Graph Representation Learning

Davide Bacciu¹, Federico Errica², Alessio Micheli¹, Nicolò Navarin³, Luca Pasa³, Marco Podda¹ and Daniele Zambon⁴

1- Department of Computer Science, University of Pisa, Italy

2- NEC Laboratories Europe, Germany

3- Department of Mathematics, University of Padova, Italy

4- The Swiss AI Lab IDSIA & Università della Svizzera italiana, Switzerland

Abstract. In a broad range of real-world machine learning applications, representing examples as graphs is crucial to avoid a loss of information. For this reason, in the last few years, the definition of machine learning methods, particularly neural networks, for graph-structured inputs has been gaining increasing attention. In particular, Deep Graph Networks (DGNs) are nowadays the most commonly adopted models to learn a representation that can be used to address different tasks related to nodes, edges, or even entire graphs. This tutorial paper reviews fundamental concepts and open challenges of graph representation learning and summarizes the contributions that have been accepted for publication to the ESANN 2023 special session on the topic.

1 Introduction

In this tutorial paper, we introduce the core concepts related to Graph Representation Learning, which is the subject of a special session organized by the authors at the 31^{th} European Symposium on Artificial Neural Networks, Computational Intelligence, and Machine Learning. Many critical real-world applications produce data that can be naturally represented by complex structures such as graphs. Graphs are particularly suited to represent relations (edges) between the components (nodes) constituting an entity.

In this paper, we focus our attention on applying Deep Learning models for learning in graph domains, where Deep Graph Networks (DGNs) [1] are nowadays the *de facto* standard. The origins of DGNs can be traced back to works on directed acyclic graphs [2, 3], where recursive neural networks were used to automatically extract information from structured data to solve the task at hand. Current DGNs are capable of modeling cyclic graphs, and the pioneering methods for this class of models are based on the recurrent [4] and convolutional [5] paradigms of computation.

DGNs are composed by several layers that can learn representations that embed information about the entities and their relations. The DGNs can be divided into three broad categories: (i) the models inspired by neural architectures [6, 7, 8, 9], (ii) the probabilistic models of graphs [10], and (iii) the hybrid models that leverage both neural and probabilistic models to generate graphs [11].

Graph-structured data are ubiquitous in nature, therefore there are a vast number of possible real-world applications of DGNs. These applications can

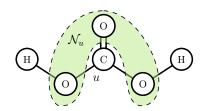


Fig. 1: Carbonic acid molecule represented as a 2D graph where nodes denote atoms with their type. The neighborhood \mathcal{N}_u of node u is marked by the green dashed area. Edges represent single or double bonds.

be associated with nodes, edges, or properties of entire graphs. One of the most common applications of DGNs concerns the predictions over nodes in a network. An example in this setting is the prediction of properties of a social network user based on his connections. Another related task involves discovering the relationships in a social network by performing an edge-level prediction task. Finally, DGNs can also be used to tackle graph-level prediction tasks. In this setting, each example is composed of a whole graph, and the learning tasks are predictions of the properties of the whole graphs. An example is the prediction of the toxicity in humans of chemical compounds represented by their molecular graph, of which we show an example in Figure 1.

2 Background Notions

Before describing the functioning of graph representation learning methods, we first present basic definitions about graphs that are used in the contributions of this special session. A commonly used formalization of a (static) graph is the tuple $g = (\mathcal{V}, \mathcal{E}, \mathcal{X}, \mathcal{A})$, with \mathcal{V} being the set of nodes/vertices and \mathcal{E} representing the set of edges/arcs that connect pairs of nodes. Depending on whether a graph is **undirected** (resp. **directed**), an edge reflects the *unordered* (resp. *ordered*) interaction between two nodes $u, v \in \mathcal{V}$. The sets \mathcal{X} and \mathcal{A} specify the domain of node and edge features/attributes, respectively. In the literature, it is very common to represent attributes on nodes and edges as vectors, but discrete edge types are also frequently used to encode various kinds of information, for example a covalent or non-covalent interaction between atoms in chemistry and physics. In addition, a Knowledge Graph (KG) can be seen as a graph where connections represent facts about two entities. In this setup, it is often the case that entities have no features and there are numerous (discrete) relations that can connect two entities.

