EDGE: Evaluation Framework for Logical vs. Subgraph Explanations for Node Classifiers on Knowledge Graphs

Rupesh Sapkota
rupezzz@mail.uni-paderborn.de
Paderborn University
Paderborn, Germany

Dominik Köhler
dominik.koehler@uni-paderborn.de
Paderborn University
Paderborn, Germany

Stefan Heindorf
heindorf@uni-paderborn.de
Paderborn University
Paderborn, Germany

Abstract

As machine learning and deep learning become increasingly integrated into our daily lives, understanding how these technologies make decisions is crucial. To ensure transparency, accountability, and ethical adherence, these so-called "black-box" models should be accompanied by human-comprehensible explanations of their predictions. This clarity is essential for establishing trust in their real-world applications. Similarly, it is crucial to compare different types of explanations to evaluate and understand their effectiveness, interpretability, and generalization capabilities for informed selection in various applications. To this end, we propose a framework called EDGE to evaluate diverse knowledge graph explanations, assessing logical rule-based and subgraph-based explanations by various explainers in terms of prediction accuracy and fidelity to the Graph Neural Network (GNN) model. Our evaluations reveal that logical methods excel in explaining complex and structured data, while subgraph-based models exhibit higher fidelity to the GNN model, earning them the label "GNN Explainers". Although further diversified evaluations are necessary to determine the superiority of one explanation type over another, our study shows that each type has pros and cons.

CCS Concepts

• **Computing methodologies** → *Supervised learning by classification*; *Inductive logic learning*; *Rule learning*; *Semantic networks.*

Keywords

Node Classification, Explainable AI, Knowledge Graphs

ACM Reference Format:
Rupesh Sapkota, Dominik Köhler, and Stefan Heindorf. 2024. EDGE: Evaluation Framework for Logical vs. Subgraph Explanations for Node Classifiers on Knowledge Graphs. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management (CIKM '24), October 21–25, 2024, Boise, Idaho, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3627673.3679904

1 Introduction

With advanced deep learning models and computational capabilities, machine learning and artificial intelligence gained momentum over the past decade and have achieved remarkable performance in areas such as natural language processing [38], image recognition [36], and graph classification [19]. However, the lack of transparency in their decision-making processes has resulted in them being labeled as "black box" models. These models need to be accompanied by human-understandable explanations of their predictions to ensure transparency, accountability, and ethical considerations to build trust for their application in real-world scenarios.

Graph neural networks (GNNs) have also gained significant attention in recent years for their ability to model complex graph structures. While they have achieved state-of-the-art results in tasks such as node classification [27], link prediction [47], and graph classification [48], they again fall into the categories of "black box" models, due to the lack of transparency in their decision-making process. To provide explainability to GNNs, various explainers such as GNNExplainer [43], SubgraphX [46], PGExplainer [28] and XGNN [44] have been proposed in the recent literature.

While a few works [1, 2, 45] have proposed explanation methods for GNNs, except for Himmelhuber et al. [16], they purely focus on subgraph explanations and neglect other explanation formats such as logical expressions, whereas in this paper, we present a framework explaining GNNs in terms of description logics. Towards this end, we employ concept learners [15, 25] that were originally developed to learn concepts in description logics from positive and negative examples in OWL knowledge bases. To compare their explanations to traditional subgraph explanations, we propose the novel evaluation framework EDGE,[1] "Evaluation of Diverse Knowledge Graph Explanations" that allows to compare and evaluate the performance of subgraph and logical explanations for node classification in knowledge graphs. As a proof-of-concept, we compare two subgraph-based explainers, SubgraphX [46] and PGExplainer [28] and two logical explainers, EvoLearner [15] and CELOE [25]. The framework evaluates the performance of explainers on two main criteria: prediction performance and explanation performance. Prediction performance evaluates how well an explainer's predictions align with ground truth labels, while explanation performance assesses the consistency between the explainer's predictions and those generated by the graph neural network. The framework also uses three real-world heterogeneous graph datasets and two state-of-the-art GNN models to perform explanations.

