez⁴ Indro Spinelli⁴
Jens Agerberg⁴ Kalyan Nadimpalli⁴ Lev Telyatnikov⁴ Luca Scofano⁴ Lucia Testa⁴ Manuel Lecha⁴
Maosheng Yang⁴ Mohammed Hassanin⁴ Odin Hoff Gardaa⁴ Olga Zaghen⁴ Paul Hausner⁴ Paul Snopoff⁴
Pavlo Melnyk⁴ Rubén Ballester⁴ Sadrodin Barikbin⁴ Sergio Escalera⁴ Simone Fiorellino⁴
Simone Scardapane⁴ Suraj Singh⁴ Tatiana Malygina⁴ Yixiao Yue⁴ Nina Miolane¹

Abstract

This paper presents the computational challenge on topological deep learning that was hosted within the ICML 2023 Workshop on Topology and Geometry in Machine Learning. The competition asked participants to provide open-source implementations of topological neural networks from the literature by contributing to the python packages `TopoNetX` (data processing) and `TopoModelX` (deep learning). The challenge attracted twenty-eight qualifying submissions in its two month duration. This paper describes the design of the challenge and summarizes its main findings. **Code:** <https://github.com/pyt-team/TopoModelX>. **DOI:** 10.5281/zenodo.7958513.

1. Introduction

Graph neural networks (GNNs) have proven to be a powerful deep learning architecture for processing relational data. More specifically, GNNs operate in a graph domain comprised of pairwise relations between nodes. *Topological neural networks* (TNNs) extend GNNs by operating on domains featuring higher-order relations. Such domains, called *topological domains*, feature part-whole and/or set-

type relations (Fig. 1) (Hajj et al., 2023), allowing a more expressive representation of the data. By operating on a topological domain, a TNN leverages the intricate relational structure at the heart of the data. Topological deep learning (Bodnar, 2022; Hajj et al., 2023) has shown great promise in many applications, ranging from molecular classification (Bodnar et al., 2021b) to social network prediction (Schaub et al., 2020). However, the adoption of its architectures, the TNNs, has been limited by the fragmented availability of open-source algorithms and lack of benchmarking between topological domains.

The challenge described in this white paper aims to fill that gap by implementing models in a unifying open source software. In doing so, the challenge contributes to fostering reproducible research in topological deep learning. Participants were asked to contribute code for a published TNN, following `TopoModelX`'s API (Hajj et al., 2023) and computational primitives, and implement a training mechanism for the algorithm's intended task.

This white paper is organized as follows. Section 2 describes the setup of the challenge, including its guidelines and evaluation criteria. Section 3 lists all qualifying submissions to the challenge and its winners.

2. Setup of the challenge

The challenge¹ was held in conjunction with the workshop Topology and Geometry in Machine Learning of the Inter-

^{*}Equal contribution ¹Challenge Organizer ²Challenge Reviewer ³Challenge Contributor ⁴Challenge Participant. Correspondence to: Mathilde Papillon <papillon@ucsb.edu>.

¹Challenge website: <https://pyt-team.github.io/topomodelx/challenge/index.html>

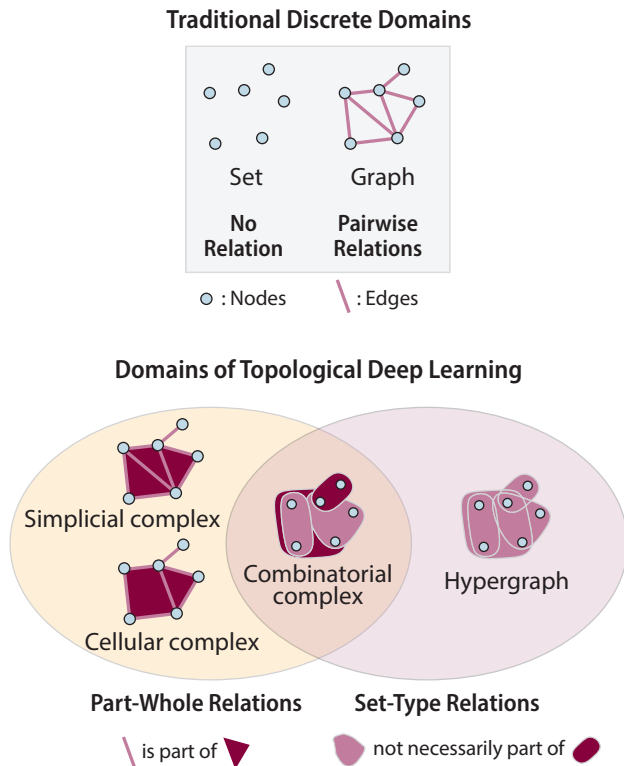


Figure 1. **Domains:** Nodes in blue, (hyper)edges in pink, and faces in dark red. Adapted from (Papillon et al., 2023), originally adapted from (Hajij et al., 2023).

national Conference on Machine Learning (ICML) 2023². Participants were asked to contribute code for a previously existing TNN and train it on a toy dataset.

