

Semi-Supervised Graph Structure Learning on Neuromorphic Computers

Guojing Cong
Computer Science and
Mathematics Division
Oak Ridge National Laboratory
Oak Ridge, Tennessee, USA

Seung-Hwan Lim
Computer Science and
Mathematics Division
Oak Ridge National Laboratory
Oak Ridge, Tennessee, USA

Shruti Kulkarni
Computer Science and
Mathematics Division
Oak Ridge National Laboratory
Oak Ridge, Tennessee, USA

Prasanna Date
Computer Science and
Mathematics Division
Oak Ridge National Laboratory
Oak Ridge, Tennessee, USA

Thomas Potok
Computer Science and
Mathematics Division
Oak Ridge National Laboratory
Oak Ridge, Tennessee, USA

Shay Snyder
Department of Electrical and
Computer Engineering
George Mason University
Fairfax, Virginia, USA

Maryam Parsa
Department of Electrical and
Computer Engineering
George Mason University
Fairfax, Virginia, USA

Catherine Schuman
cschuman@utk.edu
Department of Electrical Engineering
and Computer Science
University of Tennessee
Knoxville, Tennessee, USA

ABSTRACT

Graph convolutional networks have risen in popularity in recent years to tackle problems that are naturally represented as graphs. However, real-world graphs are often sparse, which means that implementing them on traditional accelerators such as graphics processing units (GPUs) can lead to inefficient utilization of the hardware. Spiking neuromorphic computers natively implement network-like computation and have been shown to be successful at implementing certain types of graph computations. In this work, we evaluate the use of a simulated network of spiking neurons to perform semi-supervised learning on graph data using only the graph structure. We demonstrate that our neuromorphic approach provides comparable results to graph convolutional network results, and we discuss the opportunities for using neuromorphic computers for this task in the future.

Notice: This manuscript has been authored in part by UT-Battelle, LLC under Contract No. DE-AC05-00OR22725 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (<http://energy.gov/downloads/doe-public-access-plan>).

Publication rights licensed to ACM. ACM acknowledges that this contribution was authored or co-authored by an employee, contractor or affiliate of the United States government. As such, the Government retains a nonexclusive, royalty-free right to publish or reproduce this article, or to allow others to do so, for Government purposes only.

ICONS 2022, July 27–29, 2022, Knoxville, TN, USA

© 2022 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-9789-6/22/07...\$15.00
<https://doi.org/10.1145/3546790.3546821>

CCS CONCEPTS

• **Computing methodologies** → **Neural networks**; *Genetic algorithms*; *Genetic algorithms*; **Neural networks**; **Neural networks**;
• **Hardware** → **Neural systems**; **Neural systems**; **Neural systems**.

KEYWORDS

graph neural networks, neuromorphic computing, spike timing dependent plasticity

ACM Reference Format:

Guojing Cong, Seung-Hwan Lim, Shruti Kulkarni, Prasanna Date, Thomas Potok, Shay Snyder, Maryam Parsa, and Catherine Schuman. 2022. Semi-Supervised Graph Structure Learning on Neuromorphic Computers. In *International Conference on Neuromorphic Systems (ICONS 2022)*, July 27–29, 2022, Knoxville, TN, USA. ACM, New York, NY, USA, 4 pages. <https://doi.org/10.1145/3546790.3546821>

1 INTRODUCTION

Graph convolution extends regular convolution in deep learning to problems naturally represented as graphs (e.g., see [7, 10, 16]). For a wide range of real-world machine learning applications such as text classification [20], traffic pattern prediction [3], material design [18] and drug discovery [15], graph convolutional network (GCN) demonstrates superior performance in comparison to prior approaches. Since their inception, GCNs have received much attention from the research community.

One of the most impressive result that GCNs achieve is in semi-supervised learning (also known as transductive learning), where few data samples are labeled. Graph neural networks (GNNs) effectively leverages both graph structures and graph features (e.g., node features) for learning. As acquiring labels in real-world scenarios is extremely labor intensive and costly, GNNs can play an

important role in semi-supervised learning. For several popular benchmarks (e.g., the citation network datasets [14]), where the number of training samples is below 6%, GNNs are able to achieve high accuracy.

