A Roadmap from Weights to Wisdom: Inspecting and Extracting Knowledge from Graph Neural Networks

Artem Chernobrovkin[1]

[1]Gran Sasso Science Institute (GSSI), Viale Francesco Crispi, 7, 67100 L'Aquila, Italy

Abstract

Graph Neural Networks (GNNs) and Graph Convolutional Networks (GCNs) are powerful Artificial Intelligence (AI) models designed to process graph-structured data efficiently. However, their decision-making processes are often difficult to interpret, functioning as "black boxes". This research project aims to enhance the inspectability and learning capabilities of GNNs and GCNs by integrating symbolic AI while improving the representation of the knowledge they learn.

The research proposal focuses on integrating logic with Neural Networks to achieve two key objectives: enhancing inspectability (RQ1) and facilitating the acquisition and representation of symbolic knowledge from sub-symbolic data (RQ2).

Keywords

Neurosymbolic AI, Logic Integration, Graph Neural Networks, Graph Convolutional Neural Networks, Symbolic Reasoning Systems

1. Introduction

Graph data best represents complex relational patterns in biological systems, knowledge graphs, and social networks. Graph Neural Networks (GNNs) and Graph Convolutional Networks (GCNs) process data and find correlations and patterns efficiently. On the other hand, modal and formal ontology languages are evaluated using graph-structured models or Kripke models.

Deep learning models and Neural Networks have enhanced Artificial Intelligence (AI) by efficiently processing vast amounts of unstructured data. However, they frequently operate as "black-box" models due to limited inspectability and lack of inherent reasoning capabilities [1].

In contrast, formal logic-based symbolic reasoning systems explicitly represent knowledge and can deduce new information, making them well-suited for querying structured data and performing rule-based reasoning. In their interpretability, these systems are also closer to natural language, allowing for more transparent and intuitive explanations. However, certain concepts cannot be easily represented using classical crisp axiomatic methods [2]. Misclassification often occurs when an object lacks non-essential features or includes additional benign features. For example, if we define a car as having a front bumper, a car without one would no longer be classified as such. Similarly, defining humans as having five-fingered hands would exclude a person with polydactyly.

The thesis proposal aims to use the power of neurosymbolic AI [3], especially by exploiting the formal connections between logical systems and neural networks. Neurosymbolic AI aims to integrate NNs' learning capabilities with symbolic AI's interpretability and reasoning strengths. On the one hand, logical systems can improve the inspectability of neural networks that process graph data. This is the basis for the first of our research questions presented in Section 3. On the other hand, models learnt from data can be adequately integrated as concepts into domain ontologies. This inspired our second research question. We present some related literature in Section 2.

3rd Workshop on Bias, Ethical AI, Explainability and the Role of Logic and Logic Programming (BEWARE24), co-located with AIxIA 2024, November 25-28, 2024, Bolzano, Italy

artem.chernobrovkin@gssi.it (A. Chernobrovkin)

0009-0001-9786-394X (A. Chernobrovkin)

2. Background and State of the Art

In this section, we want to present the literature search that was done before. Artificial Neural Networks (ANNs) are a class of machine learning models that have become competitive alternatives to conventional regression and statistical models [4]. The architecture of ANNs consists of multiple layers of interconnected neurons (or nodes), where each neuron processes input data and applies an activation function to produce an output [5]. However, specialized neural network models are required to handle graph structures effectively when knowledge is represented as graphs. GNNs and GCNs extend traditional ANNs to process graph-based data, capturing the complex relationships inherent in graphs. GNNs are deep learning models designed for various tasks on graphs, primarily classification tasks that can be categorized at three levels: node, edge, and graph. At the node level, tasks include node classification, where nodes are categorized, and node regression, where numerical values are predicted for each node. Edge-level tasks, such as edge classification and link prediction, involve determining the types of edges or predicting the existence of edges between node pairs. At the graph level, tasks like graph classification, regression, and matching require the model to encode and process entire graphs. A more detailed overview is available in [1].