To summarize our contributions:

- In this paper, we explain the predictions of GNNs in terms of description logics.
- We introduce and discuss the evaluation framework EDGE, to evaluate the performance of diverse GNN explainers.
- We compare subgraph explanations to logical explanations of GNNs.

[1] https://github.com/ds-jrg/EDGE

CIKM '24, October 21–25, 2024, Boise, Idaho, USA

ACM ISBN 979-8-4007-0436-9/24/10
https://doi.org/10.1145/3627673.3679904

2 Related Work

Yuan et al. [45] provide a taxonomy of GNN explainers based on their objectives and methodologies, categorizing them into instance-level and model-level explanations. Regarding the former, gradient or feature-based methods exploit gradient values or hidden features to assess input importance. For instance, SA [49] uses squared gradient values, while GRAD-CAM [34] offers 'visual explanations' based on feature importance. Perturbation-based methods like GNNExplainer [43] and SubgraphX [46] provide explanations by examining output variations resulting from input changes. Surrogate models, such as GraphLIME [17] and PGM-Explainer [39], use interpretable models to mimic the predictions of deep graph networks. Model-level explanations, such as XGNN [44], generate input-independent subgraphs. All of these approaches produce subgraphs as an explanation. To the best of our knowledge, previous work has not compared subgraph and logical explanations.

Agarwal et al. [1] introduce an evaluation framework for subgraph-based GNN explainers, called GraphXAI. The framework includes eight state-of-the-art GNN explainers: the gradient-based methods Grad [35], GradCAM [31], Integrated Gradients [37], and GuidedBP [4], the surrogate-based model PGMExplainer [39], and the subgraph-based methods SubgraphX [46], GNNExplainer [43], and PGExplainer [28]. The explainers are evaluated using four metrics: (i) Graph Explanation Accuracy (GEA), (ii) Graph Explanation Faithfulness (GEF), (iii) Graph Explanation Stability (GES), and (iv) Graph Explanation Fairness (GECF, GEGF) [1]. The experiments are performed using GIN and GCN models on both real-world and synthetic datasets. Our work employs real-world datasets without ground truth explanations, making it infeasible to calculate GEA. Whereas GEF metrics evaluate explanation performance using continuous scores, logical explainers produce discrete predictions. Therefore, we assess explanation performance using accuracy, precision, recall, and F1 score. As the GES and GEGF metrics rely on graph masks, which are not available in the case of logical explanations, we omit them. Similarly, Amara et al. [2] propose GraphFramEx, a framework for evaluating posthoc, subgraph-based GNN explainability techniques, akin to GraphXAI [1]. The key difference between GraphXAI and GraphFramEx lies in their evaluation metrics. GraphFramEx uses phenomenon fidelity to compare the explanation predictions with the ground truth, and model-focus fidelity to compare the explanation predictions with the GNN predictions, corresponding to our prediction performance and explanation performance, respectively. Neither Agarwal et al. [1] nor Amara et al. [2] include a single logical explanation method in their surveys.

In contrast to graph-based posthoc explainers, logic-based methods tackle GNN interpretability from a different perspective as they aim to learn the underlying concepts or rules governing the GNN's predictions as a global surrogate model. This leads to explanations that are rooted in human-understandable concepts. For example, EvoLearner [15] is a logic-based evolutionary method for learning concepts in description logics from sets of positive and negative examples. Similarly, Class Expression Learning for Ontology Engineering (CELOE) [25] learns logical concepts via inductive logic programming and refinement operators. Several subsequent works have extended these methods. DRILL [10] guides the refinement operator via reinforcement learning. CLIP [20] accelerates concept learning by predicting the lengths of concepts, OntoSample [3] by sampling the knowledge base, and AutoCL [26] by feature selection. NCES [22], NCES2 [21], ROCES [23] synthesize class expressions directly by "translating" sets of examples to class expressions akin to machine translation. Himmelhuber et al. [16] combine symbolic and sub-symbolic approaches for explaining GNNs (logical and subgraph approaches, respectively). They use the node features and edge masks from a sub-symbolic approach to enrich the knowledge base and provide the updated knowledge bases to a symbolic approach to generate class expressions. Compared to our paper, they focus on graph classification, use only one dataset, and do not compare different explainers.