Guidelines Each submission took the form of an implementation of pre-existing TNN listed in a survey of the field (Papillon et al., 2023). All submitted code was required to comply with TopoModelX’s GitHub Action workflow (Hajij et al., 2023), successfully passing all tests, linting, and formatting.

Each submission consisted of a pull request to TopoModelX containing three new files:

1. A Python script implementing a layer of the model in a single class using TopoModelX computational primitives. One layer is equivalent to the message passing depicted in the tensor diagram representation

²Topology and Geometry in Machine Learning Workshop website: <https://www.tagds.com/events/conference-workshops/tag-ml23>

for the model given in the survey (Papillon et al., 2023).

2. A Jupyter Notebook that builds a neural network out of the single layer, loads and pre-processes the dataset, and performs a train-test loop on this dataset. Defining training and testing in a Jupyter Notebook offers authors a natural way to communicate results that are inherently reproducible, as anyone with access to the notebook may run it to attain the same results.
3. A Python script which contains the unit tests for all functions stored in the class defining the model layer.

Teams were registered to the challenge upon submission of their pull request and there was no restriction on the number of team members, nor on the amount of submissions per team. The principal developers of TopoModelX were not allowed to participate.

Evaluation criteria The evaluation criteria were:

1. Does the submission implement the chosen model correctly, specifically in terms of its message passing scheme? (The training schemes do not need to match that of the original model).
2. How readable and clean is the code? How well does the submission respect TopoModelX’s APIs?
3. Is the submission well-written? Do the docstrings clearly explain the methods? Are the unit tests robust?

Note that these criteria were not designed to reward model performance, nor complexity of training. Rather, these criteria aimed to reward clean code and accurate model architectures that will foster reproducible research in topological deep learning.

Evaluation Method The Condorcet method was used to rank the submissions and decide on the winners. Each team whose submission respected the guidelines was given one vote in the decision process. Nine additional reviewers selected from PyT-team maintainers and collaborators were also each given a vote. Each voted allowed for the selection of best and second best model implementation in each topological domain, representing eight choices in total. Participants were not allowed to vote for their own submissions.

Software engineering practices Challenge participants were also encouraged to use software engineering best practices. All code was to be compatible with Python 3.10 and make an effort to respect the Python style guide PEP8. Datasets had to be loaded from TopoNetX (Hajij et al., 2023) or PyTorch-Geometric (Fey & Lenssen, 2019). Participants could raise GitHub issues and/or request help or guidance at any time by contacting organizers.

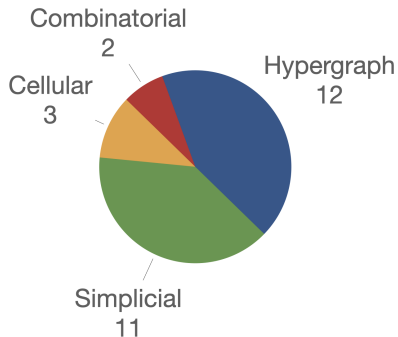


Figure 2. Distribution of submissions across the four topological domains.

3. Submissions and Winners

In total, the challenge received 32 submissions, 28 of which respected the baseline requirements to qualify. Out of the qualifying submissions, 23 unique models were implemented. All four topological domains are represented in this set of models (Fig. 2).

Table 4 lists all qualifying submissions. (Papillon et al., 2023) contains additional information on the architectures and message-passing frameworks for each of these models.

Table 4 also indicates the winning contributions, consisting of a first and second prize for each topological domain, as well as honorable mentions. The winners were announced publicly at the ICML Workshop on Topology, Algebra and Geometry in Machine Learning and on social medias. Regardless of this final ranking, we would like to stress that all the submissions were of very high quality. We warmly congratulate all participants.

Acknowledgments

The authors would like to thank the organizers of the ICML 2023 Topology, Algebra and Geometry in Machine Learning Workshop for their valuable support in the organization of the challenge.