Several classes of GNNs have been developed, including Aggregate-Combine GNNs (AC-GNNs) [6, 7], Aggregate-Combine-Readout GNNs (ACR-GNNs) [6, 7], and Graph Convolutional Networks (GCNs) [8]. GCNs extend traditional NNs to graph data by defining convolution operations over graphs. GCNs iteratively update node feature representations by aggregating information from nearby nodes, effectively capturing node properties and the local graph structure. By using spectral filters to generalise the idea of convolutions, this method enables GCNs to analyse multi-layered information and efficiently identify local patterns in the graph [9]. AC-GNNs also operate by iteratively refining node features through multiple layers. Each layer learns useful representations that enable downstream classification tasks by aggregating information from a node's neighbours and combining it with its current representation [6]. The choice of GNN variant depends on the specific task at hand. GCNs are particularly effective for tasks where local neighbourhood information is crucial and can benefit from convolutional operations. However, alternative GNN architectures may be more suitable for processing sequential or dynamic graph data or when more complex aggregation methods or attention mechanisms are required.

In the literature, various types of logic have been explored in combination with Neural Networks, particularly with GNNs and GCNs. These logics include Description Logic $\mathcal{ALCQ}$, Graded Modal Logic (GML), First-Order Logic with Counting (FOC) [10] and its two-variable fragment FOC_2 [6], and $K^{\#}$ [11]. Some notable connections between the relative expressibility of logics and classes of GNNs are represented in Figure 1.

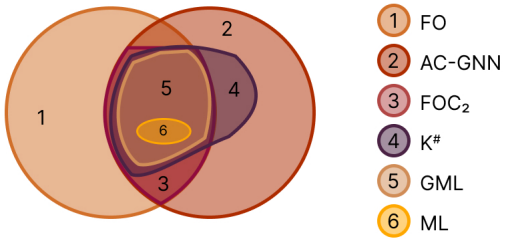

Figure 1: Relative expressivity of GNNs and logics

An Euler diagram summarizing the relative expressivity of logic- and neural network-based frameworks is shown in Figure 1.

In the articles [6, 7], authors use the guarded fragment of FOC_2 that corresponds to Graded Modal Logic (GML), or, equivalently, to the Description Logic $\mathcal{ALCQ}$. The result they provided is the following: any formula of GML can be transformed into an equivalent AC-GNN, and every AC-GNN expressible in first-order logic has an equivalent formula in GML. In article [11], authors show that the logic $K^{\#}$ captures AC-GNNs. They provided the following results: tfor any formula φ of $K^{\#}$, there exists GNN

$\mathcal{A}$ recognizing the same pointed graphs as formula φ. More essential results illustrated that it is possible to have a translation (tr) from AC-GNNs to $K^{\#}$, which can be done efficiently.

Description Logic (DL) [12] is a family of logics designed for the formal and structured representation of domain knowledge, commonly used in ontologies. $\mathcal{ALCQ}$ is an extension of the foundational $\mathcal{ALC}$ (Attributive Language with Complement) Description Logic by incorporating Qualified Number Restrictions (Q). This improvement allows for reasoning about complicated domains where counts and numbers are crucial and more expressive knowledge representation. For example, we can have the concept of "red node with at most two black neighbours and at least one not black" (❀). In $\mathcal{ALCQ}$ we obtain the following formula: Red $\sqcap$ ($\leq$ 2 hasNeighbour.Black) $\sqcap$ ($\geq$ 1 hasNeighbour.$\neg$Black)

Graded Modal Logic (GML) [13] extends modal logic by allowing quantification over the number of accessible worlds in which a proposition holds. Instead of reasoning solely about whether something is necessarily or possibly true, GML introduces numeric constraints, such as "in at least n" or "in at most n" accessible worlds, making it useful in contexts that involve reasoning over quantities. For example, the expression ❀ is represented in GML as: Red $\wedge$ $\Diamond_{\leq 2}$Black $\wedge$ $\Diamond_{\geq 1}\neg$Black