An important challenge in the current GNNs is related to sparsity of real-world graphs. Real-world graphs often exhibit power-law in its connectivity pattern among vertices in a graph. In a power-law graph, a few vertices are highly connected to other vertices, while most of them are sparsely connected to one or two other vertices. Because of this, computation pattern of GNNs is often irregular, depending upon sub-regions of the target graph. In turn, such an irregular computation pattern leads to inefficient utilization of popular hardware accelerators like graphics processing units (GPUs).

At the same time, neuromorphic computers have increasingly been used to perform a variety of graph algorithms, including shortest path [2, 11] and minimum spanning tree [9]. Neuromorphic computers have several qualities that make them attractive for implementation of graph operations, including massively parallel computation, event-driven operation, and collocated processing and memory [12]. Moreover, neuromorphic computers can potentially operate with orders of magnitude less power than their conventional computing counterparts.

In this work, we investigate whether the properties of neuromorphic computers can be used to address the same semi-supervised challenges that GNNs address. Rather than attempting to map GNN algorithms onto neuromorphic implementations directly, instead we develop our own approach that leverages the properties of networks of spiking neurons with plastic synapses to address this task. We present a neuromorphic algorithm for using networks *without* features to predict node labels.

2 METHOD

In this work, we consider three citation graph datasets [13]: Cora, Citeseer, and Pubmed. In these datasets, nodes correspond to documents, edges correspond to citation links, and each node has a sparse bag-of-words feature vector as well as a class label. The characteristics of these datasets are summarized in Table 1. For each of the three citation network graphs, we randomly select 20 papers per topic to serve as our training set.

	Cora	Citeseer	Pubmed
# nodes	2,708	3,327	19,717
# edges	5,278	4,552	44,324
# features	1,433	3,703	500
# classes	7	6	3
# nodes in training	140	120	60
# nodes in test	1,000	1,000	1,000

Table 1: Characteristics of the three citation networks

For each citation network graph, we construct a spiking neural network in the NEST neural simulator [5]. We create a neuron for each node/paper in the citation network, and we create a neuron for each possible topic. We turn off leak for both sets of neurons by setting their τ_m (membrane time constants) value to 100,000 ms,

and we set their reset voltages to 0 mV and their threshold voltages to 1 mV. For the test neurons, we set their refractory periods to be very high so that they will only fire once. For each neuron, we also create a spike recorder to track their firing output. Though we track all firing output across the network, we do not use the output fires of any of the neurons in our predictions.

For each edge connecting nodes A and B in the citation network, we create two synapses: one from A to B and one from B to A. We set these synapses to have a weight of 100 and a delay of 1 ms. Since we know the topics of the papers in the training set, we create two synapses between paper N’s neuron and topic T’s neuron: one from N to T and one from T to N, each with a weight of 1 and delay of 1. Finally, because the topics are not known for the remainder of the papers (the testing set), we create a synapse to and from each of the topics for each paper neuron in the testing set. Those synapses have a weight of 0.0001 and a delay of 1 ms. The synapses between papers and topics all have STDP turned on, with a τ_- (time constant for the depression part of STDP window) value of 30 ms. An example of this process is shown in Figure 1.

To predict the topic of each paper in the testing set, we create the network as defined above, and then we create a single spike on the corresponding testing neuron at simulation time 1. We allow the network to simulate for 20 time steps. Then, we look at the synapses between all of the topics and that test paper (from the topic to the test paper). The synapse with the lowest weight value is our predicted topic. It is worth noting that we could similarly check the highest weighted synapse between the test paper and all of the topics. The topic with the lowest weighted synaptic connection to the test paper is the predicted topic for that paper. We repeat this approach for each of the papers in the test set to obtain our testing accuracy.

It is worth noting that the network construction process here can be extended to any semi-supervised graph structure learning problem. In these problems, the goal is to predict other node labels in a graph based on a training set of labels of the nodes. In this particular case, the labels are the topics, and the nodes represent papers. The network construction process described above can be applied to any other semi-supervised graph structure by constructing networks where both the nodes and the labels are constructed as neurons, and the connectivity of the network is based on connectivity in the graph and the known labels from the training set.