4. Conclusion

This white paper presented the motivation and outcomes of the organization of the Topological Deep Learning Challenge hosted through the ICML 2023 workshop on Topology, Algebra and Geometry in Machine Learning. Challenge submissions implemented a wide variety of topological neural networks into the open-source package `TopoModelX`. We hope that this community effort will foster reproducible research and further methodological benchmark in the growing field of topological deep learning.

Domain	Model	Task Level			Submission authors
		Node	Edge	Complex	
HG	HyperSage (Arya et al., 2020)	✓			German Magai, Paul Snopoff
	AllSetTransformer (Chien et al., 2022)	✓			Luca Scofano, Indro Spinelli, Simone Scardapane, Simone Fiorellino, Olga Zaghen, Lev Telyatnikov, Claudio Battiloro, Guillermo Bernardez (first place)
	AllSetTransformer (Chien et al., 2022)	✓			Luca Scofano, Indro Spinelli, Simone Scardapane, Simone Fiorellino, Olga Zaghen, Lev Telyatnikov, Claudio Battiloro, Guillermo Bernardez
	HyperGat (Ding et al., 2020)	✓			German Magai, Paul Snopoff
	HNHN (Dong et al., 2020)	✓	✓		1. Alessandro Salatiello (hon. mention) 2. Sadrocin Barikbin
	HMPNN* (Heydari & Livi, 2022)	✓			Sadrocin Barikbin (second place)
	UniGCN (Huang & Yang, 2021)	✓			Alexander Nikitin (hon. mention)
	UniSAGE (Huang & Yang, 2021)	✓			Alexander Nikitin
	UniGCNII (Huang & Yang, 2021)	✓			Paul Hausner
	UniGIN (Huang & Yang, 2021)	✓			Kalyan Nadimpalli
	DHGCN* (Wei et al., 2021)			✓	Tatiana Malygina
SC	SCCONV (Bunch et al., 2020)			✓	Abdelwahed Khamis, Ali Zia, Mohammed Hassanin
	SNN (Ebli et al., 2020)		✓		Jens Agerberg, Georg Bökman, Pavlo Melnyk
	SAN (Giusti et al., 2022a)		✓		Luca Scofano, Indro Spinelli, Simone Scardapane, Simone Fiorellino, Olga Zaghen, Lev Telyatnikov, Claudio Battiloro, Guillermo Bernardez (first place)
	SCA (Hajij et al., 2022a)			✓	Aiden Brent (hon. mention)
	Dist2Cycle (Keros et al., 2022)		✓		Ali Zia
	SCoNe (Roddenberry et al., 2021)		✓		1. Odin Hoff Gardaa (second place) 2. Aiden Brent
	SCNN (Yang et al., 2022a)		✓		Maosheng Yang, Lucia Testa
	SCCNN (Yang & Isufi, 2023)		✓		1. Maosheng Yang, Lucia Testa 2. Jens Agerberg, Georg Bökman, Pavlo Melnyk (hon. mention)
	SCN (Yang et al., 2022b)		✓		Yixiao Yue
CC	CWN (Bodnar et al., 2021a)	✓	✓		Dmitrii Gavrilov, Gleb Bazhenov, Suraj Singh (second place)
	CAN (Giusti et al., 2022b)			✓	1. Luca Scofano, Indro Spinelli, Simone Scardapane, Simone Fiorellino, Olga Zaghen, Lev Telyatnikov, Claudio Battiloro, Guillermo Bernardez (first place) 2. Abraham Rabinowitz
CCC	HOAN (Hajij et al., 2022b)	✓	✓		1. Rubén Ballester, Manuel Lecha, Sergio Escalera (first place) 2. Aiden Brent (second place)

Table 1. Model implementations submitted to the Topological Deep Learning Challenge. We organize original models according to domain and task level.