FOC_2 is the two-variable fragment of First-Order Logic (FO) extended with counting quantifiers of the form $\exists^{\geq N} x\, \varphi(x)$, which specify that at least N nodes satisfy the formula φ. FOC_2 restricts formulas to use only two variables, typically denoted as x and y, though these variables can be reused. In [6], FOC_2 refers to this restricted fragment of first-order logic with counting quantifiers. Counting quantifiers can also be expressed using standard existential quantifiers combined with inequality conditions [14]. For example, we have the expression ❀ represented in FOC_2 by the following formula: $\text{Red}(x) \wedge (\exists^{\leq 2} y.(\text{Neighbour}(x, y) \wedge \text{Black}(y))) \wedge (\exists^{\geq 1} y.(\text{Neighbour}(x, y) \wedge \neg\text{Black}(y)))$

$K^{\#}$ [11] extends propositional logic with numeric constraints of the form $\xi \geq 0$, where ξ can be $\#\varphi$ (denoting the number of successors in a graph satisfying φ), an integer, an addition, or a multiplication. $K^{\#}$ naturally extends modal logic because we can define the modality $\Box\varphi := (\#(\neg\varphi) \leq 0)$. We have the expression ❀ that is represented in $K^{\#}$ by the following formula: Red$\wedge$(#Black ≤ 2)$\wedge$(#$\neg$Black ≥ 1). Beyond FOL expressivity, one can characterize "nodes with at least twice as many black neighbours as red neighbours" as $\#\text{Black} \geq 2 \cdot \#\text{Red}$.

Concept Learning (CL) is a field that aims to replicate human abilities in learning concepts from different types of data. In the context of Neurosymbolic AI, CL can enable systems to learn high-level representations from raw data and formalize them into symbolic concepts. We did some literature searches in this field and found articles highlighting the CL's key points. One of the frameworks we found is DL-Learner introduced in the article [15], where "DL" stands for Description Logic. The DL-Learner addresses several relevant learning problems, all unified by the reliance on background knowledge presented as an ontology. This framework focuses more on ML techniques, such as inductive learning (which can deal with well-structured data), using the ontologies represented in OWL.

The CL has to deal with data that can be presented in different types and ways and is not perfectly structured. Deep learning models, such as neural networks, can be used. One advantage of this model is its ability to learn from unstructured or noisy datasets at scale effectively. By integrating Neural Networks with symbolic reasoning systems, we can extract complex patterns and concepts from raw data and formalize them into set rules within an ontology [16]. This approach bridges the gap between unstructured data and symbolic representations, allowing the dynamic enrichment of ontologies with concepts learned directly from data. The illustration of the importance of concept learning can be presented and mentioned in several articles that highlight the application part of concept learning. One can be presented in the article [17], where they provided information about the neuro-symbolic concept learner that used the data as pictures, words and semantic parsing of sentences.

A contribution to CL was made in the article [18]. The paper studies the effect of adding weighted threshold connectives to Description Logic. It is shown that they do not increase the complexity of reasoning. The authors also show that concepts using the new connectives can be learnt effectively from data. This allows complex concepts to be seamlessly incorporated into existing ontologies.

Interpretability and explainability are crucial in developing models [19, 20]. Interpretability refers to understanding a model's inner workings—how inputs are transformed into outputs and the relationships within the data. In contrast, explainability focuses on the model's ability to justify specific decisions, offering clear and comprehensible explanations to a human audience.

3. Research Problem

An important area of research in neurosymbolic Artificial Intelligence is the integration of logical reasoning capabilities with graph-based neural network architectures, such as Graph Neural Networks (GNNs) and Graph Convolutional Networks (GCNs). This synthesis combines the robust pattern recognition and learning abilities of GNNs and GCNs with the structured reasoning strengths of logic systems.

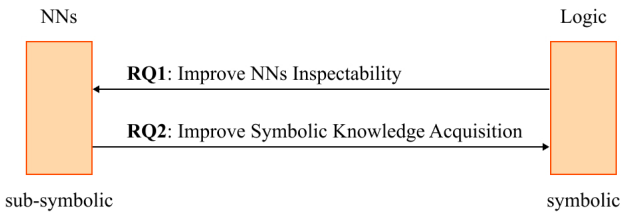

Figure 2: Research Questions

Figure 2 provides an overview of the relationship between Neural Networks and logic, highlighting the research questions (RQs) aimed at improving neural network inspectability (RQ1) and facilitating the acquisition of symbolic knowledge from sub-symbolic data representations (RQ2).

