Graph-Based Explainable AI: A Comprehensive Survey

Margarita Bugueño, Russa Biswas, Gerard de Melo

Graph-Based Explainable AI: A Comprehensive Survey

MARGARITA BUGUEÑO, Hasso Plattner Institute (HPI) / University of Potsdam, Germany
RUSSA BISWAS, Aalborg University, Denmark
GERARD DE MELO, Hasso Plattner Institute (HPI) / University of Potsdam, Germany

Graph-based learning models learn structure-aware and node-level representations through relational associations between data points, enhancing predictions and explainability. The ability of Graph Neural Network (GNN) models to learn from non-Euclidean spaces, has shifted explainability efforts towards GNNs, neglecting other significant methodologies. This survey addresses this gap by including traditional machine learning and deep learning models, based on Reinforcement Learning, Multi-Hop Reasoning, Knowledge Graphs, and GNNs. It proposes a hierarchical categorization of graph explanation models based on explanation modalities and approaches. Furthermore, it examines the merits, drawbacks, and suitability of each strategy across domains and tasks followed by future research directions.

CCS Concepts: • **General and reference** → **Reliability**; *Surveys and overviews*; • **Human-centered computing** → *Human computer interaction (HCI)*; • **Computing methodologies** → **Machine learning**; **Artificial intelligence**; **Knowledge representation and reasoning**.

Additional Key Words and Phrases: Explainable Artificial Intelligence, Interpretable Machine Learning, Explainable Graphs, Explainability, Interpretability, Graph Neural Networks

1 INTRODUCTION

The notable advancement in technology has allowed the development and deployment of intricate models with a substantial number of learning hyperparameters, such as Transformer-based language models and the subsequent series of generative pre-trained models (GPT) [18, 29, 96, 121]. These advancements have enhanced the power of deep learning systems and led to outstanding results across different fields [15, 153]. Despite achieving state-of-the-art performance, their internal processes remain largely uninterpretable, making it particularly challenging to elucidate the underlying mechanisms that contribute to the generated predictions.

While in the early 2000s, explaining the reason behind certain predictions made and identifying what patterns trigger certain behaviors was treated as an optional secondary objective, it has become a necessity today. The emergence of the field of Explainable Artificial Intelligence (XAI) has supported practitioners in various applications, ranging from object detection on images [107] to critical scenarios such as financial services [97] and healthcare [7].

Traditionally, XAI focused on explaining and interpreting the decisions of a predictor trained on Euclidean data, including tabular data, images, and time series [9, 84, 86]. However, as several high-impact machine learning tasks operate on non-euclidean data, i.e. graphs, such as physics simulations, recommendation systems, and network analysis, XAI has expanded to address the challenges posed by these structures.

Recently, the rise of Graph Neural Network (GNN) models has facilitated the resolution of numerous applications involving non-Euclidean data. These models capture relevant patterns while accounting for interdependencies between graph nodes via message passing [3, 147]. Consequently, several GNN explainers have been proposed to study the variations in the model gradients or implement component perturbations on input data to discern the relevance of distinct graph components, namely nodes, edges, and, where applicable, node features.

Authors' addresses: Margarita Bugueño, margarita.bugueno@hpi.de, Hasso Plattner Institute (HPI) / University of Potsdam, Prof.-Dr.-Helmert-Str. 2-3, Potsdam, Brandenburg, Germany, 14482; Russa Biswas, Aalborg University, A. C. Meyers Vænge 15, Copenhagen, Denmark, 2450, rubi@cs.aau.dk; Gerard de Melo, Hasso Plattner Institute (HPI) / University of Potsdam, Prof.-Dr.-Helmert-Str. 2-3, Potsdam, Brandenburg, Germany, 14482, gerard.demelo@hpi.de.

There are many efforts to explain graph data through GNNs [105, 143, 147]. As outlined in previous studies [28, 146], these proposals can be grouped into five main categories: **(i)** Gradient, **(ii)** Perturbation, **(iii)** Decomposition, **(iv)** Surrogate, and **(v)** Generation, based on their approach to generating explanations. While other surveys offer alternative categorizations for graph explainers [2, 71, 73], they are limited to GNN models and their variants, specifically for network analysis tasks such as node classification, graph classification, and link prediction. Nonetheless, many current graph-based models adopt alternative methodologies to provide human-understandable and lucid explanations without relying on GNNs. These models apply classic machine learning strategies such as logic rules extraction, relevant random walks, and the analysis and integration of input features for explaining model predictions in a broader range of tasks, including clustering, recommender systems, and diverse NLP downstream tasks.

This survey provides a comprehensive overview of the explainability of graph-based models, introducing a novel categorization of the strategies used, and the approaches followed to generate the explanations. The paper critically evaluates the merits and drawbacks of current proposals while delving into the open challenges within the field. Our contributions are listed as:

- This survey provides a novel hierarchical categorization for existing graph explainers, based on the format of the explanation (explanation modality) and the strategy followed for obtaining the explanation (explainability approach).
- The paper extends the limited research concerning the comprehension of graph explainability and its influence within the discipline. It broadens the examination of GNN explainers to encompass a more extensive array of graph techniques and methodologies.
- We review the advantages and limitations of the current approaches, with an exploration of their suitability to different domains and tasks.

The paper is structured as follows: Section 2 delves into the definition of explainability, exploring its nuances with interpretability. Section 3 provides an overview of the proposed categorization based on two levels: the explanation modality and the explainability approach followed by the corresponding explainer. Following our categorization, Section 4 introduces each graph-based explainer proposed to date and addresses their respective limitations. The methods are visited chronologically, as illustrated in Figure 1. Section 7 offers a comparative analysis concerning explanation modality, explainability approach, and target tasks. A detailed description of the evaluation scheme utilized to validate the models, along with the metrics and datasets acknowledged in the literature, is presented in Section 5. Section 6 discusses application domains and outcomes from previous studies in areas such as medicine, chemistry, natural language processing (NLP), and recommender systems. Finally, Section 8 provides a retrospective analysis of methods proposed over the years, presenting paper statistics and highlighting the open challenges within the field. A compilation of all the papers included within this survey is accessible via https://github.com/Buguemar/graphing-a-decision.

2 REVISITING EXPLAINABILITY

Explaining the rationale behind a decision is a fundamental aspect of intelligence and is often a prerequisite for establishing a trustworthy answer to a question. Currently, many Artificial Intelligence (AI) researchers advocate for explainability in the field. This is not only to elucidate black box models and render them comprehensible to fellow practitioners but also to serve practical objectives. These include understanding system failures, identifying features associated with patterns or classes, and thereby uncovering weaknesses and limitations within the learning

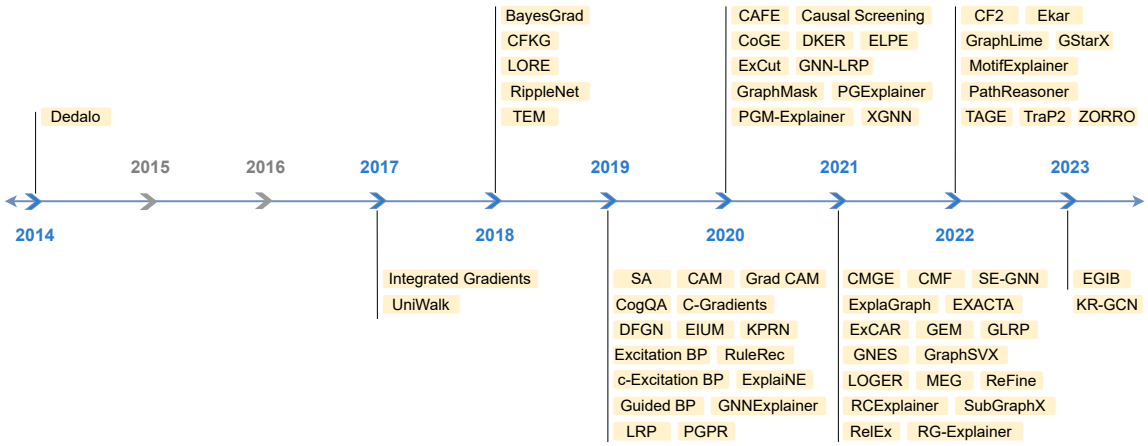


Fig. 1. Timeline of the existing graph-based explainers considered in this survey.

system. Moreover, in 2017, the Association for Computing Machinery US Public Policy Council (USACM) stated that explainability is one of the seven principles for algorithmic transparency and accountability [24].

Despite this clear requisite for AI systems, there is a particular discussion on what truly is an explainable system. Several researchers use interpretability and explainability in a unified way, while others call for the correct differentiation between the concepts. In [21], the authors refer to XAI as the field of study that focuses research on machine learning interpretability for a more transparent AI. In turn, [32] defined interpretability as the *ability to explain or to present in understandable terms to a human.*

In this survey, we take a step back and present a categorization and discussion of the multiple models and strategies proposed to date to clarify the functionality and establish a causal relationship between the inputs and output of a predictive model. Therefore, this article does not differentiate between explainability and interoperability.

However, some strategies have been involved in controversy when defining whether they are sufficient to explain and understand the reasons for the outputs of a trained model. Such is the case with the attention modules. There is a great debate questioning the validity of this strategy as a method of interpretation or explanation. Even though the use of attention modules consistently yields improved performance on some tasks [19, 20, 131, 141], previous studies have postulated that attention weights often fail to identify the most relevant representations to the model's final decision [109]. In [60], the authors performed extensive experiments across various NLP tasks, concluding that the learned attention weights are frequently uncorrelated with gradient-based measures of feature importance. Moreover, different attention distributions yield equivalent predictions despite attending to entirely different input features. On the other hand, the study has received criticism as some assumptions used in the experimental setup suggest a special amount of freedom. Therefore, their findings could be insufficient evidence against attention as an explanation [132]. In addition, [45] states that the applicability of attention as an explanation in image data may not generalize to text. Thus, human understanding mechanisms could differ between both domains.

The debate is ongoing, with numerous researchers contributing to a comprehensive literature on the topic from their various research areas [85, 100, 149]. This survey focuses on different types of graph explainers, excluding attention-based models, as these models have been the prime focus of several previous studies [70, 117].

3 CATEGORIZATION OF GRAPH EXPLAINERS

Let $G = (V, E, X)$ be a graph, where V represents the set of vertices (or nodes) contained in the graph, E is the set of edges defining associations between pairs of vertices, and X serves as a descriptor for node and edge features [3, 150]. Additionally, a graph-based learning model denoted as f is defined, with $f(G(V, E, X)) = y$, where y denotes the prediction made by the model. A graph-based explainer Φ is defined as $\Phi(f(G)) = G_\Phi$ whose input corresponds to the learning model f subjected to the input graph G, and the output yields the corresponding graph-based explanation G_Φ.

There exist several methods to explain the predictions of graph-based learning models, focussing on diverse aspects and providing different views of understanding. Some of these proposals focus on identifying how relevant the components of the input graph are for obtaining a given prediction. Others reduce the input graph to the most determining graph components based on specific heuristics or build a new graph from scratch as an explanation.

In this paper, a novel hierarchical categorization of the existing explanation strategies for graph-based learning models is proposed. Our categorization, depicted in Figure 2, is structured based on **(i)** explanation modality, followed by **(ii)** explainability approach. The former refers to the format of the explanation generated, which encompasses operations such as scoring of graph components, extraction of explanations from input data, or generation of explanations from scratch. On the other hand, the approach refers to the internal strategy employed to derive explanations, namely gradients, decomposition, path reasoning, data integration, use of a surrogate model, perturbation, or graph creation.

3.1 Explanation Modality

Scoring: Inspired by image processing [1], where saliency maps refer to unique features (pixels) that depict the visually alluring locations in an image, scoring explanation operation refers to any method that has an objective to decide on the scores to award over the multiple graph components as an explanation. These scores are distributed over the nodes, edges, and over node and edge feature vectors whenever possible. There are two different approaches to generating said distribution of relevance scores for the components of a graph: gradients and decomposition [47, 105, 118].

Extraction: The objective of the extraction explanation procedure is to elucidate the predictions obtained by a previously trained learning model for a given input graph through the derivation of a modified version of the input data. Typically, this extraction is achieved by discarding graph components from the original input graph, employing predefined heuristics that gauge the significance of these components when predicting a certain label within an already trained model. The extraction operation can be sub-classified according to the topological characteristics of the generated explanation, delineated into sequential paths [81, 124], logic rules [44, 126], and sub-graphs [57, 143]. Sequential paths aim to explain associations between nodes by looking for a directed sequential path, while subgraphs do not.

Generation: Generation-based explanation methods consist of a generator building an explanation graph from scratch so that the generated network can maximize a target prediction. Standard learning models operating on Euclidean data generally define the dependency between the input features and the learnable model parameters of the corresponding model. Conversely, this process becomes intricate for models operating on graph structures, as establishing a connection between their topological information and the corresponding model parameters poses a challenge. Nonetheless, this category has garnered attention due to its capacity to furnish high-level insights and a comprehensive understanding of the workings of graph learning models [31, 145].

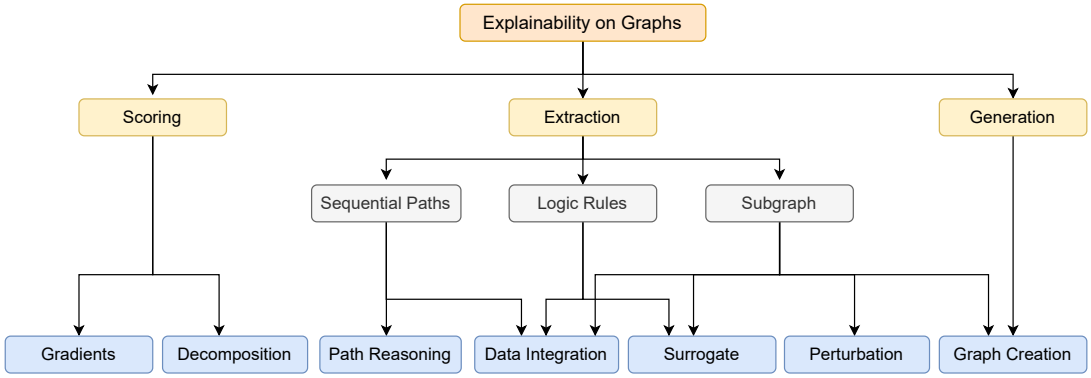


Fig. 2. **Categorization of graph-based explainers.** The first branch (in yellow) reflects the explanation modality as scoring, extraction, and generation, where extraction is further subdivided into: extraction of sequential paths, logic rules, or a subgraph. The second branching level (in blue) shows the Explainability approach associated with each explanation modality.

3.2 Explainability Approach

Gradients: Following the sensitivity analysis line, the gradient-based attribution methods employ gradients and the learned hidden feature maps to approximate the input features' importance. This is done following a backpropagation analysis, in which every gradient quantifies how much a slight change in a small neighbourhood of a specific input dimension would change the predictions.

Decomposition: Given a learning neural model, the decomposition strategy produces relevance maps by decomposing the output signal of every transformation layer into a combination of its inputs via backwards. This approach studies the model parameters to reveal the relationship of the features in the input space. Unlike gradient-based methods, the decomposition approach identifies which input features contribute the most to the final prediction rather than focusing on its variation, which is valuable for researchers interested in a deeper analysis of the contributing factors.

Path Reasoning: In the context of graph theory, a path refers to a sequence of vertices, where an edge connects each consecutive pair of vertices and defines a way to traverse the graph by moving from one vertex to another along the edges, following a specific sequence. Path reasoning explainers discern meta-paths from the input graph, revealing dependencies among nodes. Utilized predominantly in recommender system scenarios, the path-based explanatory approach leverages external knowledge bases to acquire proficient representations of network items based on their interactions.