Rule learners for *link prediction* include AMIE [11, 12, 24], DRUM [32], AnyBURL [29, 30], DeepPath [42], and MINERVA [8]. Cucala et al. [7] developed an explainable GNN-Based Model for link prediction. In contrast, we focus on explanations for *node classifiers.*

3 Evaluation Framework

The evaluation framework proposed in this work is called EDGE [2], "Evaluation of Diverse knowledge Graph Explanations". EDGE comprises two state-of-the-art Graph Neural Networks, two subgraph-based GNN explainers, two logical explainers, three datasets, and eight evaluation metrics. The primary goal of the framework is to compare subgraph explainers with logical explainers quantitatively and automatically on real-world datasets.

As models to explain, we have selected RGCN (Relational Graph Convolution Network) [33] and RGAT (Relational Graph Attention Network) [6], due to their ability to capture the relational structure in the data. We base the results and evaluations in this paper on the performance of the explainers on the RGCN model and the results for the RGAT model can be found in the GitHub code repository.

3.1 Explainers

Subgraph-based Explainers. We selected PGExplainer [28] and SubgraphX [46] as subgraph-based explainers for our framework. PGExplainer [28] trains a parameterized explainer network to generate explanations. SubgraphX searches explanations by maximizing the GNN-Score [44]. As the original implementation of SubgraphX in DGL only supports graph classification, we adapt it for node classification: When computing the marginal contribution for a specific subgraph, instead of maximizing the prediction for the graph class, we maximize the prediction for the category of the explained node. GraphXAI [1] has adapted SubgraphX for node classification, too, but only supports homogeneous graphs, whereas we support heterogeneous graphs.

Logic-based Explainers. We incorporate the logic-based explainers EvoLearner [15] and CELOE [25] from the OntoLearn [13] framework. These explainers utilize positive and negative examples to learn a class expression in description logics. We obtain the positive and negative examples from the GNN predictions. Subsequently, these learned class expressions serve as explanations and are employed to classify instances as either positive or negative.

[2] https://github.com/ds-jrg/EDGE

Table 1: Dataset statistics.

Dataset	Nodes	Edges	Attrib.	Target Cat.	Clsf. Prop.
AIFB	2,548	15,700	8,705	Persons	Affiliations
Mutag	22,372	40,666	10,845	Molecules	Mutagenicity
BGS	101,451	276,173	102,988	Unit of rock	Type of rock

3.2 Datasets

For comparison, we utilize the datasets AIFB, MUTAG, and BGS as shown in Table 1.

AIFB. The AIFB dataset [5] describes the AIFB (Institute for Applied Informatics and Formal Description Methods) research institute, including its staff, research groups, and publications. The goal is to predict the affiliation of a person. We cast it as a binary classification task by considering the largest class as positive and the remaining classes as negative.

MUTAG. The MUTAG dataset [9] deals with molecules and their potential carcinogenicity. The task is to predict the "isMutagenic" property of molecules based on structural molecule data.

BGS. The BGS dataset is a relational dataset curated by the British Geological Survey and offers comprehensive geological measurements across Great Britain to forecast lithogenesis attributes of named rock units. It features approximately 150 such units with distinct lithogenesis properties and the DGL implementation of the dataset is designed to classify the two major types of rocks: Fluvial (FLUVI) and Glacial (GLACI).

Dataset Splits. All datasets come along with training and testing sets. For training the RGCN and RGAT models, we use 80% of the training set for training and 20% for early stopping. SubgraphX requires no training. PGExplainer requires the whole graph without ground-truth labels for training. Logical approaches are trained with the positive and negative examples obtained from the GNN predictions on the whole training set.

3.3 Metrics

The EDGE framework assesses explainers based on two main criteria: prediction performance and explanation performance. The prediction performance assesses the ability of the explainer to predict the original ground truth labels of data; the explanation performance assesses the ability of the explainer to make the same prediction as the GNN. Those comparisons are done in terms of accuracy, precision, recall, and F1-score. Our goal in evaluating prediction performance was to see how closely the explainers reflect the ground truth while maintaining fidelity to the GNN.