The adjacency matrix $\mathbf{A} \in \{0, 1\}^{|\mathcal{V}| \times |\mathcal{V}|}$ represents the (binary or weighted) connectivity information of a single graph. Assuming $\mathcal{V} = \{1, 2, ..., n\}$, an entry u, v of \mathbf{A} is non-zero when nodes u and v are connected. Consequently, adjacency matrices are symmetric when the graph is undirected. At the same time, the adjacency matrix might not be the best way to represent connectivity

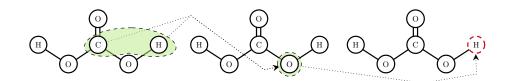


Fig. 2: We show the contextual information propagation to the hydrogen molecule (dashed red) of the Carbonic acid. Following two graph convolutions, the hydrogen atom can access the information at 2-hop distance from itself.

information, especially when the proportion of edges compared to nodes is small; in this case, we talk about a **sparse** graph. For sparse graphs, a convenient representation is an adjacency list specifying the list of *neighbors* for each node. The neighborhood of a node v is defined as the set of nodes directly connected to it: $\mathcal{N}_v = \{u \in \mathcal{V} \mid (u, v) \in \mathcal{E}\}.$

There are equivalent, namely **isomorphic**, ways to construct the same graph, which give rise to the well-studied graph isomorphism problem [12]. In particular, we can construct the same graph by applying the same random permutation to the rows and columns of the adjacency matrix and the feature matrix $\mathbf{X} \in \mathbb{R}^{|\mathcal{V}| \times d}$, where $\mathcal{X} \subseteq \mathbb{R}^{d}$.

We can extend graphs (and knowledge graphs) to the temporal setting by adding a time-stamp to each node/edge. If time is discretized, one talks about graph *snapshots*, meaning that at each time t we consider a graph with new nodes and edges. When time is continuous, we model the evolution of the graph as a stream of *events*.

Finally, we define the notion of permutation invariance and graph equivariance, which are extremely useful in graph representation learning and used in the contributions to the special session. A function is said permutation invariant if it does not change its output when the components of the input are reordered according to some permutation. The sum, the mean, the maximum and the product operators are straightforward examples of permutation invariant functions. In the context of graphs, permutation invariance is intended with respect to the permutation of the nodes. There exist DGNs that are also *equivariant*, meaning that their output is subject to the same transformation as the input: this kind of inductive bias is useful, for instance, in the prediction of chemical forces, where rotating a molecule rotates the force (or equivalent the acceleration) vector associated with it [13].

3 What is a Graph Convolution?

For the purpose of the tutorial, we will introduce the graph convolution operator in the context of message-passing architectures [14] since these represent the most prominent class of models in the literature, even though alternative approaches are possible, e.g. based on the multi-resolution approach [15, 16].

A graph convolution is an operation where a function is applied to the nodes of a graph akin to a convolutional filter on the pixels of an image. The fundamental difference is that the neighborhood of each node is irregular and might evolve over time in unpredictable ways. The framework of *message passing* [17] consists of two steps: first, at each graph convolutional layer ℓ , a message is computed for each node v relying on its current state (or embedding) \mathbf{h}_v^{ℓ} and neighboring information; after that, the message is dispatched to neighboring nodes according to the topological information of the graph. In the second step, each node collects the incoming messages and uses them to update its own embedding. Formally, we can write this process as

$$\mathbf{h}_{v}^{\ell+1} = \phi^{\ell+1} \Big(\mathbf{h}_{v}^{\ell}, \ \Psi(\{\psi^{\ell+1}(\mathbf{h}_{u}^{\ell}, \mathbf{a}_{uv}) \mid u \in \mathcal{N}_{v}\}) \Big), \tag{1}$$

where ϕ, ψ are learnable functions such as neural networks, Ψ is a permutation invariant function like the sum or the mean, \mathbf{a}_{uv} is the edge feature vector. At layer $\ell = 0$, the node v's embedding corresponds to the node feature vector \mathbf{x}_v^0 .

By stacking graph convolutions on top of each other, one can propagate information across the graph and build rich node embeddings. These embeddings can be fed into standard machine learning predictors for node and link prediction tasks. When tackling whole-graph prediction problems, an additional step of global pooling is required before being able to make predictions:

$$\mathbf{h}_{g}^{\ell} = \Psi\Big(\{f(\mathbf{h}_{v}^{\ell}) \mid v \in \mathcal{V}_{g}\}\Big),\tag{2}$$

where, with slight abuse of notation, Ψ is another permutation invariant function and f is another learnable function. Typically, the last layer representation is used to make predictions, but the concatenation across layers is also a viable and effective alternative [10, 18].

Message passing has become a successful framework for graph representation learning due to several key characteristics. First and foremost, message passing breaks down the cyclic dependencies within a graph that, under the above local processing assumption, cause an infinite (recursive) definition of a node's embedding. By iteratively passing messages between neighboring nodes, as shown in Figure 2 each node can incorporate and disseminate contextual information from its local neighborhood, enabling (at least in principle) a holistic understanding of the graph's structure and features.