Data Integration: Data integration-based strategies explain model predictions based on rigorous analysis of available data. It bases the explanation on other graphs from available training data by identifying similarities and other relevant patterns that allow it to recognize the components that make the given input graph different from others with a different label, i.e., the explanation. Data integration explainers, as in the case of path-reasoning strategies, usually integrate external knowledge bases to enhance the extraction of patterns of interest.

Surrogate: A surrogate model serves as a simplified and interpretable model used to approximate the behaviour and predictions of a more complex learning model in the neighbouring areas of the original input examples. This approach relies on the assumption that the relations in nearby regions are less complex, allowing simpler models to effectively capture them. Consequently, the surrogate model offers insights into the decision-making process of a graph-based learning model in a more accessible and transparent manner.

Perturbation: The strategy studies the output variations for different input perturbations. Given the input graph G labeled as y, the explanation model evaluates the need to retain the graph component c based on its impact on the final prediction of G when removed from the original structure. If removing c does not alter the prediction y, c is considered non-essential for explaining the graph and is thus removed. Depending on the task, masks are learned to decide the remaining graph components that describe the relevant input information.

Graph Creation: The graph creation approach involves creating a graph structure as an explanation in an iterative-generative manner. This category involves studying what input patterns lead to certain behaviours by maximizing a target prediction. In general terms, a graph creation explainer defines a seed node as the initial most influential node, and then a new node is added from the candidate neighbours based on specific heuristics step-by-step. A stopping criterion is also learned to guarantee a minimal explanatory graph.

Note that one explanation modality may incorporate various approaches, while multiple explanation modalities may adopt the same approach. A description of the explainers and their categorization is provided in Section 4.

4 EXPLAINING GRAPHS

Other surveys have proposed alternative categorizations for graph-based explainers [2, 71, 73]. However, they limited the study to only GNN-based models and thus to a severely restricted range of tasks and domains: graph classification, node classification, and link prediction. Nevertheless, a considerable number of graph-based methods were adopted for more varied tasks such as recommender systems, signal, and time series analysis, as well as a range of NLP tasks [33, 90, 104, 148]. As a result, several alternative methodologies have been proposed to provide explanations without directly relying on GNNs but applying classic machine learning strategies.

This section introduces the multiple graph-based explainers proposed up-to-date, including both approaches in a more complete way. The methods are classified according to the explanation procedure followed into graph scoring, explanation extraction, and generation of explanation.

4.1 Scoring

One of the traditional strategies for identifying the most influential input patterns for obtaining a specific output corresponds to Sensitivity Analysis (SA) [66, 156]. Since a model's parameters encapsulate what it has learned during training, they can assist in debugging the model. Then, given a differentiable function f, SA uses the norm of the prediction gradient concerning the input to describe how input variations produce changes in the output.

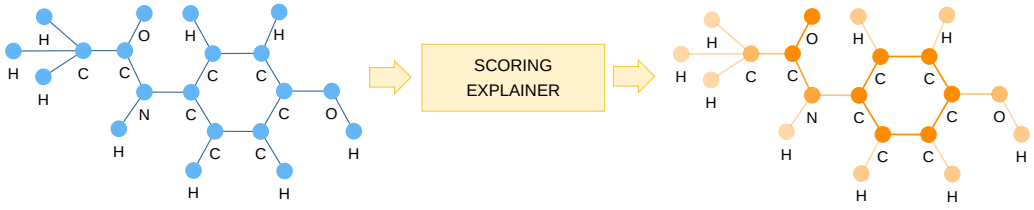


Fig. 3. A paracetamol molecule comprising carbon (C), nitrogen (N), oxygen (O), and hydrogen (H) atoms, alongside a scoring explanation example of its predicted energy. Color shade indicates how relevant is the corresponding graph element for the prediction. Example based on the results reported in [105].

While extensively utilized in image processing [1], Sensitivity Analysis was extended to encompass explanatory functionalities within the graph domain [12], specifically for node classification. The authors highlighted some crucial

differences in this domain application, as in a graph where edge features are not present or are all identical, neither gradients nor relevance would be back-propagated even though the presence of an edge between two nodes is a valuable source of information. This reflects one of the main issues of the strategy as it only reflects the sensitivity between input-output, which is not precisely the importance of the input features.

Another classic strategy is the Contrastive Gradients (C-Gradients) method [113] which was originally proposed as a gradient-based attribution method in which every gradient quantifies how much a slight change in a small neighborhood of a specific input dimension would change the predictions. Later, Guided Back-propagation (Guided BP) [12, 116] extended it by differentiating the output of a predictive model concerning the model input creating a heat map, but limiting it to the features that have an excitatory effect on the output, i.e., by clipping the negative gradients to zero. Unfortunately, both proposals suffer from a gradient saturation problem [111] which was later addressed by Integrated Gradients [118]. Unlike previously proposed methods, Integrated Gradients require no modification to the original graph and average the gradients over a set of interpolated inputs to satisfy the so-called axiom *completeness*. That is, the attributions add up to the difference between the output of a predictive model at the input x and a baseline input x'.

In 2015, LRP [9] was introduced as a widely used attribution method. It decomposes the output signal of every transformation into a combination of inputs, producing relevance maps for a neural model. Unlike the previous gradient-based methods, LRP identifies which input features contribute the most to the final prediction rather than focusing on its variation, which is valuable for researchers interested in a deeper analysis of the contributing factors. In the context of graphs, the strategy has been applied for node and graph classification tasks as well as regression [12, 106].

Some years later, Class Activation Maps (CAM) were introduced by employing global average pooling in CNNs for image processing tasks. In this sense, a class activation map for a particular category indicates the discriminative image regions the model uses to identify that category. This proposal was adapted to the graph domain by employing a global average pooling layer over the graph nodes in the corresponding input graph followed by a softmax. Hence, the node embeddings are combined by weighted summations to obtain importance scores for input nodes [93]. The architectural requirements for this method preclude the application of more complex graph neural networks, motivating the introduction of gradient-weighted CAM (Grad-CAM) [93, 108], where the architectural restriction is relaxed by employing feature map weights that are based on back-propagated gradients. Nevertheless, both variants assume that final node embeddings can reflect the input importance and only consider graph classification tasks.

Motivated by this uncertainty on the learned node representation, BayesGrad was proposed [5]. The authors noted that existing learning methods, formulated within the framework of maximum likelihood estimation, can result in unstable parameters sensitive to changes in training data. BayesGrad addresses this by quantifying prediction uncertainty using Bayesian predictive distribution, assessing the importance of each node in the input graph through dropout. The approach was evaluated across graph classification and regression tasks.

Excitation Backpropagation and its variant, Contrastive Excitation Backpropagation [93], were also introduced. While the former was designed for CNNs, the latter was extended to graph convolutional neural networks. Both methods, rooted in the law of total probability, state that the probability of a neuron a in a layer l equals the total probabilities it outputs to all connected neurons in the subsequent layer $l+1$. Thus, the importance score is determined by decomposing the target probability into several conditional probability terms. However, it shares limitations with LRP-based strategies. Since LRP can only assess the importance of graph components and not structures, GNN-LRP was proposed [105]. It inspected the prominence of different graph walks by a t-order terms Taylor decomposition of model prediction, where each corresponds to a t-step walk treated as its importance score. [22] extended the analysis for metastasis prediction in breast cancer. In this study, every patient represents a signal in the input graph and the most relevant

Table 1. A comparison table for scoring explainers. Qn and Ql stand for quantitative and qualitative validation, respectively. Note that all the listed methods were specifically designed for GNNs.

Method	Approach	GNN	Tasks			Validation		Source
		based	GC	NC	Other	Data	Quality	Code
SA [10, 12]	Gradients	✓	✓	✓	Regression	Real	Ql	✓
C-Gradients [93, 113]	Gradients	✓	✓	-	-	Real/Synth.	Qn/Ql	-
LRP [9, 12]	Decomposition	✓	✓	✓	Regression	Real	Ql	✓
Guided BP [12, 116]	Gradients	✓	✓	✓	-	Real	Ql	✓
CAM [93, 154]	Gradients	✓	✓	-	-	Real/Synth.	Qn/Ql	-
GradCAM [93, 108]	Gradients	✓	✓	-	-	Real/Synth.	Qn/Ql	-
Integrated Gradients [118]	Gradients	✓	✓	-	-	Real	Ql	✓
BayesGrad [5]	Gradients	✓	✓	-	Regression	Real/Synth.	Qn/Ql	✓
Excitation BP [93]	Decomposition	✓	✓	✓	-	Real/Synth.	Qn/Ql	-
c-Excitation BP [93]	Decomposition	✓	✓	-	-	Real/Synth.	Qn/Ql	-
GNN-LRP [105]	Decomposition	✓	✓	-	-	Real/Synth.	Qn	✓
GLRP [22]	Decomposition	✓	✓	-	-	Real	Ql	✓
GNES [47]	Gradients	✓	✓	-	-	Real	Qn/Ql	✓

vertices constitute a molecular subnetwork of explanation. However, the strategy requires an extensive understanding of the learning model architecture and thus, it can only be successfully applied by experts, as approximations for input graph sub-structures may be inaccurate and computationally complex.

Another perspective is employed by GNN Explanation Supervision (GNES) [47]. It is proposed to correct unreasonable explanations and learn how to explain GNNs accurately. The authors propose a unified explanation method to generate node and edge explanations with consistency regularization among them by optimizing model prediction and explanation with weak supervision from human explanation annotations. This is one of the first strategies proposing to learn the explanation through a joint optimization of the model prediction loss and the explanation loss.

Limitations. Despite scoring methods provide an intuitive explanation result. These methods only reflect the sensitivity between input-output, which is not precisely the importance of the input features. Additionally, explanations produced by scoring methods tend to diverge from how a human would intuitively describe the process of cause and effect. Since scoring explainers require an extensive understanding of the structure of the learning model (neural layers and the respective neurons composing them), only experts can apply them effectively. Moreover, given the high computation involved in the explanation process, these methods are only recommended for small training data.
In particular, decomposition-based scoring methods study the importance of the graph nodes, albeit in a constrained manner, as they cannot discern pertinent graph structures, such as subgraphs or walks, as other alternative explainers do. Lastly, all methods within this category were proposed to explain predictions made by GNNs (Table 1). Therefore, they exhibit proficiency solely in addressing a limited set of graph characterization tasks, such as graph classification (GC) and node classification (NC), and only a few can encompass graph feature regression.

4.2 Graph Extraction

Extraction explainers provide subgraphs of the original graph as explanations. The selection of graph components to retain is determined with regard to their significance in predicting a specific label. Specifically, given the input graph G labeled as y, the explanation model $\Phi(f(G))$ evaluates the need to retain the graph component c according to how much the final prediction of G is affected when discarding c from the structure.

According to the explanatory topology, extraction explainers are categorized into sequential paths, logic rules, or subgraphs. The former retrieves a linear structure within the graph, wherein each node is sequentially connected to the next, forming a sequence. Therefore, there is no vertex repetition, and each edge is distinct. The extraction of logic rules involves the extraction of a formal statement that can be generalized from the graph's topology, edge weights, node features, or other relevant characteristics. As a more general concept, the extraction of subgraphs refers to explainers whose explanation is a graph formed by selecting a subset of vertices and edges from the original structure without the constraints imposed by alternative extraction explainers.

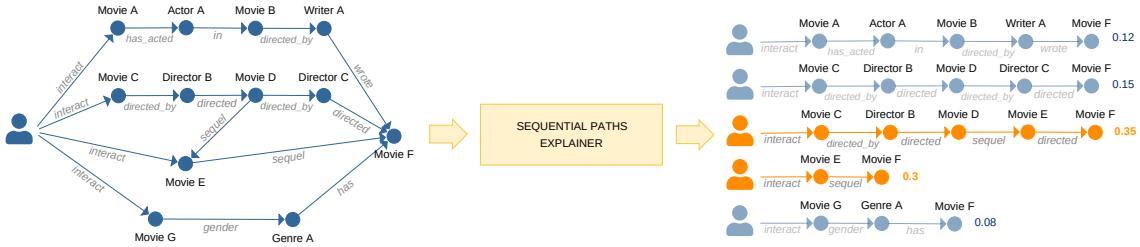


Fig. 4. An example of a sequential paths-based explainer. The input graph represents a knowledge graph considered by a system recommending the item *Movie F* for the given user. The final explanation paths are marked in orange as those higher-quality paths.

4.2.1 ***Sequential Path.*** Dedalo [120], a framework based on inductive logic programming, uses linked data to automatically generate explanations for clusters obtained through knowledge discovery processes. It uses a graph-search process, including URI expansion, path extraction, ranking, path values selection, and hypothesis evaluation. The framework employs various strategies to guide this traversal, reducing time to obtain the best explanation. Experiments show that Dedalo can find relevant and sophisticated Linked Data explanations from various domains, demonstrating its effectiveness in interpreting and explaining clusters derived from different datasets. The UniWalk [91] model is a recommendation system that uses social network and rating data to provide personalized item recommendations. It combines both types of data into a unified graph and uses network embedding to extract latent features of users and items, enabling it to predict ratings and provide explanations for the recommendations. The model's learning process involves learning from positive walks, negative walks, and unweighted walks. The performance and robustness of UniWalk are demonstrated through experimental results, positioning it as a promising method for addressing the challenges of recommendation systems in leveraging heterogeneous data sources for enhanced user experiences and engagement. RippleNet [124] is a framework that integrates the KG into recommender systems, introducing preference propagation to explore users' interests. It addresses the limitations of existing methods by unifying preference propagation and regularization of KG embedding. It explains recommendation results by tracking paths from a user's history to high-relevance items, improving acceptance and satisfaction with recommendations and increasing trust in the recommender system. The study also explores the correlation between common neighbours and relatedness in the recommender system. Another proposed novel approach, CFKG [4], for personalized and explainable recommendation systems, uses KG embeddings to integrate structured knowledge into the recommendation process. The model learns user and item representations while preserving their relationship with external knowledge, such as textual reviews, visual images, and feedback. It extends traditional collaborative filtering to learn over heterogeneous knowledge for recommendation, capturing user preferences more comprehensively. The model also incorporates a soft matching algorithm to construct explanations based on the embedded KG emphasizing the importance of model-generated explanations in recommender

systems to enhance user experience. Explainable Interaction-driven User Modeling (EIUM) [58] uses KG to create an effective and explainable sequential recommender. It captures users' dynamic interests accurately and provides path-wise explanations for the recommendation system. It models sequential interactions and employs multi-modal fusion by incorporating textural, visual, and structural knowledge into the network, where the different modal features satisfy the constraints of the structural information of entities and relations in KG. This leads to better representation learning offering highly explainable results, and enhancing transparency and trustworthiness in recommendation systems. The Dynamically Fused Graph Network (DFGN) [95] is a text-based model that constructs a dynamic entity graph based on entity mentions in a query and documents through a multi-step process. The model iteratively constructs the graph in multiple rounds, generating and reasoning on it, masking out irrelevant entities and preserving reasoning sources. It employs a fusion process to filter out noise and extract useful information, aggregating information from documents to the entity graph (doc2graph) and propagating it back to document (graph2doc) representations. This results in a less noisy entity graph and more accurate answers. The model also introduces a mask prediction module to alleviate error propagation problems and proposes a feasible way to weakly supervise mask learning. The Policy-Guided Path Reasoning (PGPR) [134] method uses a multi-hop scoring function to capture complex relationships and reasoning paths within the KG. This approach allows for efficient sampling of reasoning paths by considering k-hop patterns and 1-reverse k-hop patterns. The scoring function evaluates the relevance and significance of potential reasoning paths, filtering appropriate actions based on the starting user. It is integrated into the objective function for training the KG representation, enhancing the model's effectiveness and providing interpretable evidence for each recommendation.