4 Evaluation

4.1 Evaluation Setup

The experiments ran on a system with an Intel Xeon Platinum 8462Y+ processor, 62 GB shared memory on Ubuntu 22.04.4 LTS within a Python 3.10 virtual environment. We used an NVIDIA H100 GPU with 40 GB memory to accelerate computations. For subgraph approaches, the datasets were converted into DGL graphs using the "RDFGraphDataset" class from the DGL library [40]. For the logical approaches, the RDF datasets were converted into OWL Knowledge Graphs using the ROBOT tool [18]. Both transformations were performed on the same original RDF datasets, ensuring conversion accuracy through a series of test cases within the framework.

Table 2: Performance of explainers on AIFB, Mutag, and BGS.

Approach	Pred. Perf.				Expl. Perf.			
	A	P	R	F1	A	P	R	F1
				AIFB Dataset				
CELOE	0.722	0.647	0.733	0.688	0.739	0.682	0.743	0.711
EvoLearner	0.650	0.546	**0.973**	0.699	0.667	0.567	**0.973**	0.716
PGExplainer	**0.861**	**0.797**	0.920	**0.849**	**0.900**	**0.851**	0.948	**0.894**
SubGraphX	0.800	0.746	0.853	0.784	0.839	0.797	0.882	0.829
				Mutag Dataset				
CELOE	0.682	0.689	0.947	0.798	0.682	0.686	0.950	0.794
EvoLearner	**0.718**	**0.710**	**0.969**	**0.819**	0.694	0.695	**0.956**	0.802
PGExplainer	0.603	0.703	0.707	0.698	**0.756**	0.811	0.825	**0.813**
SubGraphX	0.579	0.688	0.676	0.674	0.744	**0.812**	0.803	0.802
				BGS Dataset				
CELOE	0.503	0.405	**0.940**	0.566	0.490	0.400	0.927	0.554
EvoLearner	0.538	0.427	**0.940**	**0.584**	0.538	0.435	**0.947**	**0.586**
PGExplainer	**0.676**	**0.578**	0.540	0.536	**0.703**	**0.589**	0.573	0.566
SubGraphX	0.655	0.527	0.580	0.537	0.697	0.561	0.629	0.580

To evaluate our approach, we selected the Relational Graph Convolution Network (RGCN) model as the base GNN model throughout the experiments. We fine-tuned the GNN model parameters for each evaluation dataset to optimize the performance using multiple test runs and following OpenHGNN's RGCN implementation [14]. The model was trained on the AIFB, MUTAG, and BGS datasets for node classification tasks, as detailed in Section 3. The model was optimized with a learning rate of 0.005 and a weight decay of 0.0005 to prevent over-fitting. Our implemented RGCN model validation accuracy is 0.94 for the AIFB dataset, 0.72 for the MUTAG dataset, and 0.93 for the BGS dataset, which is almost identical to the original implementation of the RGCN model [33].

Due to the large BGS dataset, the PGExplainer model was trained for more epochs, with early stopping applied to all datasets to prevent overfitting. For SubgraphX, parameters were carefully adjusted to ensure adequate exploration while limiting time complexity.

4.2 Evaluation Results

Table 2 shows the prediction performance and explanation performance of the various explainers in terms of accuracy, precision, recall, and F1-measure on the AIFB, MUTAG, and BGS datasets averaged across 5 independent runs. On AIFB, the subgraph-explainers PGExplainer and SubgraphX emerge as the top performers, excelling in both prediction performance and explanation performance except for recall, where EvoLearner outperforms all approaches (0.973 for both prediction and explanation recall). On the

MUTAG dataset, the logical explainers CELOE and EvoLearner exhibit remarkably high predictive performance, with EvoLearner slightly surpassing CELOE across most metrics (e.g., the prediction accuracy of CELOE is 0.682 and of EvoLearner 0.718). Meanwhile, PGExplainer and SubgraphX demonstrate comparable predictive and explanation performance to each other. However, their predictive performance is considerably lower than the logical explainers and their explanation performance slightly higher. On the BGS dataset, the subgraph explainers lead in accuracy and precision whereas the logical explainers considerably lead in recall (e.g., the prediction recall of logical explainers is 0.940 whereas PGExplainer and SubgraphX exhibit a recall of only 0.540 and 0.580, respectively).