3 PRELIMINARY RESULTS

In Table 1, only a small fraction of nodes, that is, 20 per class, are included in the training set. Some recent studies [8, 17] report significantly higher accuracies than Kipf and Welling [10], and Veličković *et al.* [16]. We note that these studies use more training data [17] and/or stronger optimizers [8].

We show the results of our neuromorphic approach in Table 2. Because we use a random train/test split (where 20 random papers are selected for each topic on each run), the neuromorphic results reported are the average accuracy across five runs. As can be seen in the table, for the Cora dataset, the results are comparable with GCN, but neither of those are approaching the state-of-the-art results

*This is the result at time of submission for 6300 samples in the testing set. We expect this result to decrease a small amount as the remaining test results are collected.

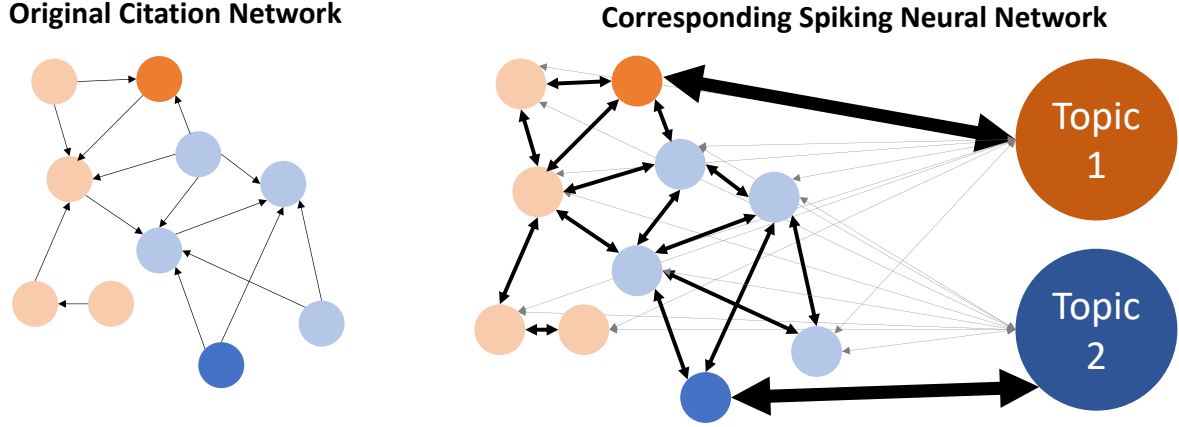


Figure 1: Example of a citation network converted to a spiking neural network for node classification label prediction. The two nodes belonging to the training set are shown as darker colors and are strongly connected to their corresponding topic neurons. All other paper nodes are weakly connected to both topics.

	Cora	Citeseer	Pubmed
<i>Node2Vec</i> [6]	0.71	0.48	0.70
<i>node2vec-a</i> [1]	0.68	0.51	-
<i>Planetoid-G</i> [19]	0.69	0.49	0.66
<i>GraphSAGE</i> [7]	0.71	0.48	0.64
<i>GCN</i> [4]	0.59	0.34	0.42
<i>Neuromorphic</i>	0.56	0.41	0.54*

Table 2: Test accuracy comparison with other implementations on plain input citation graphs (without node features)

without node features. It is worth noting here that we are primarily targeting comparing against GCN and not the other approaches, though we include those for reference.

It is also worth noting that we are not optimizing any of the hyperparameters associated with the neuromorphic approach. We expect to see an increase in performance when we tune the hyperparameters of this approach.

4 DISCUSSION AND FUTURE WORK

In this work, we leverage spike-timing dependent plasticity as a way to perform semi-supervised learning in graphs. We show that even with an initial implementation without any tuning of the parameters of the spiking neural network, we can achieve accuracies that outperform other implementations. We are still far from state-of-the-art accuracy on these tasks; as such, there are several future directions that we intend to investigate to improve performance.