The Knowledge-aware Path Recurrent Network (KPRN) [128] is a novel model that uses Knowledge Graphs (KG) to improve recommender systems. It generates path representations by combining entity and relation semantics, enabling effective reasoning on paths and inferring user-item interactions. The model also introduces a weighted pooling operation, to discriminate the strengths of different paths for better explainability. Experiments on movie and music datasets show significant improvements over existing solutions like Collaborative Knowledge Base Embedding and Neural Factorization Machines. CAFE [135], the CoArse-to-FinE neural symbolic reasoning method enhances e-commerce recommendation systems by incorporating KGs for explainable recommendations. It involves two stages: the coarse stage, which captures user behaviour, and the fine stage, which uses path reasoning guided by user profiles. The proposed Profile-guided Path Reasoning algorithm efficiently conducts batch path reasoning, resulting in substantial improvements in recommendation performance. This approach also outperforms other methods, including randomly sampled and globally assigned profiles, in terms of recommendation performance.

A joint learning framework DKER [152] is proposed that combines the embedding-based and path-based recommendation models to achieve explainable and accurate recommendations. The model trains the embedding-based model and differentiable path-based model with a mutual regularization term in the objective function, allowing the embedding-based model to learn from observed user-item interactions and augmented pseudo-labels. This enhances the expressive power of embedding-based models while preserving interpretability. Furthermore, the loss function controls the balance between imitation learning from concrete training labels and knowledge distillation. Another explainable sequential path-based model ELPE [16], proposes a framework for inductive representation learning and link prediction by offering a solution for evolving KGs with emerging entities. It introduces a variant of the Graph Transformer encoder for inductive node representation learning leveraging the paths from a KG to make the inference process explainable. It also uses policy gradient-based reinforcement learning to decode a reasoning path to the answer entity, providing prior semantic knowledge. The key contributions of this model include learning representations for unseen emerging entities during inference and predicting missing links between emerging entities and pre-existing

entities in the KG. Another recommender system, LOGER [155] comprises three components: a KG encoder, which learns embeddings of KG entities and relations followed by a neural logic model, which conducts interpretable logical reasoning to make recommendations. The KG encoder maps triplets to real-valued scores, while the neural logic model emits personalized rule importance scores to capture user behaviour and optimize logical reasoning. The third component, a rule-guided path reasoner ensures better recommendation performance by generating explainable paths to relevant items. A KG reasoning-based approach, EXACTA [136] aims to achieve explainable column annotation in industrial digital marketing data pipelines. It uses TransE embeddings [17], OpenKE [55], and Adam optimization to initialize state and action representations, and iteratively traverses the KG to generate reasoning paths for predictions. The model learns a noise-tolerant reward function from potentially less explainable paths, guiding policy learning to produce high-quality column annotations. Another novel method for a knowledge-aware recommendation approach, EKAR [115], generates meaningful paths by considering the complete path history as the current state and leads users to pertinent items in the integrated user-item-entity network. It uses deep reinforcement learning and a reward function based on existing KG representation learning methods. EKAR is a useful technique for recommendation systems since it generates paths that give concise explanations. The PathReasoner [148] framework combines external reasoning paths with structured information to improve commonsense question answering. It consists of a pathfinder module and a hierarchical path learner, which extract key entities from questions and align them with large-scale KGs. The learner uses an intra-path encoder to encode each path with the question and an inter-path encoder for soft selection. The KR-GCN model [81] improves error propagation and explainability by using transition-based methods to score triplets within multi-hop paths, integrating user-item interactions and KGs using a GCN.

A concise overview of the explanation techniques encompassed within this category is presented in Table 2.

Limitations. The sequential path-based models' performance and interpretability are limited due to their inability to capture complex interactions, long-range dependencies, and exploration of alternative paths. These models are vulnerable to adversarial attacks and face challenges in balancing complexity and interpretability. Scalability issues arise as the graph size increases, making it difficult to efficiently explore sequential paths and provide timely explanations. The sensitivity of the models to path length and the computational cost of path extraction can impact performance and interoperability. Additionally, also struggles with noisy or uncertain graph data and lacks support for incremental learning. The models face challenges in handling graphs with multiple types of relationships, hierarchical structures, domain knowledge, multi-modal graph data, interactive exploration, and multi-scale graph structures. These challenges can affect the reliability and accuracy of the model's explanations and the ability to adapt to new data. In the case of dynamic graphs, these models often face concept drift, where underlying relationships and patterns in graph data may change over time, necessitating the model's adaptation.

4.2.2 ***Logic Rules.*** Despite being proposed as an explainer for tabular data, LORE [51] (LOcal Rule-based Explanations) defines a surrogate graph-based model for generating the explanation. Given a binary predictor and a specific instance labeled as y, LORE builds a simple, interpretable predictor by generating a set of neighbor instances through a genetic algorithm and extracting a surrogate decision tree classifier. Subsequently, a local explanation is then extracted from the decision tree as (i) a logic rule, corresponding to the path in the tree that explains why the instance is labeled as y, and (ii) a set of counterfactual rules [1]. However, it only works under the assumption that in the neighborhood of a given data point, the decision boundary is clear and simple to be captured by a simple but interpretable surrogate model.

[1]Alterations that need to be made to the input sample to invert the class assigned by the classifier.

Table 2. A comparison table for path explainers where Qn and Ql denote quantitative and qualitative validation, respectively. The models mentioned here leverage graph features, namely paths, in different types of graphs such as KGs, and are not exclusively designed for GNNs. TCA: table column annotation, QA: question answering.

Method	Approach	GNN based	Task	Validation		Source Code
				Data	Quality	
Dedalo [120]	Path Reasoning	✗	Clustering	Real	Qn	-
UniWalk [91]	Data Integration	✗	recommendation	Real	Ql	✓
RippleNet [124]	Path Reasoning	✗	recommendation	Real	Ql	✓
CFKG [4]	Path Reasoning	✗	recommendation	Real	Ql	-
EIUM [58]	Path Reasoning	✗	recommendation	Real	Ql	-
DFGN [95]	Path Reasoning	✗	QA	Real	Ql	✓
KPRN [128]	Path Reasoning	✗	recommendation	Real	Ql	-
PGPR [134]	Path Reasoning	✗	recommendation	Real	Ql	✓
CAFE [135]	Path Reasoning	✗	recommendation	Real	Ql	✓
DKER [152]	Path Reasoning	✗	recommendation	Real	Ql	-
ELPE [16]	Path Reasoning	✗	KG Completion	Real	Ql	✓
LOGER [155]	Path Reasoning	✗	recommendation	Real	Qn	✓
EXACTA [136]	Path Reasoning	✗	TCA	Real	Qn	-
Ekar [115]	Path Reasoning	✗	recommendation	Real	Ql	-
PathReasoner [148]	Path Reasoning	✗	QA	Real	Ql	-
KR-GCN [81]	Path Reasoning	✓	recommendation	Real	Qn/Ql	-

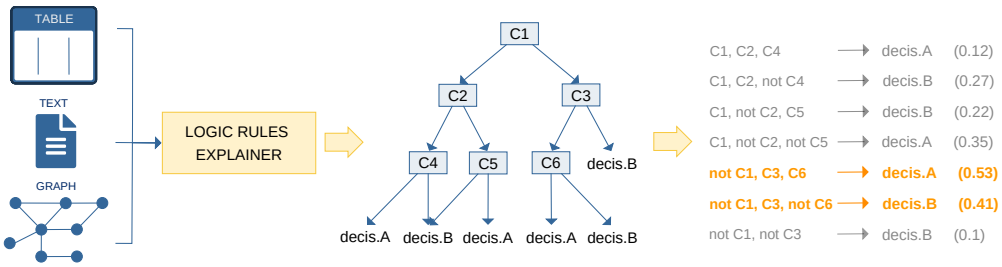


Fig. 5. A logic rules-based explainer uses the input data, such as text, tabular data, or graphs, to generate a decision tree. This tree assesses multiple conditions (Ci) to determine the final decision (decis.i). By combining conditions, the explainer evaluates the quality of the derived rules in explaining the prediction. In this example, the final explanations are highlighted in orange.

In recommender systems, embedding-based methods are acknowledged for their strong generalization but a notable deficiency in explainability, while tree-based methods offer explainability but struggle with unseen feature interactions. As an effort to overcome these limitations, the Tree-enhanced Embedding Model (TEM) [126] combines gradient-boosting decision trees (GBDT) with an embedding model. This fusion allows TEM to derive explicit decision rules from user and item data and generalize to unseen interactions using an attention network. Despite reporting good results, TEM explanations consist of simple attention scores over sampled decision rules, only serving as indicators of their respective contributions to the final prediction. RuleRec [82] combines a rule learning module and a recommendation module. It integrates explainable rules extracted from item-centric KGs to summarize common multi-hop relational patterns for inferring different item associations. These rules are integrated into the recommendation module to enhance generalization and address the cold-start[2] problem in recommender systems [42, 69, 88].

[2]A cold start problem occurs when the recommender system cannot make recommendations for users or items about which it has not yet collected enough information.

Table 3. A comparison table for logic rules-based explainers. None of them were explicitly tailored for GNN models. Qn and Ql stand for quantitative and qualitative validation, respectively.

Method	Approach	GNN based	Task	Validation		Source Code
				Data	Quality	
LORE [51]	Surrogate	✗	Classification	Real	Qn/Ql	-
TEM [126]	Surrogate	✗	recommendation	Real	Ql	✓
RuleRec [82]	Data Integration	✗	recommendation	Real	Ql	✓
ExCut [44]	Data Integration	✗	Clustering	Real	Qn/Ql	✓
ExCAR [33]	Data Integration	✗	Causal Reasoning	Real	Qn/Ql	✓

ExCut [44] combines KG embeddings with rule-mining methods to compute informative entity clusters with clear explanations. Initially, it utilizes KG embedding to identify plausible entity groups. Then, it applies logical rule mining on the entity associations to learn interpretable cluster labels, which correspond to rules formed by conjunctions of relations that characterize the most entities in a given cluster. These labels guide the iterative adaptation of entity embeddings and clustering in subsequent iterations. The authors define two metrics to evaluate the quality of the extracted logic rules: *per cluster coverage (Cov)* and *exclusive coverage (Exc)*. These metrics prioritize rules covering more entities within a cluster and ensure that explanation rules for different clusters are mostly mutually exclusive.

ExCAR [33] leverages KG to elucidate causal relationships between events. Given an event pair, ExCAR employs an evidence retrieval module to retrieve external evidence events from a prebuilt causal event graph to generate a set of logical rules. Then ExCAR conducts causal reasoning based on the obtained logical rules using a Conditional Markov Neural Logic Network. Despite reporting very good explanation results and incorporating experts to evaluate the causality strengths, ExCAR depends on the lexical overlapping between mentions and entries in a pre-existing KG. Therefore, it is inapplicable to other domains or free-form texts.

Table 3 presents a summary of the strategies centred on extracting logical rules for explaining graphs.

Limitations. Logic rules-based explainers represent easily applicable and comprehensible techniques, devoid of the need for specialized expertise. Hence, their versatility extends across various domains and tasks. Nevertheless, the explanations they yield tend to be broad and simplistic, constituting mere concatenations of the recurrent patterns and elements present in the input data. Furthermore, practitioners must engage in manual decision-making processes involving the definition of parameters when utilizing these explainer types. This encompasses decisions related to the number of iterations, the number of neighbor instances, or the number of trees for establishing the corresponding surrogate interpretable model, among other considerations.

4.2.3 ***Subgraph.*** Methods in this category identify key elements within the input graph aligned to the predicted label, aiming to generate an explanation subgraph, as Figure 6 shows. Among these, a substantial portion of explainers rely on varying input perturbation techniques on the elements of the structure to quantify the relevance of a graph component.

In link prediction, Explaining Network Embeddings (ExplaiNE) [65] quantifies how weakening or removing a link i, k affects the probability of a predicted link i, j. Those links that most strongly reduce this probability serve as counterfactual explanations. Despite its novelty, ExplaiNE was diminished by GNNExplainer [143], which can also explain graph classification (GC) and node classification (NC). GNNExplainer identifies the subgraph and relevant node features that maximize the mutual information (MI) gain of the ground truth and predicted label using graph and feature masks. However, it suffers from the introduced evidence [26] problem, as its soft masks can introduce noise, affecting explanation results. In addition, the optimization of the model is intense.

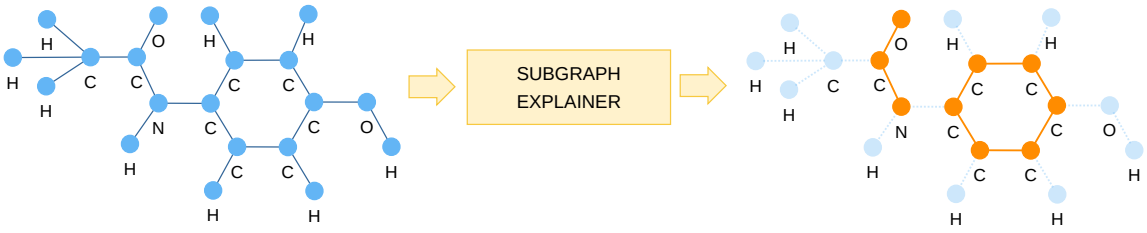


Fig. 6. A paracetamol molecule comprising carbon (C), nitrogen (N), oxygen (O), and hydrogen (H) atoms, alongside a subgraph explanation example (orange) of its predicted energy. The rest of the graph is included only for reference, as it is not part of the final explanation. Example based on the results reported in [105].

GraphMask [104] applies a parameterized deletion function on node representations to decide edge retention at each GNN layer. This approach mitigates the *hindsight bias*, where important edges might be pruned because a similar prediction can be achieved using a smaller subgraph, potentially leading to overfitting. Contrastive GNN Explanation (CoGE) [37] identifies recurring patterns without model training. It defines weights for each node on other graphs from available training data to find those components that differentiate the input graph from others with different labels, while aligning it with similar ones. Though simple and interpretable, CoGE is only effective for graph classification.

PGExplainer [80] employs a generative probabilistic model to represent graph structures as edge distributions. It approximates discrete edge masks via a re-parameterization trick, allowing it to explain new data without re-training the GNN and mitigating the introduced evidence problem. Similarly, PGM-Explainer [123] uses conditional probabilities to assess graph components' contributions by randomly perturbing node features. The top dependent variables are then used to fit a surrogate Bayesian network [41] and produce the final explanations. However, PGM-Explainer does not scale well and ignores graph edges, missing topological information. ReFine [129] integrates contrastive learning with class-specific generative probabilistic models. It defines a pre-training phase to extract global class patterns and a fine-tuning phase for local explainability, eliminating the need for retraining for each new graph. While effective for encoding class-specific knowledge, ReFine struggles to map such knowledge into a graph representation.

Causal Screening [130] adopts a causal perspective, starting with an empty set and incrementally adding edges by evaluating conditional MI [8] between the candidate edge and the original prediction, conditioned on previously selected edges. This method requires no training, avoiding introduced evidence issues, but only focuses on structure, neglecting nodes and features. Gem [75] employs Granger causality [49] and various graph rules to distill compact subgraphs, but limits the model to a fixed number of nodes, restricting the explanation to a selection of components.