References

- Arya, D., Gupta, D. K., Rudinac, S., and Worring, M. Hypersage: Generalizing inductive representation learning on hypergraphs. *arXiv preprint arXiv:2010.04558*, 2020.
- Bodnar, C. *Topological Deep Learning: Graphs, Complexes, Sheaves*. PhD thesis, Apollo - University of Cambridge Repository, 2022. URL <https://www.repository.cam.ac.uk/handle/1810/350982>.
- Bodnar, C., Frasca, F., Otter, N., Wang, Y., Lio, P., Montufar, G. F., and Bronstein, M. Weisfeiler and Lehman Go Cellular: CW Networks. *Advances in Neural Information Processing Systems*, 34:2625–2640, 2021a.
- Bodnar, C., Frasca, F., Wang, Y., Otter, N., Montufar, G. F., Lio, P., and Bronstein, M. Weisfeiler and Lehman Go Topological: Message Passing Simplicial Networks. In *International Conference on Machine Learning*, pp. 1026–1037. PMLR, 2021b.
- Bunch, E., You, Q., Fung, G., and Singh, V. Simplicial 2-complex convolutional neural networks. In *TDA & Beyond*, 2020.
- Chien, E., Pan, C., Peng, J., and Milenkovic, O. You are allset: A multiset function framework for hypergraph neural networks. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=hpBTiv2uy_E.
- Ding, K., Wang, J., Li, J., Li, D., and Liu, H. Be more with less: Hypergraph attention networks for inductive text classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 4927–4936, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.399. URL <https://aclanthology.org/2020.emnlp-main.399>.
- Dong, Y., Sawin, W., and Bengio, Y. Hnhn: Hypergraph networks with hyperedge neurons. *ICML Graph Representation Learning and Beyond Workshop*, 2020. URL <https://arxiv.org/abs/2006.12278>.
- Ebli, S., Defferrard, M., and Spreemann, G. Simplicial neural networks. In *TDA & Beyond*, 2020.
- Fey, M. and Lenssen, J. E. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.
- Giusti, L., Battiloro, C., Di Lorenzo, P., Sardellitti, S., and Barbarossa, S. Simplicial attention networks. *arXiv preprint arXiv:2203.07485*, 2022a.
- Giusti, L., Battiloro, C., Testa, L., Di Lorenzo, P., Sardellitti, S., and Barbarossa, S. Cell attention networks. *arXiv preprint arXiv:2209.08179*, 2022b.
- Hajij, M., Zamzmi, G., Papamarkou, T., Maroulas, V., and Cai, X. Simplicial complex representation learning. In *Machine Learning on Graphs (MLOG) Workshop at 15th ACM International WSDM (2022) Conference, WSDM2022-MLOG ; Conference date: 21-02-2022 Through 25-02-2022*, January 2022a.
- Hajij, M., Zamzmi, G., Papamarkou, T., Miolane, N., Guzmán-Sáenz, A., and Ramamurthy, K. N. Higher-order attention networks. *arXiv preprint arXiv:2206.00606*, 2022b.
- Hajij, M., Zamzmi, G., Papamarkou, T., Miolane, N., Guzmán-Sáenz, A., Ramamurthy, K. N., Birdal, T., Dey, T. K., Mukherjee, S., Samaga, S. N., Livesay, N., Walters, R., Rosen, P., and Schaub, M. T. Topological deep learning: Going beyond graph data, 2023.
- Heydari, S. and Livi, L. Message passing neural networks for hypergraphs. In Pimenidis, E., Angelov, P. P., Jayne, C., Papaleonidas, A., and Aydin, M. (eds.), *Proceedings of 31st International Conference on Artificial Neural Networks, Part II*, volume 13530 of *Lecture Notes in Computer Science*, pp. 583–592. Springer, 2022. doi: 10.1007/978-3-031-15931-2_48.
- Huang, J. and Yang, J. Unignn: a unified framework for graph and hypergraph neural networks. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, 2021.
- Keros, A. D., Nanda, V., and Subr, K. Dist2cycle: A simplicial neural network for homology localization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 7133–7142, 2022.
- Papillon, M., Sanborn, S., Hajij, M., and Miolane, N. Architectures of topological deep learning: A survey on topological neural networks. *arXiv preprint arXiv:2304.10031*, 2023.
- Roddenberry, T. M., Glaze, N., and Segarra, S. Principled simplicial neural networks for trajectory prediction. In *International Conference on Machine Learning*, pp. 9020–9029. PMLR, 2021.
- Schaub, M. T., Benson, A. R., Horn, P., Lippner, G., and Jadbabaie, A. Random walks on simplicial complexes and the normalized hodge 1-laplacian. *SIAM Review*, 62 (2):353–391, 2020.
- Wei, J., Wang, Y., Guo, M., Lv, P., Yang, X., and Xu, M. Dynamic hypergraph convolutional networks for skeleton-based action recognition. *arXiv preprint arXiv:2112.10570*, 2021.

- Yang, M. and Isufi, E. Convolutional learning on simplicial complexes. *arXiv preprint arXiv:2301.11163*, 2023.
- Yang, M., Isufi, E., Schaub, M. T., and Leus, G. Simplicial convolutional filters. *IEEE Transactions on Signal Processing*, 70:4633–4648, 2022a.
- Yang, R., Sala, F., and Bogdan, P. Efficient representation learning for higher-order data with simplicial complexes. In *Learning on Graphs Conference*, pp. 13–1. PMLR, 2022b.