Comparing the results for the RGAT model (not shown in table, only on GitHub) with the RGCN model reveals additional insights. While the explainers generally demonstrate slightly lower predictive and explanation scores with RGAT compared to RGCN, some differences are worth noting. For instance, PGExplainer's prediction accuracy drops from 0.861 to 0.667, and its explanation F1-score decreases from 0.849 to 0.666 on AIFB. Similarly, SubgraphX sees a decline in prediction accuracy from 0.800 to 0.656 and explanation precision from 0.797 to 0.624. Logical approaches have slightly higher explanation performance on the BGS dataset with the RGAT model (except for recall).

Overall, we observe that subgraph approaches excel in terms of precision whereas logical approaches excel in terms of recall. We attribute this behavior to the expressiveness of subgraph and logical explainers. Whereas the subgraph approaches produce a single graph as an explanation that allows them to explain a few predictions with a high precision, the increased expressiveness of logical explainers (e.g., via disjunction and cardinality restrictions) allows them to explain many more predictions leading to much higher recall. With a few exceptions, the explanation performance of subgraph explainers in terms of explanation accuracy, also known as fidelity, is higher across datasets.

Interestingly, on MUTAG, the logical explainers yield explanations that are closer to the ground truth than to the GNN predictions, e.g., EvoLearner achieves a predictive F1 measure of 0.819 whereas only an explanatory F1 measure of 0.802. We attribute this to the structural characteristics of the MUTAG dataset, which is frequently used in logical approaches [41].

5 Discussion

Runtime. Logical approaches consistently take about 1 minute for explanations on AIFB and MUTAG but extend to around 6 minutes for BGS. Subgraph approaches, however, can take considerably longer, up to several hours. For example, PGExplainer requires approximately 5 minutes for AIFB and around 2 hours for MUTAG, while SubgraphX takes around 45 minutes and 2 hours, respectively. Notably, on BGS, PGExplainer takes a significant 12 hours to converge, whereas SubgraphX only needs around 1 hour for explanation. The difference arises from their methodologies: PGExplainer trains an explainer network on the entire dataset for instant explanations, while SubgraphX generates explanations for each target node individually. As SubgraphX uses Monte-Carlo Tree Search (MCTS) to explore plausible explanations, the number of subgraphs explored, and runtime can vary between runs.

Explanation formalism and dataset (pre)processing. Whereas subgraph approaches identify important subgraphs as explanations, logical approaches leverage semantic information to yield logical expressions. Explainers differ in how they process datasets. For example, subgraph methods often treat literals (e.g., numeric and binary literals) as nodes, whereas logical methods have special support for literals and data properties. Future work might explore alternative preprocessing schemes and focus on enabling GNN models to integrate and process semantic knowledge directly.

Real-world vs. synthetic datasets. In this paper, we focus on real-world datasets where the "correct" explanation is unknown. Future work might integrate synthetic datasets with known ground truth explanations. At the time of writing, however, publicly available synthetic datasets for explainability are limited to homogeneous graphs whereas we focus on heterogeneous graphs.

Local vs. global explainability. Explainable AI (XAI) is still in its early stages of development, with significant strides made in providing local explanations but limited progress in achieving global explainability. While local explainers offer insights into individual predictions, allowing users to understand why a model made a specific decision, global explainability remains a challenge. Global explanations provide a holistic understanding of a model's behavior across an entire dataset or system, offering insights into overall patterns, biases, and decision-making processes. However, due to the complexity of many AI models, such as deep neural networks and graph neural networks, achieving comprehensive global explanations is difficult. The authors of the subgraph-based explainers used in this work argue that while they aim to provide a sense of global explanations, they fall short of achieving true global explainability. As XAI research continues to evolve, addressing the gap in global explainability will be crucial for building trust, transparency, and accountability in AI systems across various domains.