Based on Reinforcement Learning, the Molecular Explanation Generator (MEG) [90] identifies graph modifications that maximally change the model's prediction, but its application is limited to molecular contexts due to its heavy reliance on domain-specific knowledge. Conversely, RG-Explainer [110] generates explanations from scratch by starting with a seed node and iteratively adding neighboring nodes based on the MI between the original and the explanation graph labels. A learned stopping criterion prevents overly large explanations. Self-Explainable GNN (SE-GNN) [27] finds the k-nearest labeled nodes for each unlabeled node, using them for both label prediction and explanation. It adopts a contrastive learning-based similarity module for self-supervision of node embedding and local n-hop graph structure similarity. While eliminating the need for a separate explanation generation step, It is limited to node classification.

Contextualized Multilevel Feature (CMF) [30] utilizes news articles and events for explainable event forecasting. It includes a predictor that models multilevel contextualized features within a hierarchical graph and processes temporal

data, along with an MI-based interpreter to identify key features at different levels (documents and graph components). While CMF provides comprehensive explanations, its effectiveness depends on accurately modeling news and event data, which may be local and overlook the broader context. CMGE [133] explains clinical diagnoses from medical records by organizing report text into a hierarchical graph. It establishes causal relationships between multi-granular features and diagnosis through counterfactual interventions on the graph. It utilizes a Graph Attention Network (GAT) [122] to mask nodes or edges while preserving the diagnosis. However, the underlying model is not interpretable. Robust Counterfactual Explanations (RCExplainer) [11] models GNN decision logic via decision regions, using a 2-layer feed-forward (FF) neural network to identify edges whose removal reduces prediction confidence. For node classification, RelEx [151] randomly samples connected subgraphs in a breadth-first search from the target node's computational n-hop neighborhood. It uses a Graph Convolutional Network (GCN) [67] as a surrogate model to generate explanations. Despite not being GNN-specific, RelEx uses a GCN as an auxiliary model, lacking interpretability like FF.

SubGraphX [147] claims that prominent graph elements may not form connected structures, reducing human comprehensibility. It uses Monte Carlo Tree Search (MCTS) [112] to explore subgraphs of a given graph G, measuring their importance with Shapley values [68]. Then, the highest-scoring subgraph is prioritized as the explanation. However, high computational costs limit its use for large-scale graphs. Likewise, SubGraphX selects only one subgraph, being unable to cover two or more important node groups. GraphSVX [34] captures node and feature contributions toward the explained prediction using binary masks derived from graph perturbations. It employs a surrogate weighted linear regression model and uses Shapley values to attribute the contributions. Arguing that Shapley values lack structure awareness, Graph Structure-aware eXplanation (GStarX) [150] uses the Hamiache-Navarro (HN) value [54], which considers interactions among neighboring nodes through an iterative aggregation algorithm. Nevertheless, due to the complexity of comparing all node and feature combinations, the method becomes impractical for large graphs, necessitating auxiliary approximations and sampling techniques.

Adopting a causal inference perspective, CounterFactual and Factual reasoning (CF2) [119] evaluates the necessity and sufficiency of explanations by combining factual and counterfactual reasoning. The authors point out that factual explanations, like [80, 143], may be sufficient for prediction but often include redundant components. In contrast, counterfactual explanations contain only crucial information that, if removed, changes the prediction. CF2 balances these views by learning edge and node feature masks and evaluating explanation strength (effectiveness) and complexity (number of edges and features) by introducing the *probability of necessity* and *probability of sufficiency* for quantitative evaluation of the explanations. Based on rate-distortion theory [114], ZORRO [43] identifies essential nodes and features for sparse, valid, and stable explanations. Using discrete masks via graph perturbation, ZORRO iteratively selects graph elements to maximize the fidelity score, reflecting the GNN's ability to reconstruct the prediction from the explanation. ZORRO requires no additional training, but its effectiveness relies on manually defined hyperparameters. TraP2 [63] uses a three-layer framework: Translation, Perturbation, and Paraphrase. The translation layer limits the interpretable region to the subgraph of n-hop neighbors around the target node. The perturbation layer modifies the graph structure and node features using action variables from Bernoulli and Normal distributions. Then, the perturbed graphs are converted into vectors and fed into a classifier that mimics the GNN's responses. However, TraP2 may become inefficient for graph classification, as it must process each node to determine individual contributions.

GraphLIME [57] adapts LIME [101] for GNNs using a surrogate kernel-based nonlinear feature selection algorithm, HSIC Lasso [50, 142]. For a target node, the n-hop neighboring nodes and their predictions are used to fit the HSIC Lasso model. Then, the learned coefficients determine the relevance of the target node's features. However, GraphLIME disregards graph structure and is limited to only node-level tasks. MotifExplainer [144] explains GNN predictions by

Table 4. A comparison table of subgraph extraction explainers. Qn and Ql stand for quantitative and qualitative validation, respectively.

Method	Approach	GNN based	Tasks			Validation		Source
			GC	NC	Other	Data	Quality	Code
ExplaiNE [65]	Perturbation	✗	-	-	Link Prediction	Real	Qn/Ql	-
GNNExplainer [143]	Perturbation	✓	✓	✓	Link Prediction	Real/Synth.	Qn/Ql	✓
GraphMask [104]	Perturbation	✓	✓	✓	-	Real/Synth.	Qn	✓
CoGE [37]	Data Integration	✓	✓	-	-	Real/Synth.	Qn/Ql	✓
PGExplainer [80]	Perturbation	✓	✓	✓	-	Real/Synth.	Qn/Ql	✓
PGM-Explainer [123]	Surrogate	✓	✓	✓	-	Real/Synth.	Qn	✓
ReFine [129]	Perturbation	✓	✓	-	-	Real/Synth.	Qn/Ql	✓
Causal Screening [130]	Perturbation	✓	✓	✓	-	Real	Qn/Ql	-
Gem [75]	Perturbation	✓	✓	✓	-	Real/Synth.	Qn/Ql	✓
MEG [90]	Perturbation	✓	✓	-	Regression	Real	Qn/Ql	-
RG-Explainer [110]	Graph Creation	✓	✓	✓	-	Real/Synth.	Qn/Ql	-
SE-GNN [27]	Data Integration	✓	-	✓	-	Real/Synth.	Qn	✓
CMF [30]	Perturbation	✓	-	-	Event Forecasting	Real	Qn/Ql	-
CMGE [133]	Perturbation	✓	✓	✓	-	Real	Qn	✓
RCExplainer [11]	Surrogate	✓	✓	✓	-	Real/Synth.	Qn	-
RelEx [151]	Surrogate	✗	-	✓	-	Real/Synth.	Qn/Ql	-
SubGraphX [147]	Perturbation	✓	✓	✓	Link Prediction	Real/Synth.	Qn/Ql	✓
GraphSVX [34]	Perturbation	✓	✓	✓	-	Real/Synth.	Qn/Ql	✓
GStarX [150]	Perturbation	✓	✓	✓	-	Real/Synth.	Qn/Ql	✓
CF2 [119]	Perturbation	✓	✓	✓	-	Real/Synth.	Qn/Ql	-
ZORRO [43]	Perturbation	✓	✓	✓	-	Real/Synth.	Qn/Ql	✓
TraP2 [63]	Perturbation	✓	✓	✓	-	Real/Synth.	Qn/Ql	✓
GraphLime [57]	Surrogate	✓	-	✓	-	Real	Qn	-
MotifExplainer [144]	Data Integration	✓	✓	✓	-	Real/Synth.	Qn/Ql	-
TAGE [138]	Perturbation	✓	✓	✓	-	Real/Synth.	Qn/Ql	✓
EGIB [125]	Perturbation	✓	✓	✓	-	Real	Qn/Ql	-

extracting motifs using predefined rules based on domain knowledge. It generates motif embeddings for each of them and uses an attention mechanism to identify the most influential ones. This approach efficiently reduces the search space by limiting the number of extracted motifs, enhancing performance on large-scale graphs. However, effective motif extraction rules are necessary for its effectiveness.

Recently, TAGE [138] and EGIB [125] proposed task-agnostic solutions for explaining GNNs trained under self-supervision, decomposing the prediction model into a two-stage pipeline without requiring knowledge of downstream tasks. Task-Agnostic GNN Explainer (TAGE) [138] comprises an embedding model and an embedding explainer, followed by downstream models and their respective explainers. The embedding explainer maximizes the MI between the input graph and the explanation subgraph in a self-supervised manner, using a masking vector sampled from a multivariate Laplacian distribution. It can collaborate with any downstream explainer for end-to-end explanations by substituting the masking vector entries. Efficient computation is achieved using Jenson Shannon [89] and InfoNCE [52] estimators.

Explainable Graph Information Bottleneck (EGIB) [125] employs a pretraining strategy for a task-agnostic explainer within the representation space of the target GNN, with optional fine-tuning for task-specific information. Using an information bottleneck paradigm [6], EGIB generates sufficient and compact explanation subgraphs by maximizing the MI between the intended explanation subgraph and the GNN representations, subject to ϵ-explanatory constraints. As in [138], EGIB also employs the InfoNCE [52] estimator and a Lagrangian relaxation algorithm for optimization purposes. A summary of the subgraph extraction explainers is presented in Table 4.

Limitations. Explanatory methodologies within this category demonstrate advantages solely in domains where practitioners seek to identify the underlying substructures influencing specific predictions, such as those encountered in domains like chemistry, biology, image analysis, or social network analysis. These approaches necessitate substantial computational resources, leading to widely adopting approximation techniques and other optimization methodologies. Since the search space for the relevant subgraphs is restricted by the graph's size, the usefulness of explainers based on subgraphs in real-world situations is diminished, given the presence of considerably larger and more intricate structures.

Despite operating on graph structures and possessing the ability to access the constituent elements' features, none of these methods comprehensively integrate all three levels of components into their explanations. Certain proposals focus solely on structural aspects, encompassing nodes and edges while disregarding node features. Conversely, others prioritize nodes and their attributes but neglect the information provided by edges. Moreover, various perturbation-based explainers adopt hard masks to generate meaningful and unambiguous explanations for end users. Nevertheless, this approach may prove overly simplistic for certain domains and tasks, resulting in the loss of valuable information from discarded sources. The combination of factual and counterfactual approaches for obtaining sufficient and minimal explanation subgraphs represents a promising avenue. Nonetheless, these methods continue to face efficiency limitations.

4.3 Graph Generation

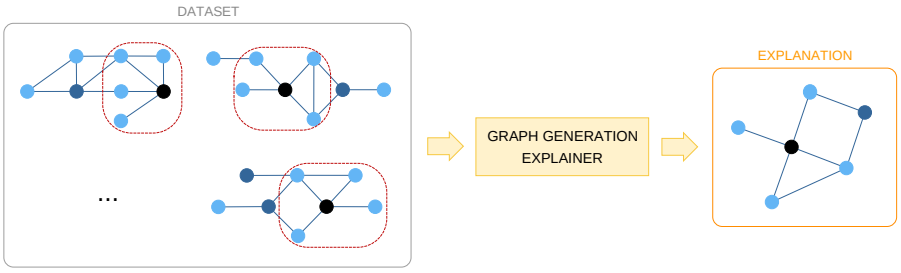


Fig. 7. An example of a graph generation-based explainer. The shared patterns among all input graphs within the dataset are highlighted to facilitate comprehension.

Generating an explanation graph from scratch using training data is complex due to the challenge of associating topological information with the learning model. Thus, this area has been little explored, as indicated in Table 5.

Cognitive Graph QA (CogQA) [31] employs an iterative approach to build a cognitively inspired graph. It considers an extractor that identifies question-relevant entities and answer candidates from paragraphs, encoding their semantic information and organizing them as a graph. Then, a reasoning module, powered by a GNN model, conducts reasoning over the graph, gathering clues to iteratively enhance the extraction of entities until all possible answers are found. The final answer is chosen based on the GNN results. Although intuitive, CogQA expects the reader to find the reasoning paths from the root to the generated answer.

Subsequently, XGNN [145] emerged as one of the initial techniques for explaining GNNs at the model level. Unlike approaches that focus on elucidating individual data instances, XGNN elucidates the learning accomplished by the underlying model. It investigates which input graph patterns can induce specific behaviors in the model being explained. To achieve this, XGNN employs Reinforcement Learning (RL) to train a graph generator to produce a structure that maximizes the model's predictions. During each iteration, the generator predicts where to add a new edge to the existing

Table 5. A comparison table for generation-based explainers.

Method	Approach	GNN based	Tasks			Validation		Source Code
			GC	NC	Other	Data	Quality	
CogQA [31]	Graph Creation	✓	-	-	QA	Real	Ql	✓
XGNN [145]	Graph Creation	✓	✓	-	-	Real/Synth.	Qn/Ql	-
ExplaGraph [102]	Graph Creation	✗	-	-	Stance Detection	Real	Qn	✓

graph structure. The impact of this addition is evaluated through a forward pass in the trained GNN model, adapting the corresponding generator component based on gradients. Thus, the generated graph encapsulates the discerning patterns deemed explanations for the target prediction. However, it is crucial to note that expert knowledge is essential for maintaining the validity and comprehensibility of the generated graph to human interpreters. Additionally, it is worth mentioning that the effectiveness of XGNN has been tested solely on graph classification tasks.

Recently, ExplaGraph [102] was also proposed. The authors collect instances of belief on Amazon, their arguments, and their corresponding stances to learn the corresponding commonsense explanation graphs for the stance detection problem. The corresponding commonsense explanation graphs are collected by writing multiple facts which are then verified and refined to be used as ground truth. The authors compare BART and T5-based models as liberalized graph generation strategies by measuring the structural and semantic correctness of the generated graphs. Despite following an intuitive graph construction, the authors reported a large gap concerning human performance.

Limitations. Balancing simplicity for interpretability and accurately capturing the global model's behavior nuances, as well as the overall decision-making process, proves more challenging when compared to concentrating on local facets for explaining individual instances. Furthermore, generation explainers might fail to preserve the local fidelity of the generated graphs, i.e., there is no guarantee that the generated explanation graph follows the graph structure of the previously encountered training data. Moreover, generating graphs for large datasets or intricate models can be computationally expensive and may suffer from scalability issues, constraining the practical applicability of generation explainers in real-world scenarios.

5 ON THE QUALITY OF EXPLAINABILITY

This section introduces the graph-based datasets commonly utilized in prior studies. We categorize these datasets as (i) synthetic datasets, which consist of ground truth, and (ii) real-world datasets that lack gold standards. Furthermore, we outline the metrics used to assess the quality of the explanation quantitatively.

5.1 Datasets

To evaluate graph-based explanation methods under controlled environments, multiple synthetic datasets have been introduced. These datasets are crafted using specialized techniques for graph generation and validated by expert knowledge. They incorporate distinct structural connection rules, allowing the establishment of ground truth explanations for the enclosed graphs. This allows for a predetermined understanding of the graphs, facilitating a direct evaluation of explanation quality concerning the assessed models. Conversely, real-world data collections typically feature larger-scale graphs where ground truth explanations are unknown. Only a small number of real datasets possess gold standards, which are only obtained through validation by domain experts.

Synthetic Data. Synthetic datasets typically consist of graphs of limited size and simple configurations, facilitating the unambiguous identification of pertinent graph elements relevant to the corresponding sample prediction. Given that these datasets are fashioned as a toy set for validation of explainers, the majority can be fabricated on-site using the practitioner's equipment by adhering to the construction guidelines of the dataset authors or employing the commonly disseminated generation code provided by them, as detailed in Table 6.

Various researchers used Barabasi Albert (BA) structures as the foundation for their graph datasets. `BA-Growth` constitutes a synthetic dataset with two classes, each characterized by distinct configurations of BA graphs. `BA-2Motifs` dataset is constructed by appending BA graphs with house motifs and five-node cycle motifs. An extended version, `BA-3Motifs`, incorporates grid graphs as additional motifs for graph attachment. `BA-SHAPES` involves attaching house-like structures to randomly chosen nodes within BA base graphs. Subsequently, nodes are categorized based on their structural role as nodes within the houses, including top, middle, bottom, and non-house-affiliated nodes. An expansion of this approach, `BA-COMMUNITY`, concatenates two `BA-SHAPES` graphs to represent a unified community structure.