We will implement a Bayesian meta learner along with the spike-timing dependent plasticity to optimize the hyperparameters and adaptively guide the learning toward optimum performance. We envision that this hierarchical learning paves the way to not only improve the plasticity learning for GNNs, but also optimize several performance metrics such as accuracy, latency, and resiliency.

We restricted our attention in this work to learning based on the structure of the graph only. In future work, we intend to incorporate

in the feature information of the graph as well as part of the learning process. We will investigate several approaches for leveraging the features in a spiking neural network implementation, including representing the feature vectors as spike patterns on each of the nodes, representing features as separate nodes in the graph that are connected to the nodes that include those features, and representing features as synaptic connections between nodes that share those features. It is not clear which of these approaches will be most appropriate, nor is it clear what the performance implications in terms of energy usage or network size and density will be. Metrics such as these will be critical to evaluate as we move to neuromorphic hardware rather than neuromorphic simulation.

We plan to compare and contrast learning on graphs through SNNs vs. GNNs on traditional architectures such as GPUs or CPUs. We need to understand the fundamental properties of graph learning with SNNs. For example, what are the canonical ways to represent graphs and the corresponding paradigm; mathematically, what are the properties of the class of functions learned through SNN in comparison to GNNs; are the issues in (vanilla) GNNs such as over-smoothing present also with SNNs, if yes, how to address this issue; and are there fundamental advantages and disadvantages of learning with SNNs. From our preliminary study with the citation network it is already clear that although both approaches operate on the graph (e.g., nodes) and leverage the structure for learning, there are very clear differences. With SNNs there is a natural recurrence through the propagation of spikes that for a node in the graph may receive information from another node several hops away. In GNNs, this is achieved through adding more layers, and that may result in over-smoothing. So it is reasonable to expect that learning achieved on SNNs may be quite different than that on GNNs (and we cautiously believe there can be advantages for learning on SNNs). There are also challenges associated with SNNs as research in this area is in its nascent stage. Research in theory, software support, and application study is needed for graph learning with SNNs.

ACKNOWLEDGMENTS

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, under contract number DE-AC05-00OR22725.

This research used resources of the Compute and Data Environment for Science (CADES) at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.