6 Conclusions

In this work, we evaluated subgraph-based and logic-based explainers on real-world heterogeneous graphs and developed an evaluation framework called EDGE. We found that logic-based methods exhibit a high recall, whereas subgraph approaches often yield a high precision. Today's subgraph-based approaches offer a higher fidelity to the GNN predictions; logical approaches are faster and produce a single explanation per GNN and class. Whereas logic-based methods excel in handling complex datasets with semantic information, they do not support multi-class classification and regression tasks out of the box.

Future work includes the addition of explainers with a diverse range of explanation formalisms, as well as additional datasets and evaluation metrics. We plan to extend the framework as a standalone evaluation framework and release it as an open-source library with a focus on knowledge graph-based explainable AI.

Acknowledgments

This work has been supported by the Ministry of Culture and Science of North Rhine-Westphalia (MKW NRW) within the project SAIL under the grant no NW21-059D.

References

[1] Chirag Agarwal, Owen Queen, Himabindu Lakkaraju, and Marinka Zitnik. 2023. Evaluating explainability for graph neural networks. *Scientific Data* 10, 1 (2023), 144.

[2] Kenza Amara, Zhitao Ying, Zitao Zhang, Zhichao Han, Yang Zhao, Yinan Shan, Ulrik Brandes, Sebastian Schemm, and Ce Zhang. 2022. GraphFramEx: Towards Systematic Evaluation of Explainability Methods for Graph Neural Networks. In *LoG (Proceedings of Machine Learning Research, Vol. 198)*. PMLR, 44.

[3] Alkid Baci and Stefan Heindorf. 2023. Accelerating Concept Learning via Sampling. In *CIKM*. ACM, 3733–3737.

[4] Federico Baldassarre and Hossein Azizpour. 2019. Explainability Techniques for Graph Convolutional Networks. *CoRR* abs/1905.13686 (2019).

[5] Stephan Bloehdorn and York Sure. 2007. Kernel Methods for Mining Instance Data in Ontologies. In *ISWC/ASWC (Lecture Notes in Computer Science, Vol. 4825)*. Springer, 58–71.

[6] Dan Busbridge, Dane Sherburn, Pietro Cavallo, and Nils Y. Hammerla. 2019. Relational Graph Attention Networks. *CoRR* abs/1904.05811 (2019).

[7] David Jaime Tena Cucala, Bernardo Cuenca Grau, Egor V. Kostylev, and Boris Motik. 2022. Explainable GNN-Based Models over Knowledge Graphs. In *ICLR*. OpenReview.net.

[8] Rajarshi Das, Shehzaad Dhuliawala, Manzil Zaheer, Luke Vilnis, Ishan Durugkar, Akshay Krishnamurthy, Alex Smola, and Andrew McCallum. 2018. Go for a Walk and Arrive at the Answer: Reasoning Over Paths in Knowledge Bases using Reinforcement Learning. In *ICLR (Poster)*. OpenReview.net.

[9] Asim Kumar Debnath, Rosa L Lopez de Compadre, Gargi Debnath, Alan J Shusterman, and Corwin Hansch. 1991. Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. correlation with molecular orbital energies and hydrophobicity. *Journal of medicinal chemistry* 34, 2 (1991), 786–797.

[10] Caglar Demir and Axel-Cyrille Ngonga Ngomo. 2023. Neuro-Symbolic Class Expression Learning. In *IJCAI*. ijcai.org, 3624–3632.

[11] Luis Galárraga, Christina Teflioudi, Katja Hose, and Fabian M. Suchanek. 2015. Fast rule mining in ontological knowledge bases with AMIE+. *VLDB J.* 24, 6 (2015), 707–730.

[12] Luis Antonio Galárraga, Christina Teflioudi, Katja Hose, and Fabian M. Suchanek. 2013. AMIE: association rule mining under incomplete evidence in ontological knowledge bases. In *WWW*. International World Wide Web Conferences Steering Committee / ACM, 413–422.

[13] DICE Research Group. 2023. *Ontolearn*. https://github.com/dice-group/Ontolearn Accessed: November 26, 2023.

[14] Hui Han, Tianyu Zhao, Cheng Yang, Hongyi Zhang, Yaoqi Liu, Xiao Wang, and Chuan Shi. 2022. OpenHGNN: An Open Source Toolkit for Heterogeneous Graph Neural Network. In *CIKM*. ACM, 3993–3997.