Tree structures have also seen extensive use as the basis for synthetic graphs. In `TREE-CYCLES`, a balanced binary tree serves as the core structure, complemented by incorporating 80 six-node cycle motifs attached to randomly selected nodes. Conversely, grid motifs are attached to the base tree graph in `TREE-GRID`. In `TREE-BA` multiple BA nodes are connected to a binary tree. By randomly aggregating edges between BA and basic nodes, the prediction involves predicting the correct class to which each node belongs.

Alternative structures have been explored. The `Is Acyclic` dataset categorizes graphs based on the presence of cycles, while the Cycles and Cliques (`CYCLIQ`) dataset comprises random trees with additional cycles or cliques. The `Star Graph` dataset focuses on star topologies, where each edge is assigned one of multiple colors. The goal is to predict if the count of edges assigned to color a exceeds that of edges assigned to color b.

To imitate the dynamics of real-world graphs, the `Infection` dataset is constructed as a graph where nodes represent individuals who can be sick, healthy, or immune to spreading disease. Edges between nodes represent relationships, categorized as virtual or not. The objective is to predict the state of every node after one step of the spread. Synthesized from the Cora graph, `Syn-Cora` samples local graphs of nodes from the original dataset and applies alterations to attributes and structures, thus producing similar local graphs. Subsequently, a subgraph of Cora is sampled as the basis graph, to which the synthetic local graphs are appended by connecting three nodes at random.

Although providing golden explanations to validate the effectiveness of the explainer, these datasets are constrained to a narrow spectrum of tasks, encompassing graph classification or node classification.

Real Data. Real graph-based datasets encompass multiple tasks and domains, as outlined in Table 7.

Among the classification datasets, `MUTAG` is a widely known benchmark for categorizing molecule graphs by their mutagenic effect on bacteria. Similarly, the Proteins dataset distinguishes enzymatic from non-enzymatic protein graphs. Collections of chemical compounds, such as PTC and NCI1, report carcinogenicity for rats, and cell activity against lung cancer, respectively. The `MoleculeNet` library offers datasets like `HIV`, `SIDER`, `BACE`, `BBBP`, `TOX21`, `ESOL`, and `QM9`, enabling the prediction of various molecular properties, with `ESOL` and `QM9` addressing the task through regression.

`REDDIT-BINARY` comprises 2,000 graphs depicting online discussion threads on Reddit, where nodes represent users and edges denote replies. Graphs are labeled based on user interactions. `REDDIT-MULTI 5K`[3] extends this by labeling graphs according to community topics. `IMDB BINARY` is a dataset on movie collaborations, with nodes representing actors and directors and edges indicating shared appearances.

[3] https://networkrepository.com/REDDIT-MULTI-5K.php

Table 6. Previously employed graph-based synthetic datasets. (★) indicates that although the dataset is not publicly available, its generation is supported by PyTorch Geometric.

Dataset	Task	Base Structure	Publicly Available
BA-Growth	GC	Barabasi Albert	(★) ExplainerDataset
BA-2Motifs	GC	Barabasi Albert	PyTorch Datasets
BA-3Motifs	GC	Barabasi Albert	GitHub
BA-SHAPES	NC	Barabasi Albert	PyTorch Datasets
BA-COMMUNITY	NC	Barabasi Albert	GitHub
TREE-CYCLES	NC	Tree	GitHub
TREE-GRID	NC	Tree	GitHub
TREE-BA	NC	Tree	(★) ExplainerDataset
Is Acyclic	GC	Cycle	(★) ExplainerDataset
CYCLIQ	GC	Cycle & Clique	GitHub
Star Graphs	GC	Star	GitHub
Infection Toy dataset	NC	Barabasi Albert	PyTorch Datasets
Syn-Cora	NC	Real data	GitHub

Graphs are also utilized in other data modalities. Visual Genome pairs images with scene graphs, where objects are delineated by bounding boxes, while MNIST SuperPixel-Graph converts original MNIST images into graphs, with each node representing a superpixel and including intensity and location as features. Conversely, the Stanford Sentiment Treebank (SST) dataset employs graphs to depict syntactic trees of sentences, with nodes representing words. Graph-SST2[4] and Graph-Twitter are sentiment graph datasets where nodes denote words and edges represent the relationships between them. Graph labels are positive, negative, or neutral sentiments.

Specifically for node classification, the Protein-Protein Interaction (PPI) dataset records physical interactions among proteins, with each protein representing a node and edges indicating interactions. Bitcoin-OTC and Bitcoin-Alpha datasets reflect accounts trading Bitcoin on Bitcoin-OTC and Bitcoin-Alpha, respectively. Members rate each other's trustworthiness. Three citation graph datasets, namely Cora, CiteSeer, and PubMed, involve articles as nodes and citations as edges, focusing on document classification.

For link prediction, the DBLP dataset represents a co-authorship network with papers and authors as nodes, the Karate Club dataset represents a social network with 34 members as nodes and 78 friendship links, and the Game of Thrones dataset features novel characters as nodes, linked if mentioned within 15 words of each other. Predicting missing relationships or facts in a KG has also been explored in subsets like FB15k-237, WN18RR, and NELL-995 stemmed from FB15k, WN18, and NELL KG, respectively.

For recommender system deployment, Epinions is a who-trusts-whom online social network of general consumer reviews, Amazon provides item association data spanning various domains like CDs, Vinyl, Clothing, Cellphones, and Beauty, and Yelp, FilmTrust, Flixster, Douban, MovieLens (including MovieLens-1M[5] and MovieLens-20M[6] extensions), and MI (MovieLens-1M combined with IMDb) represent rating networks. Last FM and KKBox offer user-item interaction data in the music domain, while Pinterest captures user-image interactions. Alternatively, interconnected web pages (WebKB) and the interaction networks between entities and resources (UW-CSE, Yago Artwork) along different domains have also served for entity clustering.

Other specialized collections serve different purposes. The Breast Cancer Patient dataset comprises over 12,000 genes across 969 patients, aiding in graph signal prediction. In Natural Language Processing, English CoNLL-2009

[4] https://github.com/divelab/DIG/tree/main/dig/xgraph/datasets
[5] https://pytorch-geometric.readthedocs.io/en/latest/modules/datasets.html
[6] https://grouplens.org/datasets/movielens/20m/

features sentence predicates as dependency parse trees for Semantic Role Labeling (SRL), and EXPLAGRAPHS [102] focuses on stance detection, providing commonsense explanation graphs for beliefs and arguments on various topics.

Numerous graph-based models can also be applied to raw text and tabular data, extending the applicability of such models to other tasks. To do so, the original data is transformed into a graph structure by identifying relevant facts and entities through cross-referencing external KG [124, 148], defining decision trees based on data similarities and common patterns [51, 126], and others [95, 133]. Some text-based datasets include HotPotQA[7], CommonsenseQA[8], and WIQA[9].

5.2 Metrics

Assessing the quality of explanations generated by a graph-based explainer poses a challenge. Currently, there is no widely accepted standard for this evaluation, resulting in extensive discourse and the proposal of various evaluation metrics, often tailored to specific domains, limiting their applicability across diverse environments and tasks. Moreover, real-world datasets lack gold explanations, introducing inconsistency in the quantitative validation of explanation methods when applied to synthetic versus real data. Synthetic datasets offer clear ground truth explanations, facilitating straightforward measurement of accuracy, precision, and recall, without significant hurdles.

In the absence of a gold standard explanation, a common practice is qualitative analysis, scrutinizing individual use cases. However, this method can introduce subjectivity and hinder the generalization of findings across different samples or conditions. While qualitative analysis can offer valuable insights, it should be viewed as supplementary and accompanied by clear explanation metrics, such as:

- *Accuracy*: It evaluates the matching between the predicted explanation and the ground truth by quantifying the fraction of graph components found in the predicted explanation relative to those in the ground truth explanations.
- *Area Under the ROC Curve*: Formalizing the explanation task as binary classification, the ROC curve is a graphical representation of the true positive rate against the false positive rate at different threshold values. The area under the curve quantifies the overall explainer performance in identifying the relevant graph components.
- *Precision*: Quantifies the proportion of truly relevant graph components identified by the explainer relative to the full retrieved explanation, indicating the explainer's ability to exclude irrelevant components.
- *Recall*: It quantifies the proportion of genuinely relevant graph components the explainer identifies relative to the sum of correctly identified components and those relevant components that the explainer did not retrieve. Intuitively, it measures the ability of the explainer to find all the relevant graph components.

All the preceding metrics require the ground truth explanation for the respective graphs. In the absence of a gold standard, alternative metrics can evaluate the quality of the explainer:

- *Stability*: Stability, or explanation similarity, examines the consistency of explanations across similar instances, assessing both structural and node feature similarity [27, 65, 90]. This involves comparing explanations for two samples either through an intersection over the union method [30], tallying common elements in graphs, or by assessing similarity in vector-based representations. Essentially, their explanations should align when slightly different graphs or nodes yield identical prediction labels. Lower values indicate greater stability, indicating the explanation method's robustness to noise. Given two similar graph samples G^1 and G^2 such that $f(G^1) \sim f(G^2)$, as

[7] https://hotpotqa.github.io/
[8] https://www.tau-nlp.sites.tau.ac.il/commonsenseqa
[9] https://allenai.org/data/wiqa

Table 7. Previously employed graph-based real datasets.

Dataset	Task	Domain	Variants	Publicly Available
MUTAG	GC & Clustering	Chemistry	-	PyTorch Datasets
Proteins	GC	Chemistry	-	PyTorch Datasets
PTC	GC	Chemistry	-	ChemDB
NCI1	GC	Chemistry	-	GitHub
MoleculeNet	GC & Regression	Chemistry	-	MoleculeNet
REDDIT-BINARY	GC	Social	REDDIT-MULTI 5K	PyTorch Datasets
IMDB BINARY	GC& Clustering	Social	-	PyTorch Datasets
Visual Genome	GC	Visual	-	Visualgenome
MNIST SuperPixel-Graph	GC	Visual	-	PyTorch Datasets
SST	GC	Text	Graph SST2	Stanford NLP Group
Graph-Twitter	GC	Text	-	GitHub
PPI	NC	Chemistry	-	PyTorch Datasets
Bitcoin-OTC	NC	Social	-	Stanford NLP Group
Bitcoin-Alpha	NC	Social	-	Stanford NLP Group
Cora	NC	Text	-	PyTorch Datasets
CiteSeer	NC	Text	-	PyTorch Datasets
PubMed	NC	Text	-	PyTorch Datasets
DBLP	LP	Social	-	PyTorch Datasets
Karate Club	LP	Social	-	PyTorch Datasets
Game of Thrones	LP	Entertainment	-	GitHub
FB15k-237	KG Completion	KG	-	PyTorch Datasets
WN18RR	KG Completion	KG	-	GitHub
NELL-995	KG Completion	KG	-	GitHub
Epinions	recommendation	Social	-	DataLab SNU
Amazon	recommendation	Commerce	-	PyTorch Datasets
FilmTrust	recommendation	Entertainment	-	DataLab SNU
Flixster	recommendation	Entertainment	-	PyTorch Datasets
Douban	recommendation	Entertainment	-	PyTorch Datasets
MovieLens	recommendation, GC, LP	Entertainment	MovieLens-1M/20M	PyTorch Datasets
MI	recommendation	Entertainment	-	GitHub
Book-Crossing	recommendation	Entertainment	-	GitHub
Last FM	recommendation	Entertainment	-	PyTorch Datasets
KKBox	recommendation	Entertainment	-	Kaggle
Pinterest	recommendation	Entertainment	-	ICCV Dataset
Yelp	recommendation	Entertainment	-	PyTorch Datasets
WebKB	Clustering	Organization	-	PyTorch Datasets
UW-CSE	Clustering	Organization	-	Alchemy UW
Yago Artwork	Clustering	KG	-	YAGO KG
Breast Cancer Patient	Signal Prediction	Medicine	-	GitLab
English CoNLL-2009	SRL	Text	-	UFAL CU
EXPLAGRAPHS	Stance Detection	Text	-	GitHub

mentioned in Section 3, stability can be computed as:

$$Stability = \frac{|G^1_\Phi \cap G^2_\Phi|}{|G^1_\Phi \cup G^2_\Phi|}$$

where union and intersection methods encompass any set of graph components, whether nodes or edges, depending on the scope of the explanation method applied. $|\cdot|$ denotes the graph size based on its total number of nodes.

- *Fidelity*: Fidelity assesses the reduction in prediction confidence upon the removal of the explanation from the input graph [11, 93, 147, 150]. A higher fidelity score means more pronounced counterfactual traits, albeit sensitivity to

sparsity should be acknowledged. Fidelity essentially measures whether the prediction remains faithfully contingent on the model's decision after eliminating the graph components identified as relevant.

$$Fidelity = f(G) - f(G_\Phi^c)$$

where $G_\Phi^c = G - G_\Phi$ corresponds to the complement of the graph explanation, i.e. those graph components that were not selected as relevant.

- *Inverse Fidelity*: Instead of removing the explanation from the input graph, the inverse fidelity measures whether the prediction is faithfully important to the model prediction by only keeping the selected components [150].

$$Inverse\ Fidelity = f(G) - f(G_\Phi)$$

- *Sparsity*: Explanation sparsity concerning an input graph denotes the proportion of edges retained after the explanation's removal [93, 125, 138, 144, 147]. A higher sparsity score indicates fewer edges identified as explanations. This fosters equitable comparison by standardizing explanation sizes, as explanations of differing sizes are not directly comparable and larger explanations typically enhance *fidelity* and *inverse fidelity*.

$$Sparsity = 1 - \frac{|G_\Phi|}{|G|}$$

- *Contrastivity*: It evaluates the decisiveness of the explanation [63, 93]. Assuming that explanations for separate classes should differ and have no common patterns, a contrastive explainer should delineate a discernible boundary, with explanation elements of markedly contrasting contribution values. Contrastivity is the ratio of the distance between explanations for different classes, normalized by the total identified relevant features.

$$Contrastivity = \frac{\text{distance}(G_\Phi^1, G_\Phi^2)}{|G_\Phi^1| + |G_\Phi^2|}$$

- *Per cluster Coverage (Cov)* These metrics are designed specifically for clustering explainers rooted in logic rules [44]. Coverage of a rule pertains to the entities it encompasses. When comparing two explanation rules within a cluster, preference is given to the one covering more entities. However, since a given explanation rule may be too general, and cover entities from more than a single cluster, coverage exclusiveness is introduced. This concept stipulates that explanation rules for distinct clusters should be (approximately) mutually exclusive, featuring high coverage for the given cluster but low coverage for others.

$$Cov = \frac{|\{e \in C | \text{match}(e, r)\}|}{|C|}$$

where match$(\cdot)$ evaluates if an entity e, representing the cluster C, matches all the predicates contained in the explanation rule r.

- *Faithfulness*: In recommender systems, faithfulness measures path-based explainers [155] by assessing the alignment between generated explainable paths and the user's past behavior. Faithfulness is quantified by evaluating the deviation of collected paths as rule-related distributions across training and test datasets. Thus, smaller divergence values mean stronger alignment between explainable paths and the observed user history during training.

$$Faithfulness = \mathbb{E}_{u \sim U}[D(Q(u)||R(u))]$$

where $D(\cdot)$ is a measure of divergence between two provided distributions, $R(u)$ represents the rule distribution of the existing paths for each user u in the training set, and $Q(u)$ corresponds to the rule distribution of the extracted explanation paths for each user u during testing.