RQ1: *How translation from Logic to Neural Networks can help the inspectability?*

Inspectability is a combination of interpretability and explainability. Inspectability seeks to improve transparency in Neural Network (NN) outputs. Symbolic rules are logical statements or constraints that specify properties or relationships inside the domain. They improve interpretability by explaining how Neural Networks' predictions match the organized GCNs.

Following decision-making, high-dimensional feature vectors are mapped back to interpretable symbolic words using post-hoc symbolic logic translations, which correlate to ideas or principles in a formal logic or ontology. This allows for the analysis of NN outputs. Symbolic rules are logical statements or constraints that specify properties or relationships inside the domain. They improve interpretability by precisely explaining how the predictions made by the Neural Networks match the organised domain knowledge.

Requiring models to conform to set rules enables people to comprehend the reasoning behind decisions by cross-validating choices against logical requirements. For enhancing explainability, decisions from GNNs or GCNs can be examined against logical formulas from systems like $\mathcal{ALCQ}$, $K^{\#}$, or Graded Modal Logic, enabling symbolic explanations. Post hoc explanations generate symbolic explanations after a Neural Network decision, trace back through logical constraints to identify specific rules that influenced the decision, and explain the cases leading to a classification.

The process of analysing a Neural Network ($\mathcal{A}$) is shown in Figure 3, where the structure of the network is compared to predetermined logical constraints. The set of pointed models, $[[\mathcal{A}]]$, for which the network yields a value of 1, is examined in the analysis to see if it is with the logical formula φ as a consequence. Inspection can also be about checking other kinds of properties, such as the consistency of the GNN $\mathcal{A}$ with a logical formula φ: $[[\mathcal{A}]] \cap [[\varphi]] \neq \emptyset$. This comparison provides a more structured and comprehensible assessment of the network's behaviour by guaranteeing that the network's decision-making follows predicted logical patterns and conforms to symbolic norms.

This approach is inspired by the work in [11], particularly the decision problem asking whether $[[\mathcal{A}]] \subseteq [[\varphi]]$, that is, whether the set of pointed models for which the GNN $\mathcal{A}$ has an output 1 is a subset

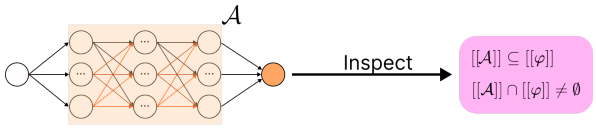

Figure 3: RQ1. An overview of the inspection process comparing the Neural Network's behaviour to logical constraints, ensuring the network's decision-making adheres to predefined symbolic norms.

of the set of pointed models where the $K^{\#}$ formula φ is satisfied. As said in Section 2, the authors show that for each GNN $\mathcal{A}$, there is a formula $tr(\mathcal{A})$ recognizing the same pointed graph. It extends a result of [6] about GML and is generalised by [21]. The idea of how $tr(\mathcal{A})$ works is the following: each GNN layer is mapped to logical conditions that describe how node features are updated. In contrast, the aggregation of neighbour features is translated into counting modalities. The GNN's final classification is expressed as a logical condition derived from these formulas, allowing its decisions to be examined against pre-established logical rules, thereby enhancing transparency and inspectability. We aim to obtain similar results for GCNs and other logics.

We can provide an example of Inspecting a Neural Network using ontological concepts. These examples illustrate how a Neural Network, trained on a dataset about dogs, can be inspected by leveraging concepts from an ontology about animals. Here, we translate the trained NN, $\mathcal{A}^{\text{Dog}}$, into logical expressions to verify its alignment with specific ontological properties of dogs. For instance: $\left[\left[\mathcal{A}^{\text{Dog}}\right]\right] \subseteq [[\text{Animal}]]$: this condition ensures that the NN's classify things that are "Animal". $\left[\left[\mathcal{A}^{\text{Dog}}\right]\right] \subseteq [[\exists \text{hasBodyPart.Head}]]$: this checks that all the things that the NN classified "have a head". $\left[\left[\mathcal{A}^{\text{Dog}}\right]\right] \cap [[\neg\exists \text{hasBodyPart.Tail}]] \neq \emptyset$: this is true that a thing that NN classified doesn't "have a tail". $\left[\left[\mathcal{A}^{\text{Dog}}\right]\right] \cap [[\leq 3.\text{hasBodyPart.Leg}]] \neq \emptyset$: this is true that a thing that NN classified "has less or equal than three legs". By mapping NN outputs to ontological rules, this approach facilitates inspection and validation of the NN's behaviour against known domain concepts, thus enhancing transparency and inspectability.