Akin to convolutions on images, each message passing step is amenable to parallel computation on all nodes, which is crucial when handling large-scale graphs. Because message passing operates locally on individual nodes, computations can be easily distributed across multiple processors or computing units, resulting in significant speed-ups.

Another advantage of message passing is its invariance to the permutation of nodes. By leveraging the permutation invariant function Ψ , the neighborhood aggregation learns representations that remain consistent regardless of the node ordering. This invariance to permutation is particularly useful because cyclic graphs do not come with a pre-defined node ordering.

Lastly, message passing is agnostic to the size of the graph and the neighborhood of each node. Whether the graph is sparse or dense, and regardless of the number of neighbors each node has, the message passing framework can adapt and accommodate various graph topologies. This flexibility is crucial for generalizing to unseen graph topologies.

In summary, the success of message passing stems from its ability to break down cycles, facilitate efficient parallel computation, exhibit invariance to node permutations, and handle graphs of different sizes and neighborhoods. These characteristics make it a powerful and versatile framework for graph representation learning. At the same time, message passing has its own limitations to be still solved, and we mention in particular oversmoothing and oversquashing. Oversmoothing occurs when the iterative message passing process ultimately results in overly similar representations for different nodes in the graph. As the number of message passing steps increases, the learned representations may become increasingly indistinguishable, losing the ability to capture subtle differences and fine-grained features that are essential for accurate downstream tasks. On the other hand, oversquashing refers to the problem of information compression or loss during the message passing process. As messages are aggregated and combined, there is a risk of losing important details and nuances present in the original node features. Oversquashing can be particularly problematic when dealing with graphs that exhibit high topological complexity.

4 **Promising Directions**

Learning relational information The success achieved by DGNs largely relies on the suitability of the considered relational information which, not rarely, is incomplete or completely missing at design time. As demonstrated in the literature, addressing link prediction and multi-node representation tasks following standard graph deep learning setups turned out to be cursed by inherent limitations [19]. In addition, graph transformers – despite their ability to automatically extract relevant pairwise relations – suffer from poor scalability with larger graphs [20], while enabling efficient learning through sparse computational graphs requires dedicated gradient estimators [21].

Spatio-temporal data Graph deep learning has recently become popular in the processing of spatio-temporal data too, which involves interconnected entities generating data observations over time, such as those coming from meteorological monitoring, transportation networks, epidemic models, and social sciences [22, 23, 24]. In this context, new challenges emerge primarily associated with the possible appearance of new nodes and changes in the relational information over time, missing data, irregular sampling, and scalability [25, 26, 27], showing that there is still a long way to go in terms of developing mature solutions.

Probabilistic models As per any machine learning problem, dealing with uncertainty associated with the data-generating process and the learned model can be

the key to accurate and trustworthy DGN solutions. Despite the recent progress in integrating DGNs with probabilistic components [28, 29, 30], estimating the uncertainty associated with combinatorial objects like the adjacency matrix and estimating non-factorized probability distribution over graphs are two of the most challenging problems.

Biophysics and chemistry Lately, there has been growing research focus on employing DGNs to model complex interactions in physical and biological systems, where the use of graph-based processing holds great potential for, e.g., scalable computer-aided simulations of fluid dynamics, prediction of molecule properties, analyzing extensive databases derived from particle physics experiments, and discovery and repurposing of drugs [31, 32, 33, 34]. Indeed, these applications further emphasize the importance of the above-mentioned research directions.

5 Special Session's Contributions

This year's special session comprises several original contributions ranging on a set of diverse topics related to the graph representation learning field:

- Landolfi et al. [35] investigate the relationship between Graph Neural Networks and tropical algebra in the context of learning Dynamic Programming (DP) algorithms [36]. Specifically, they show that a Graph Isomorphism Network (GIN) [37] coupled with a proper encoder and decoder functions can approximate several interesting DP algorithms on graphs up to arbitrary precision, effectively drawing a connection between the two fields.
- Wang et al. [38] present a method to extract categorical embeddings from data by applying Differential Pooling (DiffPool) [39] on a graph where the adjacency matrix is the co-occurrence matrix of categorical values, and the feature matrix is the one-hot encoding of the categorical features.
- Errica et al. [40] propose a deep Hidden Markov Model for temporal graphs. The model consists of a stack of layers which perform probabilistic message passing, trained with Expectation-Maximization. They apply the model to temporal node prediction tasks, with results comparable to neural approaches.
- Tortorella et al. [41] analyse the richness of the node embeddings learned by Graph Echo State Networks [6]. Specifically, they study how the quality of the node embeddings affects the accuracy in classification tasks. To facilitate the analysis, they propose an entropy-based measure to quantify the information carried by a node embedding. With the aid of this tool, they compare for the first time different reservoir designs for Graph Echo State Networks on node classification tasks.