[15] Stefan Heindorf, Lukas Blübaum, Nick Düsterhus, Till Werner, Varun Nandkumar Golani, Caglar Demir, and Axel-Cyrille Ngonga Ngomo. 2022. EvoLearner: Learning Description Logics with Evolutionary Algorithms. In *WWW*. ACM, 818–828.

[16] Anna Himmelhuber, Stephan Grimm, Sonja Zillner, Mitchell Joblin, Martin Ringsquandl, and Thomas A. Runkler. 2021. Combining Sub-symbolic and Symbolic Methods for Explainability. In *RuleML+RR (Lecture Notes in Computer Science, Vol. 12851)*. Springer, 172–187.

[17] Qiang Huang, Makoto Yamada, Yuan Tian, Dinesh Singh, and Yi Chang. 2023. GraphLIME: Local Interpretable Model Explanations for Graph Neural Networks. *IEEE Trans. Knowl. Data Eng.* 35, 7 (2023), 6968–6972.

[18] Rebecca C Jackson, James P Balhoff, Eric Douglass, Nomi L Harris, Christopher J Mungall, and James A Overton. 2019. ROBOT: a tool for automating ontology workflows. *BMC bioinformatics* 20 (2019), 1–10.

[19] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR (Poster)*. OpenReview.net.

[20] N'Dah Jean Kouagou, Stefan Heindorf, Caglar Demir, and Axel-Cyrille Ngonga Ngomo. 2022. Learning Concept Lengths Accelerates Concept Learning in ALC. In *ESWC (Lecture Notes in Computer Science, Vol. 13261)*. Springer, 236–252.

[21] N'Dah Jean Kouagou, Stefan Heindorf, Caglar Demir, and Axel-Cyrille Ngonga Ngomo. 2023. Neural Class Expression Synthesis. In *ESWC (Lecture Notes in Computer Science, Vol. 13870)*. Springer, 209–226.

[22] N'Dah Jean Kouagou, Stefan Heindorf, Caglar Demir, and Axel-Cyrille Ngonga Ngomo. 2023. Neural Class Expression Synthesis in *ALCHIQ(D)*. In *ECML/PKDD (4) (Lecture Notes in Computer Science, Vol. 14172)*. Springer, 196–212.

[23] N'Dah Jean Kouagou, Stefan Heindorf, Caglar Demir, and Axel-Cyrille Ngonga Ngomo. 2024. ROCES: Robust Class Expression Synthesis in Description Logics via Iterative Sampling. In *IJCAI*. ijcai.org.

[24] Jonathan Lajus, Luis Galárraga, and Fabian M. Suchanek. 2020. Fast and Exact Rule Mining with AMIE 3. In *ESWC (Lecture Notes in Computer Science, Vol. 12123)*. Springer, 36–52.

[25] Jens Lehmann, Sören Auer, Lorenz Bühmann, and Sebastian Tramp. 2011. Class expression learning for ontology engineering. *J. Web Semant.* 9, 1 (2011), 71–81.

[26] Jiayi Li, Sheetal Satheesh, Stefan Heindorf, Diego Moussallem, René Speck, and Axel-Cyrille Ngonga Ngomo. 2024. AutoCL: AutoML for Concept Learning. In *World Conference on Explainable Artificial Intelligence*. Springer, 117–136.

[27] Meng Liu, Hongyang Gao, and Shuiwang Ji. 2020. Towards Deeper Graph Neural Networks. In *KDD*. ACM, 338–348.

[28] Dongsheng Luo, Wei Cheng, Dongkuan Xu, Wenchao Yu, Bo Zong, Haifeng Chen, and Xiang Zhang. 2020. Parameterized Explainer for Graph Neural Network. In *NeurIPS*.

[29] Christian Meilicke, Melisachew Wudage Chekol, Patrick Betz, Manuel Fink, and Heiner Stuckenschmidt. 2024. Anytime bottom-up rule learning for large-scale knowledge graph completion. *VLDB J.* 33, 1 (2024), 131–161.