RQ2: *Can Neural Networks improve symbolic knowledge representation by incorporating insights from sub-symbolic data?*

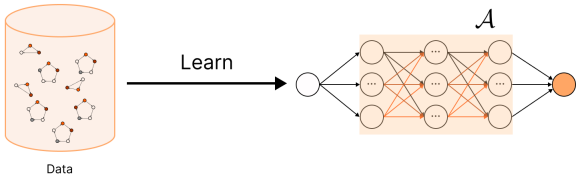

Figure 4: Training a GNN model on graph-structured data

The challenge of this research question lies in the difficulty traditional ontological frameworks face when trying to accurately capture specific complex, domain-specific concepts using only TBox axioms. The suggested method bridges the gap between symbolic and sub-symbolic representations by employing Neural Networks to directly learn these complicated concepts (C) from data. The Neural Networks are trained to learn such concepts (Figure 4), which can be integrated into the ontology as logical definitions ($C \equiv tr(\mathcal{A})$), enhancing the expressivity and adaptability of the ontology. Additionally, patterns found in data can be translated into new symbolic relationships or restrictions by ANNs, opening the door to creating a more adaptable logical framework that considers empirical findings.

Figure 5 illustrates how, after learning, a GNN (denoted as $\mathcal{A}$) develops a classification for a concept C, which is then translated (tr) into symbolic form. The axiom $C \equiv tr(\mathcal{A})$ is then added to a domain ontology $Ontology$ to enrich it with the new concept.

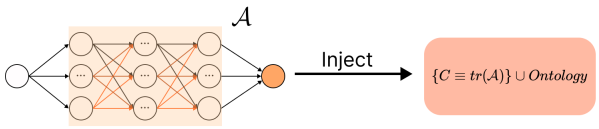

Figure 5: RQ2. Injection of a GNN ($\mathcal{A}$) classification (C) into an ontology, with translation (tr) into symbolic form.

We can illustrate this with an example. Suppose that we have a Neural Network $\mathcal{A}^{\text{Dog}}$ trained on the dataset "Dogs" for input. We can obtain a concept $tr(\mathcal{A}^{\text{Dog}})$ that represents the classification of $\mathcal{A}^{\text{Dog}}$, that is, supposedly the concept of a "dog". Suppose we also have an ontology $Ontology$ of animals. It may contain the concept Dog, and maybe axioms such as Dog $\sqsubseteq$ Canid. According to the strategy, we would add to $Ontology$ the axiom Dog $\equiv tr(\mathcal{A}^{\text{Dog}})$. In doing so, we obtain a hybrid ontology that describes expert knowledge about animals and contains the concept of a "dog" learnt from actual data.

The results of these research questions will be validated theoretically and empirically, with the models tested in real-world contexts to confirm their ability to handle large, complex, and heterogeneous datasets effectively.

References

[1] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, M. Sun, Graph neural networks: A review of methods and applications, AI Open 1 (2020) 57–81. doi:10.1016/j.aiopen.2021.01.001.

[2] E. Margolis, S. Laurence, Concepts, in: E. N. Zalta, U. Nodelman (Eds.), The Stanford Encyclopedia of Philosophy, Fall 2023 ed., Metaphysics Research Lab, Stanford University, 2023.

[3] A. Sheth, K. Roy, M. Gaur, Neurosymbolic artificial intelligence (why, what, and how), IEEE Intelligent Systems 38 (2023) 56–62. doi:10.1109/MIS.2023.3268724.