- Torres et al. [42] propose a DGN-Transformer architecture specifically designed for the prediction of toxicity and side-effects of chemical compounds, where labeled data is historically scarce. In the few-shot scenario, they show that the proposed architecture can outperform standard DGNs.
- Vu et al. [43] present an architecture for link prediction on knowledge graphs. In order to learn better entity-relation embeddings, they first transform the entity embeddings similarly to 2D images, then apply a combination of MetaFormer and Fast Fourier Transform layers as feature extractors, outperforming common baselines.
- Navarin et al. [44] investigate the over-parameterization regime in the context of untrained graph neural models. In particular, they combine an approximation of the algorithmic stability measure with empirical evidence to understand when over-parameterization allows these models to generalize better.

6 Conclusions

With this tutorial and the corresponding ESANN 2023 special session, we aim to give an overview of the machine learning models and techniques that are nowadays applied in learning from graphs. Indeed, graphs can be the key to solving complex problems related to fields where the intrinsic nature of the datum is relational, such as biology, network science, graphics as well as chemistry. Among the recent research directions that have drawn attention, we highlighted some of the most promising ones. In the final section of the article, a summary of the interesting papers presented at the special session is provided.

References

- Davide Bacciu, Federico Errica, Alessio Micheli, and Marco Podda. A gentle introduction to deep learning for graphs. *Neural Networks*, 129:203–221, 9 2020.
- [2] Alessandro Sperduti and Antonina Starita. Supervised neural networks for the classification of structures. *IEEE Transactions on Neural Networks*, 8(3):714–735, 1997.
- [3] Paolo Frasconi, Marco Gori, and Alessandro Sperduti. A general framework for adaptive processing of data structures. *IEEE Transactions on Neural Networks*, 9(5):768–786, 1998.
- [4] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions* on Neural Networks, 20(1):61–80, 2009.
- [5] Alessio Micheli. Neural network for graphs: A contextual constructive approach. *IEEE Transactions on Neural Networks*, 20(3):498–511, 2009.

- [6] Claudio Gallicchio and Alessio Micheli. Graph echo state networks. In IJCNN, pages 1–8. IEEE, 2010.
- [7] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.
- [8] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. In *ICLR*, 2018.
- [9] Claudio Gallicchio and Alessio Micheli. Fast and deep graph neural networks. In AAAI, 2020.
- [10] Davide Bacciu, Federico Errica, and Alessio Micheli. Probabilistic learning on graphs via contextual architectures. *Journal of Machine Learning Research*, 21(134):1–39, 2020.
- [11] Federico Errica, Davide Bacciu, and Alessio Micheli. Graph mixture density networks. In Proceedings of the 38th International Conference on Machine Learning (ICML), pages 3025–3035, 2021.
- [12] Johannes Kobler, Uwe Schöning, and Jacobo Torán. *The graph isomorphism problem: its structural complexity.* Springer Science & Business Media, 2012.
- [13] Simon Batzner, Albert Musaelian, Lixin Sun, Mario Geiger, Jonathan P Mailoa, Mordechai Kornbluth, Nicola Molinari, Tess E Smidt, and Boris Kozinsky. E (3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nature communications*, 13(1):2453, 2022.
- [14] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural Message Passing for Quantum Chemistry. In *ICML*, pages 1263—1272, apr 2017.
- [15] Luca Pasa, Nicolò Navarin, and Alessandro Sperduti. Polynomial-based graph convolutional neural networks for graph classification. *Machine Learning*, pages 1–33, 2022.
- [16] Luca Pasa, Nicolò Navarin, and Alessandro Sperduti. Multiresolution reservoir graph neural network. *IEEE Transactions on Neural Networks and Learning Systems*, 33(6):2642–2653, 2021.
- [17] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In Proceedings of the 34th International Conference on Machine Learning (ICML), pages 1263–1272, 2017.
- [18] Nicolò Navarin, Dinh Van Tran, and Alessandro Sperduti. Learning kernelbased embeddings in graph neural networks. In ECAI 2020, pages 1387– 1394. IOS Press, 2020.