6 APPLICATIONS

Graph-based explainers are versatile and effective across various fields. This section delves into practical applications of these explainers, including adaptations and extensions of previously proposed graph-based explainers (Section 4), innovative application areas, and comparative studies assessing their effectiveness.

In financial transaction data analysis, [72] extended GNNExplainer by considering edge weights and adding entropy regularization to ensure the connectivity of the final explanation graph. [99] introduced xFraud as an explainable fraud transaction prediction framework based on GNNExplainer. xFraud converts transaction logs into a graph structure to generate a transaction risk score for each transaction record. GNNExplainer has also been extended in various technological contexts. XG-Bot [77] detects malicious botnet nodes within large-scale networks by highlighting network flows and related botnet nodes. Lin et al. [76] defines a GNN policy to study manipulation tasks, identifying the most influential spatial relationships and neighbors affecting policy decisions. Traditional approaches and LORE were applied for sanity checks to detect Android malware [38], with criteria to evaluate the stability, robustness, and effectiveness of five explanation methods. In autonomous driving, [87] employs ExCut to elucidate clustering outcomes and understand scene context, integrating commonsense knowledge into driving knowledge graphs.

A recent work exploring counterfactual reasoning is CF-GNNExplainer [79]. Based on GNNExplainer, it generates minimal yet crucial explanations for GNNs. However, their validation is limited to synthetic data. Alternatively, [36] undertakes a comparative study evaluating IG, Grad-CAM, PGMExplainer, and GNNExplainer methods in graph analysis tasks, such as community prediction for nodes and graph dynamics.

Graph explainers are also used in NLP. GNNExplainer was adapted in [48] to identify key subgraph structures for question generation, while [106] extended LRP to elucidate the contribution of graph elements for sentiment analysis.

An experimental evaluation of GNNExplainer, PGExplainer, and GraphMASK for node classification on citation network datasets is presented in [71], including a human-free evaluation metric. Another explainable recommender system is introduced in [127]. Motivated by PGPR, the authors propose a reinforcement learning framework for multi-level recommendation reasoning over KGs to model multi-level user interests. Effectiveness studies for a range of explainable recommendation approaches based on sequential paths were also conducted in [13, 14].

Understanding social phenomena dynamics is also relevant. Inspired by gradient-based techniques, Xie and Lu [137] proposes a node attribution method for explaining node classification in a citation network. Ganesan et al. [46] studies Covid contact tracing as a link prediction task, while WGNNExplainer [83] is introduced for the analysis of a drug abuse social network. Inspired by GNNExplainer, it incorporates edge weight information to elucidate node classification predictions. GNN-SubNet [92] detects disease subnetworks by categorizing patients in a graph and employing a modified GNNExplainer algorithm to reveal classifier decisions. In a separate study, Halliwell et al. [53] compares explanations from ExplaiNE and GNNExplainer for link prediction on KGs containing royalty family members[10].

Graph-based explainers have significantly supported chemistry and biology domains. Motivated by GNNExplainer, [62] presents a cell-graph explainer that only defines a masking function over the graph nodes for disease diagnosis, while [140] proposes a variant of the explainer to identify the proteins associated with polypharmacy side effects.

[10] https://gitlab.com/halliwelln/royalty-datasets

Alternatively, Xiong et al. [139] extended GNN-LRP by proposing a polynomial-time algorithm to handle the exponential complexity of explanation identification in large-scale graphs. In turn, IG was applied to reveal the most relevant components of a molecular compound in retrosynthetic reaction prediction and molecule classification [59, 94].

Graph-based explainers have been utilized to assess the efficacy of novel approaches for training GNNs by examining their influence on model prediction interpretability. Loveland et al. [78] investigated the effects of adversarial training, [56] proposed two regularization techniques, while Fan et al. [39] tested the generalization ability of GNNs in out-of-distribution (OOD) settings. Two other comparative studies are presented in [98, 103].

In the realm of medicine, graph explainers have made significant contributions, supporting the identification of brain regions of interest and related diseases [7, 25], interpreting results in prostate cancer detection [7], and identifying major depressive disorders [64]. Moreover, an extension of integrated gradients, Expected Gradients [35], was designed for regularization as an attribution prior that can be regularized during training rather than explain a model's prediction as a post-hoc method. Additionally, two effectiveness studies have explored breast cancer detection as a graph classification task. In [23] the authors analyzed the explanations obtained by employing LRP and the traditional Shapley values, while [61] contrasted GraphLRP, GNNExplainer, and Grad-CAM, suggesting pathologically quality metrics.

Other studies expanded the comparison across various domains. Agarwal et al. [3] considers scoring and subgraph-based explainers in nine real-world graph datasets encompassing graph and node classification, and link prediction. In [2], a synthetic graph data generator was introduced to create benchmark datasets with ground-truth explanations, focusing on more challenging scenarios than traditional synthetic datasets. In turn, Fang et al. [40] addresses the OOD issue by introducing a novel evaluation metric and assessing explainers across molecular, visual, and synthetic data.

7 COMPARATIVE ANALYSIS

Explanation methods grounded in logic rules and generative approaches are the least utilized and least developed explanation category, featuring a restricted assortment of models (Table 3 and Table 5). Even though logic rules-based explainers have demonstrated effectiveness across various data models, including tabular data, text, and arbitrary graphs, they suffer from significant drawbacks that severely constrain their applicability. They necessitate specialized expertise, with practitioners often required to engage in manual decision-making processes to define model hyperparameters. Additionally, the explanations derived from logic rules-based models tend to be broad and simplistic. Conversely, generation-based explainers are the only category aiming to capture the global model's behavior, a notable challenge when compared to explaining individual instances. Therefore, this category tends to be computationally intensive, facing scalability issues. Furthermore, there are no guarantees that the generated explanation graph will follow the graph structure of previously encountered training samples, difficulting results interpretability. Consequently, both categories have unique characteristics but face challenges for effective and efficient application. An additional constraint hindering the advancement of new explanation models within these two categories is the absence of standardized quality metrics. Both logic rules-based explainers and generation-based explainers, define metrics that depend on the task and application domain, being very specialized. This pushes practitioners to validate their proposals through qualitative case studies.

Scoring and subgraph-based explainers have garnered considerable attention and exhibit significant development across various proposals. However, they are tailored exclusively for explaining GNN models, which are constrained to elucidate nodes and features within a limited neighborhood determined by the number of neural layers utilized during training. Due to the phenomenon of over-smoothing, GNNs can only stack a few layers in their architecture, restricting their ability to unveil dependencies beyond this local computational graph. Consequently, the explainers

may overlook relevant structures beyond the defined neighborhood [43]. Moreover, they work in a post-hoc manner identifying only a subset of relevant graph components for the given prediction. This has been shown to be highly vulnerable to adversarial perturbations [74], where minor alterations of the original graph structure that still preserve the model's predictions may result in significantly different explanations.

More generally, all explanation categories encounter scalability challenges, with some employing approximation strategies to mitigate these issues. Notably, the graph explainers discussed here have predominantly been tested in controlled environments, neglecting large-scale graph scenarios, which could significantly limit their applicability in more complex domains.

Focusing on the specific approaches utilized by each explanation modality, Figure 2 illustrates that while half are exclusive to a single explanation format, the remaining half are versatile approaches applicable across multiple modalities, proving beneficial for generating various explanation formats. For instance, gradient-based and decomposition approaches typically pertain to scoring explainers, aiming to depict the sensitivity between input and output. Similarly, path reasoning corresponds to sequential path-based explainers, while perturbation techniques contribute to subgraph-based explanations. Conversely, other methods span multiple explanation categories and demonstrate applicability across broader domains. These include graph creation, surrogate, and data integration, with the latter offering the most flexibility by facilitating the construction of various extraction explainers (sequential path, logic rules, or subgraph). Hence, a rigorous analysis of available data and comparison of extracted patterns, though seemingly straightforward, exhibit effectiveness within extraction-based explanation contexts.

An often overlooked aspect in discussions surrounding explainability is the assessment of explanation quality. Objective evaluation of explanation correctness necessitates standardized definitions and calculation formulas accepted within the scientific community. This survey highlights the existence of numerous metrics for quantitatively measuring explanation quality. However, disparities persist in metric definitions, particularly influenced by application domains, leading to considerable variability in measurement approaches [30, 43].

Scoring-based and subgraph explainers benefit from well-defined metrics widely accepted within the community, given their focus on input-output sensitivity measurement in GNN models. Conversely, other categories face greater challenges in metric definition due to their broader applicability beyond GNNs, as Table 8 shows. Instead, they can explain textual data, KGs predictions (for new associations, and item recommendations), and natural language-related tasks, employing graphs. This variability in data formats and learning models complicates the establishment of standard quality metrics. Consequently, authors often propose specialized metrics tailored to specific tasks and explanation strategies or resort to qualitative validation through expert analysis within the application domain [43, 44]. Therefore, new metrics are needed to assess the explanation quality in a general manner, irrespective of the application domain or learning model employed. Current approaches are inadequate for tasks where predictions extend beyond individual values. For instance, in clustering, the evaluation extends to the error across cluster members and entities classified differently. Similarly, the direct correspondence between explanation correctness and prediction accuracy is elusive in multi-label environments, where predictions are not singular.

8 UNDERSTANDING THE DYNAMICS OF THE FIELD

To understand the development of the area of explanation in graph structures, Figure 8a shows the annual introduction of new methods and extensions of previously proposed explainers. Figure 8b illustrates the distribution of explainers based on the explanation modality they offer. Notably, the figures are based solely on papers referenced in Section 4, excluding those detailed in Section 6. A grouping of explanation methods by explanation approach is shown in Figure 9.

Table 8. Comparative table for each explainer category. Target column indicates the target structure to be explained.

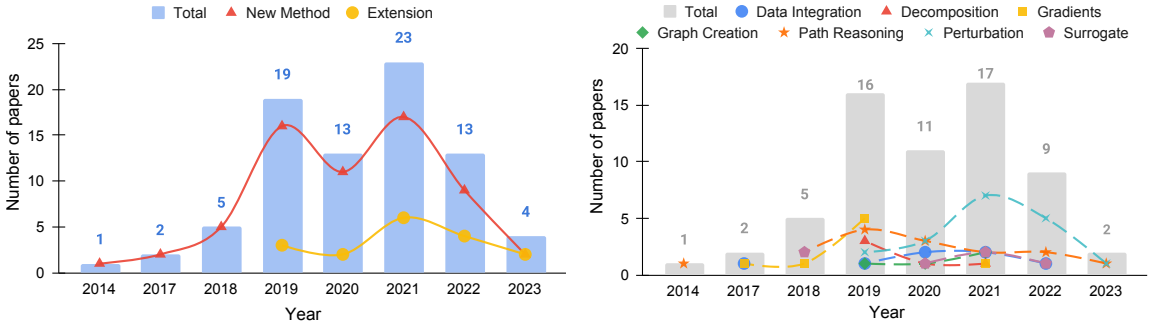
Modality	Approach	Target	Metrics		Domains
			Standard	Special	
Scoring	Decomposition Gradients	GNN	•		Biology & Chemistry NLP Visual
Sequential Path Extraction	Data Integration Path Reasoning	KG		•	Commerce Entertainment NLP Organizations
Logic Rules Extraction	Data Integration Surrogate	KG Tabular data Text		•	Biology & Chemistry Entertainment NLP Social
Subgraph Extraction	Data Integration Graph Creation Perturbation Surrogate	GNN	•		Biology & Chemistry Commerce Entertainment Visual
Generation	Graph Creation	GNN Text		•	Biology & Chemistry NLP

Despite substantial growth in the field, a significant surge in new model proposals is observed primarily between 2019 and 2022, surpassing other years notably. This trend extends to extension methods and adaptations of existing models, which began in 2019 and peaked in 2021, aligning with the distribution of new methods (Figure 8a).

The notable increase in the number of new methods underscores the growing interest in the graph-based explanation area. This trend is paralleled by the proliferation of scoring and subgraph strategies (Figure 8b), each reaching a distinct peak. Scoring explainers peaked in 2019, as they were the initial strategies employed for graph explanation, influenced by traditional explainer models in Euclidean space. Subsequently, in 2021, subgraph-based explainers experienced another distinct peak, gradually diminishing after that. Intriguingly, the number of new models and the number of adaptations were comparable in the last year, indicating a decline in the exploration of novel approaches for graph explanation across various applications and tasks. Notably, the explanation methods based on subgraphs and scoring were extensively studied during their peak. This surge correlates with the rise of GNN models during the same period, as evidenced by Table 1 and Table 4, wherein all scoring-based explainers and the majority of subgraph-based explainers were tailored for GNNs. Consequently, these explainers are limited in their ability to address a broad range of tasks, posing challenges for the proposal of new models within the same modality.

Figure 9 shows that the predominant approach among explainers involves perturbing the input space to observe their impact on the final prediction of the underlying learning model. It is consistent with the trends observed in Figure 8b, where subgraph-based explainers dominate (refer to Figure 2 for categorization). However, explainers of alternative modalities, not specifically tailored for GNN models, exhibit ongoing development over the years. Some have maintained a presence over the past decade, while others are limited to specific periods.

In particular, explainers based on logic rules demonstrate minimal presence despite their ease of application and comprehensibility. This is attributed to the broad and simplistic nature of the provided explanations compared to other strategies. Similarly, generation-based explainers show limited development and the least presence, as shown in Table 5 and discussed in Section 7. Conversely, explanation models based on sequential paths represent the only category of explainers that exhibit effective and sustained development over time. Despite lacking a distinct development peak in



(a) Explainer methods and extensions of proposed models.

(b) Explainer methods according to the explanation modality.

Fig. 8. Number of papers across years by different categories.

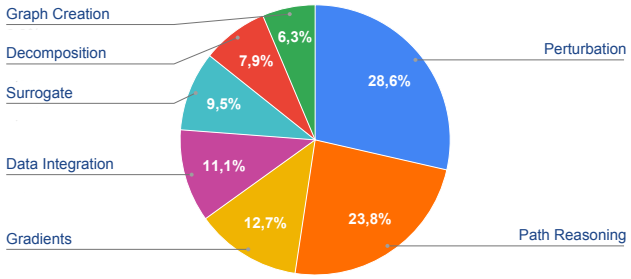


Fig. 9. Portion of explainers proposed, categorized by the approach used to generate the final explanation.

the timeline of Figure 8b, they have remained relevant for over a decade. Thus, path reasoning approaches on a given graph, along with data integration and pattern analysis, serve as effective methods for elucidating predictions obtained by learning models across various tasks such as recommender systems, KG completion, link prediction, and multiple NLP tasks, as validated by previous research discussed in Section 4.2.1.

8.1 Open Challenges

In addition to the necessity for innovative strategies addressing the limitations of previous models, several significant challenges could greatly impact the development of the field.

Ground Truth Explanations - The need for new real datasets tailored for explanation in graph data, including ground truth explanations, is a concern. While there are plenty of real datasets, they lack actual explanations. Only a very limited fraction, such as MUTAG, has been previously treated as a real dataset with verified ground truths, albeit through expert validation reliant on domain-specific knowledge.

Standardized Metrics - While various metrics have been proposed and widely accepted by the research community, many are limited to assessing the explainer's sensitivity, while others are specialized for specific settings [43, 44]. Moreover, metrics reported across multiple papers often possess differing definitions and measurement methodologies [30, 43], contributing to inconsistency. A precise and standardized definition of metrics is imperative to enhance the applicability of explanation methods, facilitate results reproducibility, and allow direct comparison with other methods.