[30] Christian Meilicke, Melisachew Wudage Chekol, Daniel Ruffinelli, and Heiner Stuckenschmidt. 2019. Anytime Bottom-Up Rule Learning for Knowledge Graph Completion. In *IJCAI*. ijcai.org, 3137–3143.

[31] Phillip E. Pope, Soheil Kolouri, Mohammad Rostami, Charles E. Martin, and Heiko Hoffmann. 2019. Explainability Methods for Graph Convolutional Neural Networks. In *CVPR*. Computer Vision Foundation / IEEE, 10772–10781.

[32] Ali Sadeghian, Mohammadreza Armandpour, Patrick Ding, and Daisy Zhe Wang. 2019. DRUM: End-To-End Differentiable Rule Mining On Knowledge Graphs. In *NeurIPS*. 15321–15331.

[33] Michael Sejr Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling Relational Data with Graph Convolutional Networks. In *ESWC (Lecture Notes in Computer Science, Vol. 10843)*. Springer, 593–607.

[34] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2017. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. In *ICCV*. IEEE Computer Society, 618–626.

[35] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. In *ICLR (Workshop Poster)*.

[36] Karen Simonyan and Andrew Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *ICLR*.

[37] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic Attribution for Deep Networks. In *ICML (Proceedings of Machine Learning Research, Vol. 70)*. PMLR, 3319–3328.

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *NIPS*. 5998–6008.

[39] Minh N. Vu and My T. Thai. 2020. PGM-Explainer: Probabilistic Graphical Model Explanations for Graph Neural Networks. In *NeurIPS*.

[40] Minjie Wang, Lingfan Yu, Da Zheng, Quan Gan, Yu Gai, Zihao Ye, Mufei Li, Jinjing Zhou, Qi Huang, Chao Ma, Ziyue Huang, Qipeng Guo, Hao Zhang, Haibin Lin, Junbo Zhao, Jinyang Li, Alexander J. Smola, and Zheng Zhang. 2019. Deep Graph Library: Towards Efficient and Scalable Deep Learning on Graphs. *CoRR* abs/1909.01315 (2019).

[41] Patrick Westphal, Lorenz Bühmann, Simon Bin, Hajira Jabeen, and Jens Lehmann. 2019. SML-Bench - A benchmarking framework for structured machine learning. *Semantic Web* 10, 2 (2019), 231–245.

[42] Wenhan Xiong, Thien Hoang, and William Yang Wang. 2017. DeepPath: A Reinforcement Learning Method for Knowledge Graph Reasoning. In *EMNLP*. Association for Computational Linguistics, 564–573.

[43] Zhitao Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. 2019. GNNExplainer: Generating Explanations for Graph Neural Networks. In *NeurIPS*. 9240–9251.

[44] Hao Yuan, Jiliang Tang, Xia Hu, and Shuiwang Ji. 2020. XGNN: Towards Model-Level Explanations of Graph Neural Networks. In *KDD*. ACM, 430–438.

[45] Hao Yuan, Haiyang Yu, Shurui Gui, and Shuiwang Ji. 2023. Explainability in Graph Neural Networks: A Taxonomic Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* 45, 5 (2023), 5782–5799.

[46] Hao Yuan, Haiyang Yu, Jie Wang, Kang Li, and Shuiwang Ji. 2021. On Explainability of Graph Neural Networks via Subgraph Explorations. In *ICML (Proceedings of Machine Learning Research, Vol. 139)*. PMLR, 12241–12252.

[47] Muhan Zhang and Yixin Chen. 2018. Link Prediction Based on Graph Neural Networks. In *NeurIPS*. 5171–5181.

[48] Muhan Zhang, Zhicheng Cui, Marion Neumann, and Yixin Chen. 2018. An End-to-End Deep Learning Architecture for Graph Classification. In *AAAI*. AAAI Press, 4438–4445.

[49] Bolei Zhou, Aditya Khosla, Àgata Lapedriza, Aude Oliva, and Antonio Torralba. 2016. Learning Deep Features for Discriminative Localization. In *CVPR*. IEEE Computer Society, 2921–2929.