- [19] Muhan Zhang, Pan Li, Yinglong Xia, Kai Wang, and Long Jin. Labeling Trick: A Theory of Using Graph Neural Networks for Multi-Node Representation Learning. In Advances in Neural Information Processing Systems, volume 34, pages 9061–9073. Curran Associates, Inc., 2021.
- [20] Luis Müller, Mikhail Galkin, Christopher Morris, and Ladislav Rampášek. Attending to Graph Transformers, February 2023.
- [21] Andrea Cini, Daniele Zambon, and Cesare Alippi. Sparse graph learning for spatiotemporal time series, 2022.
- [22] Alessio Micheli and Domenico Tortorella. Discrete-time dynamic graph echo state networks. *Neurocomputing*, 496:85–95, 2022.
- [23] Ming Jin, Huan Yee Koh, Qingsong Wen, Daniele Zambon, Cesare Alippi, Geoffrey I. Webb, Irwin King, and Shirui Pan. A Survey on Graph Neural Networks for Time Series: Forecasting, Classification, Imputation, and Anomaly Detection, July 2023.
- [24] Alessio Gravina and Davide Bacciu. Deep learning for dynamic graphs: Models and benchmarks, July 2023.
- [25] Xiang Zhang, Marko Zeman, Theodoros Tsiligkaridis, and Marinka Zitnik. Graph-Guided Network for Irregularly Sampled Multivariate Time Series. In International Conference on Learning Representations, October 2021.
- [26] Andrea Cini, Ivan Marisca, Filippo Maria Bianchi, and Cesare Alippi. Scalable Spatiotemporal Graph Neural Networks, September 2022.
- [27] Ivan Marisca, Andrea Cini, and Cesare Alippi. Learning to Reconstruct Missing Data from Spatiotemporal Graphs with Sparse Observations. Advances in Neural Information Processing Systems, 35:32069–32082, December 2022.
- [28] Jakob Gawlikowski, Cedrique Rovile Njieutcheu Tassi, Mohsin Ali, Jongseok Lee, Matthias Humt, Jianxiang Feng, Anna Kruspe, Rudolph Triebel, Peter Jung, Ribana Roscher, et al. A survey of uncertainty in deep neural networks. arXiv preprint arXiv:2107.03342, 2021.
- [29] Luca Franceschi, Mathias Niepert, Massimiliano Pontil, and Xiao He. Learning discrete structures for graph neural networks. In *ICML*, pages 1972–1982. PMLR, 2019.
- [30] Daniele Zambon, Andrea Cini, Lorenzo Livi, and Cesare Alippi. Graph state-space models, January 2023.
- [31] George Em Karniadakis, Ioannis G Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, 2021.

- [32] P. Gainza, F. Sverrisson, F. Monti, E. Rodolà, D. Boscaini, M. M. Bronstein, and B. E. Correia. Deciphering interaction fingerprints from protein molecular surfaces using geometric deep learning. *Nature Methods*, 17(2):184–192, February 2020.
- [33] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, and et al. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589, August 2021.
- [34] Davide Bacciu, Federico Errica, Alessio Gravina, Lorenzo Madeddu, Marco Podda, and Giovanni Stilo. Deep graph networks for drug repurposing with multi-protein targets. *IEEE Transactions on Emerging Topics in Computing*, 2023.
- [35] Francesco Landolfi, Davide Bacciu, and Danilo Numeroso. A tropical view of graph neural networks. In ESANN, 2023.
- [36] Danilo Numeroso, Davide Bacciu, and Petar Veličković. Dual algorithmic reasoning. In The Eleventh International Conference on Learning Representations (ICLR 2022), 2022.
- [37] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *ICLR*, 2019.
- [38] Weiwei Wang, Stefano Bromuri, and Michel Dumontier. Graph based categorical embedding. In *ESANN*, 2023.
- [39] Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, Will Hamilton, and Jure Leskovec. Hierarchical graph representation learning with differentiable pooling. In *NeurIPS*, pages 4800–4810, 2018.
- [40] Federico Errica, Alessio Gravina, Davide Bacciu, and Alessio Micheli. Hidden markov models for temporal graph representation learning. In ESANN, 2023.
- [41] Domenico Tortorella and Alessio Micheli. Richness of node embeddings in graph echo state networks. In ESANN, 2023.
- [42] Luis Torres, Bernardete Ribeiro, and Joel Arrais. Convolutional transformer via graph embeddings for few-shot toxicity and side effect prediction. In ESANN, 2023.
- [43] Thanh Vu, Huy Ngo, Bac Le, and Thanh Le. Fourier: Link prediction by mixing tokens with fourier-enhanced metaformer. In ESANN, 2023.
- [44] Nicolò Navarin, Luca Pasa, Luca Oneto, and Alessandro Sperduti. An empirical study of over-parameterized neural models based on graph random features. In ESANN, 2023.