Efficiency and Scalability - Resource utilization and efficiency pose a persistent challenge for existing graph-based explainers. New explanation models designed for graph data must exhibit both efficiency and scalability, integrating effective resource management to ensure their suitability for real-world, large-scale scenarios.
Unexplored Areas - While graphs can represent diverse real-world data, certain applications remain underexplored. Tasks within political-social domains, crime analysis, and economic and financial services could benefit significantly from incorporating explanation methods, enhancing predictions in sensitive domains.
Multi-label Settings - Despite recent efforts to develop task-agnostic explainers and explainers for handling multi-task frameworks [125, 138], there remains a gap in addressing multi-label environments. In broad terms, the primary aim of current explanation models is to examine the relationship between the generated explanation and the label predicted by the underlying learning model. However, effectively explaining cases where this label corresponds to a range of values remains an unexplored area worthy of investigation.

9 CONCLUSIONS

Previous surveys on explainability for graph-based learning models have focused solely on GNN models, neglecting significant methodologies not rooted in GNNs. This paper introduces a novel hierarchical categorization of graph explanation models based on their explanation modality and the associated explainability approaches. This encompasses deep learning models grounded in Reinforcement Learning, Multi-Hop Reasoning, Knowledge Graphs, GNNs, and traditional machine learning models. We have examined the strengths and limitations of each approach, considering their suitability across diverse domains and downstream tasks, as well as methods for evaluating explanation quality across different explainer categories. Throughout this survey, we emphasize that there are no inherently superior or inferior explainers. Rather, each serves a distinct purpose with varying operational modes. Our comprehensive analysis aims to provide readers with an understanding of the contexts and advantages of each explainer, allowing informed decisions in various graph-related scenarios. Furthermore, we highlight ongoing challenges and areas for improvement in graph-based explainability to enhance its development and applicability in real-world settings.

ACKNOWLEDGMENTS

This study was possible due to the funding of the Data Science and Engineering (DSE) Research School program at Hasso Plattner Institute.

REFERENCES

[1] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. 2018. Sanity checks for saliency maps. In *Advances in neural information processing systems*, Vol. 31. Curran Associates Inc., ., 9505–9515.
[2] Chirag Agarwal, Owen Queen, Himabindu Lakkaraju, and Marinka Zitnik. 2023. Evaluating explainability for graph neural networks. *Scientific data* 10, 1 (2023), 144. https://doi.org/10.1038/s41597-023-01974-x
[3] Chirag Agarwal, Marinka Zitnik, and Himabindu Lakkaraju. 2022. Probing GNN explainers: A rigorous theoretical and empirical analysis of GNN explanation methods. In *International conference on artificial intelligence and statistics*. PMLR, ., 8969–8996.
[4] Qingyao Ai, Vahid Azizi, Xu Chen, and Yongfeng Zhang. 2018. Learning heterogeneous knowledge base embeddings for explainable recommendation. *Algorithms* 11, 9 (2018), 137. https://doi.org/10.3390/a11090137
[5] Hirotaka Akita, Kosuke Nakago, Tomoki Komatsu, Yohei Sugawara, Shin-ichi Maeda, Yukino Baba, and Hisashi Kashima. 2018. BayesGrad: Explaining predictions of graph convolutional networks. In *Neural information processing*, Long Cheng, Andrew Chi Sing Leung, and Seiichi Ozawa (Eds.). Springer, Cham, ., 81–92. https://doi.org/10.1007/978-3-030-04221-9_8
[6] Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. 2016. Deep variational information bottleneck. In *International conference on learning representations*. ICLR, ., 1–19.
[7] Valentin Anklin, Pushpak Pati, Guillaume Jaume, Behzad Bozorgtabar, Antonio Foncubierta-Rodriguez, Jean-Philippe Thiran, Mathilde Sibony, Maria Gabrani, and Orcun Goksel. 2021. Learning whole-slide segmentation from inexact and incomplete labels using tissue graphs. In *Medical*

image computing and computer assisted intervention. Springer, Cham, ., 636–646. https://doi.org/10.1007/978-3-030-87196-3_59

[8] Nihat Ay and Daniel Polani. 2008. Information flows in causal networks. *Advances in complex systems* 11 (2008), 17–41. https://doi.org/10.1142/S0219525908001465

[9] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. 2015. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one* 10, 7 (2015), e0130140. https://doi.org/10.1371/journal.pone.0130140

[10] David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert Müller. 2010. How to explain individual classification decisions. *Journal of machine learning research* 11 (2010), 1803–1831.

[11] Mohit Bajaj, Lingyang Chu, Zi Yu Xue, Jian Pei, Lanjun Wang, Peter Cho-Ho Lam, and Yong Zhang. 2021. Robust counterfactual explanations on graph neural networks. In *Advances in neural information processing systems*, Vol. 34. Curran Associates Inc., ., 5644–5655.

[12] Federico Baldassarre and Hossein Azizpour. 2019. Explainability techniques for graph convolutional networks.

[13] Giacomo Balloccu, Ludovico Boratto, Christian Cancedda, Gianni Fenu, and Mirko Marras. 2023. Knowledge is power, understanding is impact: Utility and beyond goals, explanation quality, and fairness in path reasoning recommendation. In *Advances in information retrieval*, Jaap Kamps, Lorraine Goeuriot, Fabio Crestani, Maria Maistro, Hideo Joho, Brian Davis, Cathal Gurrin, Udo Kruschwitz, and Annalina Caputo (Eds.). Springer, Cham, ., 3–19. https://doi.org/10.1007/978-3-031-28241-6_1

[14] Giacomo Balloccu, Ludovico Boratto, Gianni Fenu, and Mirko Marras. 2023. Reinforcement recommendation reasoning through knowledge graphs for explanation path quality. *Knowledge-based systems* 260 (2023), 110098. https://doi.org/10.1016/j.knosys.2022.110098

[15] Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big?. In *ACM conference on fairness, accountability, and transparency*. ACM, ., 610–623. https://doi.org/10.1145/3442188.3445922

[16] Rajarshi Bhowmik and Gerard de Melo. 2020. Explainable link prediction for emerging ntities in knowledge graphs. In *Semantic web*, Jeff Z. Pan, Valentina Tamma, Claudia d'Amato, Krzysztof Janowicz, Bo Fu, Axel Polleres, Oshani Seneviratne, and Lalana Kagal (Eds.). Springer, Cham, ., 39–55. https://doi.org/10.1007/978-3-030-62419-4_3

[17] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, Vol. 26. Curran Associates Inc., ., 2787–2795.

[18] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. In *Advances in neural information processing systems*, Vol. 33. Curran Associates Inc., ., 1877–1901.

[19] Margarita Bugueño and Gerard de Melo. 2023. Connecting the dots: What graph-based text representations work best for text classification using graph neural networks?. In *Findings of the association for computational linguistics*. ACL, ., 8943–8960. https://doi.org/10.18653/v1/2023.findings-emnlp.600

[20] Margarita Bugueño and Marcelo Mendoza. 2020. Learning to Detect Online Harassment on Twitter with the Transformer. In *Machine learning and knowledge discovery in databases Workshop at ECML-PKDD*. Springer, Cham, ., 298–306. https://doi.org/10.1007/978-3-030-43887-6_23

[21] Diogo V Carvalho, Eduardo M Pereira, and Jaime S Cardoso. 2019. Machine learning interpretability: A survey on methods and metrics. *Electronics* 8, 8 (2019), 832. https://doi.org/10.3390/electronics8080832

[22] Hryhorii Chereda, Annalen Bleckmann, Kerstin Menck, Júlia Perera-Bel, Philip Stegmaier, Florian Auer, Frank Kramer, Andreas Leha, and Tim Beißbarth. 2021. Explaining decisions of graph convolutional neural networks: Patient-specific molecular subnetworks responsible for metastasis prediction in breast cancer. *Genome medicine* 13, 1 (2021), 1–16. https://doi.org/10.1186/s13073-021-00845-7

[23] Hryhorii Chereda, Andreas Leha, and Tim Beißbarth. 2024. Stable feature selection utilizing Graph Convolutional Neural Network and Layer-wise Relevance Propagation for biomarker discovery in breast cancer. *Artificial intelligence in medicine* 151 (2024), 102840. https://doi.org/10.1016/j.artmed.2024.102840

[24] ACM US Public Policy Council. 2017. Statement on algorithmic transparency and accountability.

[25] Hejie Cui, Wei Dai, Yanqiao Zhu, Xiaoxiao Li, Lifang He, and Carl Yang. 2021. BrainNNExplainer: An interpretable graph neural network framework for brain network based disease analysis.

[26] Piotr Dabkowski and Yarin Gal. 2017. Real time image saliency for black box classifiers. In *Advances in neural information processing systems*, Vol. 30. Curran Associates Inc., ., 6967–6976.

[27] Enyan Dai and Suhang Wang. 2021. Towards self-explainable graph neural network. In *ACM international conference on information & knowledge management*. ACM, ., 302–311. https://doi.org/10.1145/3459637.3482306

[28] Enyan Dai, Tianxiang Zhao, Huaisheng Zhu, Junjie Xu, Zhimeng Guo, Hui Liu, Jiliang Tang, and Suhang Wang. 2022. A comprehensive survey on trustworthy graph neural networks: Privacy, robustness, fairness, and explainability.

[29] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive Language Models beyond a Fixed-Length Context. In *Annual meeting of the association for computational linguistics*. ACL, ., 2978–2988. https://doi.org/10.18653/v1/P19-1285

[30] Songgaojun Deng, Huzefa Rangwala, and Yue Ning. 2021. Understanding event predictions via contextualized multilevel feature learning. In *ACM international conference on information & knowledge management*. ACM, ., 342–351. https://doi.org/10.1145/3459637.3482309

[31] Ming Ding, Chang Zhou, Qibin Chen, Hongxia Yang, and Jie Tang. 2019. Cognitive graph for multi-hop reading comprehension at scale. In *Annual meeting of the association for computational linguistics*. ACL, ., 2694–2703. https://doi.org/10.18653/v1/P19-1259

[32] Finale Doshi-Velez and Been Kim. 2017. Towards a rigorous science of interpretable machine learning.

[33] Li Du, Xiao Ding, Kai Xiong, Ting Liu, and Bing Qin. 2021. ExCAR: Event graph knowledge enhanced explainable causal reasoning. In *Annual meeting of the Association for computational linguistics and the international joint conference on natural language processing*. ACL, ., 2354–2363. https://doi.org/10.18653/v1/2021.acl-long.183

[34] Alexandre Duval and Fragkiskos D Malliaros. 2021. GraphSVX: Shapley value explanations for graph neural networks. In *Joint European conference on machine learning and knowledge discovery in databases*. Springer, Cham, ., 302–318. https://doi.org/10.1007/978-3-030-86520-7_19

[35] Gabriel Erion, Joseph D Janizek, Pascal Sturmfels, Scott M Lundberg, and Su-In Lee. 2021. Improving performance of deep learning models with axiomatic attribution priors and expected gradients. *Nature machine intelligence* 3, 7 (2021), 620–631. https://doi.org/10.1038/s42256-021-00343-w

[36] Lukas Faber, Amin K. Moghaddam, and Roger Wattenhofer. 2021. When comparing to ground truth is wrong: On evaluating GNN explanation methods. In *ACM SIGKDD conference on knowledge discovery & data mining*. ACM, ., 332–341. https://doi.org/10.1145/3447548.3467283

[37] Lukas Faber, Amin K Moghaddam, and Roger Wattenhofer. 2020. Contrastive graph neural network explanation.

[38] Ming Fan, Wenying Wei, Xiaofei Xie, Yang Liu, Xiaohong Guan, and Ting Liu. 2020. Can we trust your explanations? Sanity checks for interpreters in Android malware analysis. *IEEE transactions on information forensics and security* 16 (2020), 838–853. https://doi.org/10.1109/TIFS.2020.3021924

[39] Shaohua Fan, Xiao Wang, Chuan Shi, Peng Cui, and Bai Wang. 2023. Generalizing graph neural networks on out-of-distribution graphs. *IEEE transactions on pattern analysis and machine intelligence* 46 (2023), 322–337. https://doi.org/10.1109/TPAMI.2023.3321097

[40] Junfeng Fang, Wei Liu, Yuan Gao, Zemin Liu, An Zhang, Xiang Wang, and Xiangnan He. 2024. Evaluating post-hoc explanations for graph neural networks via robustness analysis. In *Advances in neural information processing systems*, Vol. 36. Curran Associates Inc., ., 0–10.

[41] Nir Friedman, Dan Geiger, and Moises Goldszmidt. 1997. Bayesian network classifiers. *Machine learning* 29, 2 (1997), 131–163. https://doi.org/10.1023/A:1007465528199

[42] Wenjing Fu, Zhaohui Peng, Senzhang Wang, Yang Xu, and Jin Li. 2019. Deeply fusing reviews and contents for cold start users in cross-domain recommendation systems. In *AAAI conference on artificial intelligence*, Vol. 33. AAAI Press, ., 94–101. https://doi.org/10.1609/aaai.v33i01.330194

[43] Thorben Funke, Megha Khosla, Mandeep Rathee, and Avishek Anand. 2022. Zorro: Valid, sparse, and stable explanations in graph neural networks. *IEEE transactions on knowledge and data engineering* 35 (2022), 8687–8698. https://doi.org/10.1109/TKDE.2022.3201170

[44] Mohamed H Gad-Elrab, Daria Stepanova, Trung-Kien Tran, Heike Adel, and Gerhard Weikum. 2020. ExCut: Explainable embedding-based clustering over knowledge graphs. In *International semantic web conference*. Springer, Cham, ., 218–237. https://doi.org/10.1007/978-3-030-62419-4_13

[45] Andrea Galassi, Marco Lippi, and Paolo Torroni. 2020. Attention in natural language processing. *IEEE transactions on neural networks and learning systems* 32, 10 (2020), 4291–4308. https://doi.org/10.1109/TNNLS.2020.3019893

[46] Balaji Ganesan, Hima Patel, and Sameep Mehta. 2020. Explainable link prediction for privacy-preserving contact tracing.

[47] Yuyang Gao, Tong Sun, Rishab Bhatt, Dazhou Yu, Sungsoo Hong, and Liang Zhao. 2021. GNES: Learning to explain graph neural networks. In *IEEE international conference on data mining*. IEEE, ., 131–140. https://doi.org/10.1109/ICDM51629.2021.00023

[48] Yong Liang Goh, Yongbin Li, and Yisong Miao. 2021. Connecting the dots: Explaining human reasoning on the graph A case study on deep question generation.

[49] Clive WJ Granger. 1969. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: Journal of the econometric society* 37, 3 (1969), 424–438. https://doi.org/10.2307/1912791

[50] Arthur Gretton, Olivier Bousquet, Alex Smola, and Bernhard Schölkopf. 2005. Measuring statistical dependence with Hilbert-Schmidt norms. In *International conference on algorithmic learning theory*. Springer, Cham, ., 63–77. https://doi.org/10.1007/11564089_7

[51] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Dino Pedreschi, Franco Turini, and Fosca Giannotti. 2018. Local rule-based explanations of black box decision systems.

[52] Michael Gutmann and Aapo Hyvärinen. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *International conference on artificial intelligence and statistics*. PMLR, ., 297–304.

[53] Nicholas Halliwell, Fabien Gandon, and Freddy Lecue. 2022. A simplified benchmark for ambiguous explanations of knowledge graph link prediction using relational graph convolutional networks. In *AAAI conference on artificial intelligence*, Vol. 36. AAAI Press, ., 12963–12964. https://doi.org/10.1609/aaai.v36i11.21618

[54] Gérard Hamiache and Florian Navarro. 2020. Associated consistency, value and graphs. *International journal of game theory* 49, 1 (2020), 227–249. https://doi.org/10.1007/s00182-019-00688-y

[55] Xu Han, Shulin Cao, Xin Lv, Yankai Lin, Zhiyuan Liu, Maosong Sun, and Juanzi Li. 2018. OpenKE: An open toolkit for knowledge embedding. In *Conference on empirical methods in natural language processing: System demonstrations*. ACL, ., 139–144. https://doi.org/10.18653/v1/D18-2024

[56] Ryan Henderson, Djork-Arné Clevert, and Floriane Montanari. 2021. Improving molecular graph neural network explainability with orthonormalization and induced sparsity. In *International conference on machine learning*. PMLR, ., 4203–4213.