REFERENCES

- [1] Rami Abu-El-Hajja, Bryan Perozzi, Rami Al-Rfou, and Alex Alemi. 2018. Watch Your Step: Learning Node Embeddings via Graph Attention. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems* (Montréal, Canada) (NIPS'18). Curran Associates Inc., Red Hook, NY, USA, 9198–9208.
- [2] James B Aimone, Yang Ho, Ojas Parekh, Cynthia A Phillips, Ali Pinar, William Severa, and Yipu Wang. 2020. Provable neuromorphic advantages for computing shortest paths. In *Proceedings of the 32nd ACM Symposium on Parallelism in Algorithms and Architectures*. 497–499.
- [3] Fanglan Chen, Zhiqian Chen, Subhodip Biswas, Shuo Lei, and Naren Ramakrishnan. 2020. Graph Convolutional Networks with Kalman Filtering for Traffic Prediction. In *Proceedings of the 28th International Conference on Advances in Geographic Information Systems* (Seattle, WA, USA) (SIGSPATIAL '20). Association for Computing Machinery, New York, NY, USA, 135–138.
- [4] Guojing Cong and Seung-Hwan Lim. 2021. Versatile feature learning with graph convolutions and graph structures. In *Proceedings of 2021 International Conference on Data Mining Workshops (ICDMW)*. 669–677.
- [5] Robin de Schepper, Jochen Martin Eppler, Anno Kurth, Pooja Nagen-dra Babu, Rajalekshmi Deepu, Sebastian Spreizer, Guido Trench, Jari Pronold, Stine Brekke Vennemo, Steffen Graber, Aitor Morales-Gregorio, Charl Linssen, Mohamed Ayssar Benelhedi, Håkon Mørk, Abigail Morrison, Dennis Terhorst, Jessica Mitchell, Sandra Diaz, Itaru Kitayama, Mahdi Enan, Nilton Liuji Kamiji, and Hans Ekkehard Plesser. 2022. NEST 3.2. <https://doi.org/10.5281/zenodo.5886894>
- [6] Aditya Grover and Jure Leskovec. 2016. Node2vec: Scalable Feature Learning for Networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (San Francisco, California, USA) (KDD '16). Association for Computing Machinery, New York, NY, USA, 855–864. <https://doi.org/10.1145/2939672.2939754>
- [7] William L. Hamilton, Rex Ying, and Jure Leskovec. 2017. Inductive Representation Learning on Large Graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (Long Beach, California, USA) (NIPS'17). Curran Associates Inc., Red Hook, NY, USA, 1025–1035.
- [8] Mohammad R. Izadi, Yihao Fang, Robert Stevenson, and Lizhen Lin. 2020. Optimization of Graph Neural Networks with Natural Gradient Descent. In *2020 IEEE International Conference on Big Data (Big Data)*. 171–179. <https://doi.org/10.1109/BigData50022.2020.9378063>
- [9] Bill Kay, Prasanna Date, and Catherine Schuman. 2020. Neuromorphic graph algorithms: Extracting longest shortest paths and minimum spanning trees. In *Proceedings of the Neuro-inspired Computational Elements Workshop*. 1–6.
- [10] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *Proceedings of the 5th International Conference on Learning Representations* (Palais des Congrès Neptune, Toulon, France) (ICLR '17). <https://openreview.net/forum?id=SJU4ayYgl>
- [11] Catherine D Schuman, Kathleen Hamilton, Tiffany Mintz, Md Musabbir Adnan, Bon Woong Ku, Sung-Kyu Lim, and Garrett S Rose. 2017. Shortest path and neighborhood subgraph extraction on a spiking memristive neuromorphic implementation. In *Proceedings of the 7th Annual Neuro-inspired Computational Elements Workshop*. 1–6.
- [12] Catherine D Schuman, Thomas E Potok, Robert M Patton, J Douglas Birdwell, Mark E Dean, Garrett S Rose, and James S Plank. 2017. A survey of neuromorphic computing and neural networks in hardware. *arXiv preprint arXiv:1705.06963* (2017).
- [13] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, and et al. 2008. Collective Classification in Network Data. *AI Magazine* 29, 3 (2008), 93–106. <http://www.cs.iit.edu/~ml/pdfs/sen-aimag08.pdf>
- [14] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. 2008. Collective Classification in Network Data. *AI Magazine* 29, 3 (Sep. 2008), 93. <https://doi.org/10.1609/aimag.v29i3.2157>
- [15] Mengying Sun, Sendong Zhao, Coryandar Gilvary, and et al. 2019. Graph convolutional networks for computational drug development and discovery. *Briefings in Bioinformatics* 21, 3 (06 2019), 919–935. <https://doi.org/10.1093/bib/bbz042> <https://academic.oup.com/bib/article-pdf/21/3/919/3322726/bbz042.pdf>
- [16] Petar Veličković, Guillem Cucurull, Arantxa Casanova, and et al. 2017. Graph Attention Networks. *6th International Conference on Learning Representations* (2017).
- [17] Hongwei Wang and Jure Leskovec. 2020. Unifying Graph Convolutional Neural Networks and Label Propagation. *CoRR abs/2002.06755* (2020). [arXiv:2002.06755](https://arxiv.org/abs/2002.06755)
- [18] Tian Xie and Jeffrey C. Grossman. 2018. Crystal Graph Convolutional Neural Networks for an Accurate and Interpretable Prediction of Material Properties. *Phys. Rev. Lett.* 120 (Apr 2018), 145301. Issue 14. <https://doi.org/10.1103/PhysRevLett.120.145301>
- [19] Zhilin Yang, William W. Cohen, and Ruslan Salakhutdinov. 2016. Revisiting Semi-Supervised Learning with Graph Embeddings. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48* (New York, NY, USA) (ICML'16). JMLR.org, 40–48.
- [20] Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Graph Convolutional Networks for Text Classification. In *AAAI*.