[4] V. S. Dave, K. Dutta, Neural network based models for software effort estimation: a review, Artificial Intelligence Review 42 (2014) 295–307. doi:10.1007/s10462-012-9339-x.

[5] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. Mohamed, H. Arshad, State-of-the-art in artificial neural network applications: A survey, Heliyon 4 (2018). doi:10.1016/j.heliyon.2018.e00938.

[6] P. Barceló, E. V. Kostylev, M. Monet, J. Pérez, J. L. Reutter, J.-P. Silva, The expressive power of graph neural networks as a query language, ACM SIGMOD Record 49 (2020) 6–17. doi:10.1145/3442322.3442324.

[7] M. Grohe, The logic of graph neural networks, in: 2021 36th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), IEEE, 2021, pp. 1–17. doi:10.1109/LICS52264.2021.9470677.

[8] L. C. Lamb, A. d. Garcez, M. Gori, M. O. Prates, P. H. Avelar, M. Y. Vardi, Graph neural networks meet neural-symbolic computing: A survey and perspective, in: C. Bessiere (Ed.), Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI, International Joint Conferences on Artificial Intelligence Organization, 2020, pp. 4877–4884. doi:10.24963/ijcai.2020/679, survey track.

[9] T. N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, IEEE Transactions on Neural Networks and Learning Systems (2017) 1–14.

[10] M. Grohe, The descriptive complexity of graph neural networks, in: 38th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), 2023, pp. 1–14. doi:10.1109/LICS56636.2023.10175735.

[11] P. Nunn, M. Sälzer, F. Schwarzentruber, N. Troquard, A logic for reasoning about aggregate-combine graph neural networks, in: K. Larson (Ed.), Proceedings of the Thirty-Third International

Joint Conference on Artificial Intelligence, IJCAI-24, International Joint Conferences on Artificial Intelligence Organization, 2024, pp. 3532–3540. doi:10.24963/ijcai.2024/391, main Track.

[12] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, P. Patel-Schneider, The Description Logic Handbook: Theory, Implementation and Applications, 2 ed., Cambridge University Press, 2007. doi:10.1017/CBO9780511711787.

[13] M. de Rijke, A note on graded modal logic, Studia Logica: An International Journal for Symbolic Logic 64 (2000) 271–283.

[14] E. Gradel, M. Otto, E. Rosen, Two-variable logic with counting is decidable, in: Proceedings of Twelfth Annual IEEE Symposium on Logic in Computer Science, 1997, pp. 306–317. doi:10.1109/LICS.1997.614957.

[15] L. Bühmann, J. Lehmann, P. Westphal, DL-Learner—A framework for inductive learning on the semantic web, Journal of Web Semantics 39 (2016) 15–24. doi:10.1016/j.websem.2016.06.001.

[16] A. S. Garcez, L. C. Lamb, D. M. Gabbay, Neural-Symbolic Cognitive Reasoning, 1 ed., Springer Berlin, Heidelberg, 2008. doi:10.1007/978-3-540-73246-4.

[17] J. Mao, C. Gan, P. Kohli, J. B. Tenenbaum, J. Wu, The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision, arXiv preprint arXiv:1904.12584 (2019).

[18] P. Galliani, G. Righetti, O. Kutz, D. Porello, N. Troquard, Perceptron connectives in knowledge representation, in: Knowledge Engineering and Knowledge Management, Springer International Publishing, Cham, 2020, pp. 183–193. doi:10.1007/978-3-030-61244-3_13.

[19] V. Shah, S. R. Konda, Neural networks and explainable ai: Bridging the gap between models and interpretability, International Journal of Computer Science and Technology 5 (2021) 163–176.

[20] P. Linardatos, V. Papastefanopoulos, S. Kotsiantis, Explainable ai: A review of machine learning interpretability methods, Entropy 23 (2021). doi:10.3390/e23010018.

[21] M. Benedikt, C.-H. Lu, B. Motik, T. Tan, Decidability of graph neural networks via logical characterizations, in: 51st International Colloquium on Automata, Languages, and Programming, volume 297 of *Leibniz International Proceedings in Informatics*, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024, pp. 127:1–127:20. doi:10.4230/LIPIcs.ICALP.2024.127.