[57] Qiang Huang, Makoto Yamada, Yuan Tian, Dinesh Singh, and Yi Chang. 2022. GraphLime: Local interpretable model explanations for graph neural networks. *IEEE transactions on knowledge and data engineering* 35 (2022), 6968–6972. https://doi.org/10.1109/TKDE.2022.3187455

[58] Xiaowen Huang, Quan Fang, Shengsheng Qian, Jitao Sang, Yan Li, and Changsheng Xu. 2019. Explainable interaction-driven user modeling over knowledge graph for sequential recommendation. In *ACM international conference on multimedia*. ACM, ., 548–556. https://doi.org/10.1145/3343031.3350893

[59] Shoichi Ishida, Kei Terayama, Ryosuke Kojima, Kiyosei Takasu, and Yasushi Okuno. 2019. Prediction and interpretable visualization of retrosynthetic reactions using graph convolutional networks. *Journal of chemical information and modeling* 59, 12 (2019), 5026–5033. https://doi.org/10.1021/acs.

jcim.9b00538

[60] Sarthak Jain and Byron C Wallace. 2019. Attention is not explanation. In *Conference of the North American chapter of the association for computational linguistics: Human language technologies*. ACL, ., 3543–3556. https://doi.org/10.18653/v1/N19-1357

[61] Guillaume Jaume, Pushpak Pati, Behzad Bozorgtabar, Antonio Foncubierta, Anna Maria Anniciello, Florinda Feroce, Tilman Rau, Jean-Philippe Thiran, Maria Gabrani, and Orcun Goksel. 2021. Quantifying explainers of graph neural networks in computational pathology. In *IEEE/CVF conference on computer vision and pattern recognition*. IEEE, ., 8102–8112. https://doi.org/10.1109/CVPR46437.2021.00801

[62] Guillaume Jaume, Pushpak Pati, Antonio Foncubierta-Rodriguez, Florinda Feroce, Giosue Scognamiglio, Anna Maria Anniciello, Jean-Philippe Thiran, Orcun Goksel, and Maria Gabrani. 2020. Towards explainable graph representations in digital pathology.

[63] Chaojie Ji, Ruxin Wang, and Hongyan Wu. 2022. Perturb more, trap more: Understanding behaviors of graph neural networks. *Neurocomputing* 493 (2022), 59–75. https://doi.org/10.1016/j.neucom.2022.04.070

[64] Eunji Jun, Kyoung-Sae Na, Wooyoung Kang, Jiyeon Lee, Heung-Il Suk, and Byung-Joo Ham. 2020. Identifying resting-state effective connectivity abnormalities in drug-naïve major depressive disorder diagnosis via graph convolutional networks. *Human brain mapping* 41, 17 (2020), 4997–5014. https://doi.org/10.1002/hbm.25175

[65] Bo Kang, Jefrey Lijffijt, and Tijl De Bie. 2019. ExplaiNE: An approach for explaining network embedding-based link predictions.

[66] Ehud D Karnin. 1990. A simple procedure for pruning back-propagation trained neural networks. *IEEE transactions on neural networks* 1, 2 (1990), 239–242. https://doi.org/10.1109/72.80236

[67] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. In *International conference on learning representations*. ICLR, ., 1–14.

[68] Harold William Kuhn and Albert William Tucker. 1953. *Contributions to the theory of games*. Vol. 1. JSTOR, .

[69] Hoyeop Lee, Jinbae Im, Seongwon Jang, Hyunsouk Cho, and Sehee Chung. 2019. MeLU: Meta-learned user preference estimator for cold-start recommendation. In *ACM SIGKDD international conference on knowledge discovery & data mining*. ACM, ., 1073–1082. https://doi.org/10.1145/3292500.3330859

[70] John Boaz Lee, Ryan A Rossi, Sungchul Kim, Nesreen K Ahmed, and Eunyee Koh. 2019. Attention models in graphs: A survey. *ACM transactions on knowledge discovery from data* 13, 6 (2019), 1–25. https://doi.org/10.1145/3363574

[71] Peibo Li, Yixing Yang, Maurice Pagnucco, and Yang Song. 2022. Explainability in graph neural networks: An experimental survey.

[72] Xiaoxiao Li, João Saúde, Prashant P. Reddy, and Manuela M. Veloso. 2019. Classifying and understanding financial data using graph neural network.

[73] Yiqiao Li, Jianlong Zhou, Sunny Verma, and Fang Chen. 2022. A survey of explainable graph neural networks: Taxonomy and evaluation metrics.

[74] Zhong Li, Simon Geisler, Yuhang Wang, Stephan Günnemann, and Matthijs van Leeuwen. 2024. Explainable Graph Neural Networks Under Fire. arXiv:2406.06417 [cs.LG] https://arxiv.org/abs/2406.06417

[75] Wanyu Lin, Hao Lan, and Baochun Li. 2021. Generative causal explanations for graph neural networks. In *International conference on machine learning*. PMLR, ., 6666–6679.

[76] Yixin Lin, Austin S Wang, Eric Undersander, and Akshara Rai. 2022. Efficient and interpretable robot manipulation with graph neural networks. *IEEE robotics and automation letters* 7, 2 (2022), 2740–2747. https://doi.org/10.1109/LRA.2022.3143518

[77] Wai Weng Lo, Gayan Kulatilleke, Mohanad Sarhan, Siamak Layeghy, and Marius Portmann. 2023. XG-BoT: An explainable deep graph neural network for botnet detection and forensics. *Internet of things* 22 (2023), 100747. https://doi.org/10.1016/j.iot.2023.100747

[78] Donald Loveland, Shusen Liu, Bhavya Kailkhura, Anna Hiszpanski, and Yong Han. 2021. Reliable graph neural network explanations through adversarial training.

[79] Ana Lucic, Maartje A Ter Hoeve, Gabriele Tolomei, Maarten De Rijke, and Fabrizio Silvestri. 2022. CF-GNNExplainer: Counterfactual explanations for graph neural networks. In *International conference on artificial intelligence and statistics*. PMLR, ., 4499–4511.

[80] Dongsheng Luo, Wei Cheng, Dongkuan Xu, Wenchao Yu, Bo Zong, Haifeng Chen, and Xiang Zhang. 2020. Parameterized explainer for graph neural network. In *Advances in neural information processing systems*, Vol. 33. Curran Associates Inc., ., 19620–19631.

[81] Ting Ma, Longtao Huang, Qianqian Lu, and Songlin Hu. 2023. KR-GCN: Knowledge-aware reasoning with graph convolution network for explainable recommendation. *ACM transactions on Information Systems* 41, 1 (2023), 1–27. https://doi.org/10.1145/3511019

[82] Weizhi Ma, Min Zhang, Yue Cao, Woojeong Jin, Chenyang Wang, Yiqun Liu, Shaoping Ma, and Xiang Ren. 2019. Jointly learning explainable rules for recommendation with knowledge graph. In *World wide web conference*. ACM, ., 1210–1221. https://doi.org/10.1145/3308558.3313607

[83] Zuanjie Ma, Hongming Gu, and Zhenhua Liu. 2021. Understanding drug abuse social network using weighted graph neural networks explainer. In *International conference on computational science and its applications*. Springer, Cham, ., 52–61. https://doi.org/10.1007/978-3-030-86970-0_5

[84] Aravindh Mahendran and Andrea Vedaldi. 2016. Visualizing deep convolutional neural networks using natural pre-images. *International journal of computer vision* 120, 3 (2016), 233–255. https://doi.org/10.1007/s11263-016-0911-8

[85] Akash Kumar Mohankumar, Preksha Nema, Sharan Narasimhan, Mitesh M Khapra, Balaji Vasan Srinivasan, and Balaraman Ravindran. 2020. Towards transparent and explainable attention models. In *Annual meeting of the association for computational linguistics*. ACL, ., 4206–4216. https://doi.org/10.18653/v1/2020.acl-main.387

[86] Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. 2018. Methods for interpreting and understanding deep neural networks. *Digital signal processing* 73 (2018), 1–15. https://doi.org/10.1016/j.dsp.2017.10.011

[87] Sreyasi Nag Chowdhury, Ruwan Wickramarachchi, Mohamed Hassan Gad-Elrab, Daria Stepanova, and Cory Henson. 2021. Towards leveraging commonsense knowledge for autonomous driving. In *International semantic web conference*. CEUR-ws, ., 1–5.

[88] Senthilselvan Natarajan, Subramaniyaswamy Vairavasundaram, Sivaramakrishnan Natarajan, and Amir H Gandomi. 2020. Resolving data sparsity and cold start problem in collaborative filtering recommender system using Linked Open Data. *Expert systems with applications* 149 (2020), 113248. https://doi.org/10.1016/j.eswa.2020.113248

[89] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. 2016. f-GAN: Training generative neural samplers using variational divergence minimization. In *Advances in neural information processing systems*, Vol. 29. Curran Associates Inc., ., 271–279.

[90] Danilo Numeroso and Davide Bacciu. 2021. MEG: Generating molecular counterfactual explanations for deep graph networks. In *International joint conference on neural networks*. IEEE, IEEE, ., 1–8. https://doi.org/10.1109/IJCNN52387.2021.9534266

[91] Haekyu Park, Hyunsik Jeon, Junghwan Kim, Beunguk Ahn, and U Kang. 2017. UniWalk: Explainable and accurate recommendation for rating and network data.

[92] Bastian Pfeifer, Anna Saranti, and Andreas Holzinger. 2022. GNN-SubNet: Disease subnetwork detection with explainable graph neural networks. *Bioinformatics* 38 (2022), ii120–ii126. https://doi.org/10.1093/bioinformatics/btac478

[93] Phillip E Pope, Soheil Kolouri, Mohammad Rostami, Charles E Martin, and Heiko Hoffmann. 2019. Explainability methods for graph convolutional neural networks. In *IEEE/CVF conference on computer vision and pattern recognition*. IEEE, ., 10772–10781. https://doi.org/10.1109/CVPR.2019.01103

[94] Kristina Preuer, Günter Klambauer, Friedrich Rippmann, Sepp Hochreiter, and Thomas Unterthiner. 2019. Interpretable deep learning in drug discovery. *Explainable AI: Interpreting, explaining and visualizing deep learning* 11700 (2019), 331–345. https://doi.org/10.1007/978-3-030-28954-6_18

[95] Lin Qiu, Yunxuan Xiao, Yanru Qu, Hao Zhou, Lei Li, Weinan Zhang, and Yong Yu. 2019. Dynamically fused graph network for multi-hop reasoning. In *Conference of the association for computational linguistics*. ACL, ., 6140–6150. https://doi.org/10.18653/V1/P19-1617

[96] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

[97] Arun Rai. 2020. Explainable AI: From black box to glass box. *Journal of the academy of marketing science* 48, 1 (2020), 137–141. https://doi.org/10.1007/s11747-019-00710-5

[98] Jiahua Rao, Shuangjia Zheng, Yutong Lu, and Yuedong Yang. 2022. Quantitative evaluation of explainable graph neural networks for molecular property prediction. *Patterns* 3, 12 (2022), 100628. https://doi.org/10.1016/j.patter.2022.100628

[99] Susie Xi Rao, Shuai Zhang, Zhichao Han, Zitao Zhang, Wei Min, Zhiyao Chen, Yinan Shan, Yang Zhao, and Ce Zhang. 2021. xFraud: Explainable fraud transaction detection. *VLDB Endowment* 15, 3 (2021), 427–436. https://doi.org/10.14778/3494124.3494128

[100] Gabrielle Ras, Ning Xie, Marcel van Gerven, and Derek Doran. 2022. Explainable deep learning: A field guide for the uninitiated. *Journal of artificial intelligence research* 73 (2022), 329–397. https://doi.org/10.1613/jair.1.13200

[101] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why should I trust you?" Explaining the predictions of any classifier. In *ACM SIGKDD international conference on knowledge discovery and data mining*. ACM, ., 1135–1144. https://doi.org/10.1145/2939672.2939778

[102] Swarnadeep Saha, Prateek Yadav, Lisa Bauer, and Mohit Bansal. 2021. ExplaGraphs: An explanation graph generation task for structured commonsense reasoning. In *Conference on empirical methods in natural language processing*. ACL, ., 7716–7740. https://doi.org/10.18653/v1/2021.emnlp-main.609

[103] Benjamin Sanchez-Lengeling, Jennifer Wei, Brian Lee, Emily Reif, Peter Wang, Wesley Qian, Kevin McCloskey, Lucy Colwell, and Alexander Wiltschko. 2020. Evaluating attribution for graph neural networks. In *Advances in neural information processing systems*, Vol. 33. Curran Associates Inc., ., 5898–5910.

[104] Michael Sejr Schlichtkrull, Nicola De Cao, and Ivan Titov. 2021. Interpreting graph neural networks for NLP with differentiable edge masking. In *International conference on learning representations*. ICLR, ., 1–21.

[105] Thomas Schnake, Oliver Eberle, Jonas Lederer, Shinichi Nakajima, Kristof T Schütt, Klaus-Robert Müller, and Grégoire Montavon. 2021. Higher-order explanations of graph neural networks via relevant walks. *IEEE transactions on pattern analysis and machine intelligence* 44, 11 (2021), 7581–7596. https://doi.org/10.1109/TPAMI.2021.3115452

[106] Robert Schwarzenberg, Marc Hübner, David Harbecke, Christoph Alt, and Leonhard Hennig. 2019. Layerwise relevance visualization in convolutional text graph classifiers. In *Graph-based methods for natural language processing Workshop*. ACL, ., 58–62. https://doi.org/10.18653/v1/D19-5308

[107] Jonas Herskind Sejr, Peter Schneider-Kamp, and Naeem Ayoub. 2021. Surrogate Object Detection Explainer (SODEx) with YOLOv4 and LIME. *Machine learning and knowledge extraction* 3, 3 (2021), 662–671. https://doi.org/10.3390/make3030033

[108] 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 *IEEE international conference on computer vision*. IEEE, ., 618–626. https://doi.org/10.1109/ICCV.2017.74

[109] Sofia Serrano and Noah A Smith. 2019. Is attention interpretable?. In *Annual meeting of the association for computational linguistics*. ACL, ., 2931–2951. https://doi.org/10.18653/v1/P19-1282

[110] Caihua Shan, Yifei Shen, Yao Zhang, Xiang Li, and Dongsheng Li. 2021. Reinforcement learning enhanced explainer for graph neural networks. In *Advances in neural information processing systems*, Vol. 34. Curran Associates Inc., ., 22523–22533.

[111] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. 2017. Learning important features through propagating activation differences. In *International conference on machine learning*. PMLR, ., 3145–3153.

[112] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. 2017. Mastering the game of Go without human knowledge. *Nature* 550, 7676 (2017), 354–359. https://doi.org/10.1038/nature24270

[113] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *International conference on learning representations*. ICLR, ., 1–8.

[114] Chris R Sims. 2016. Rate–distortion theory and human perception. *Cognition* 152 (2016), 181–198. https://doi.org/10.1016/j.cognition.2016.03.020

[115] Weiping Song, Zhijian Duan, Ziqing Yang, Hao Zhu, Ming Zhang, and Jian Tang. 2019. Ekar: An explainable method for knowledge aware recommendation.

[116] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. 2015. Striving for simplicity: The all convolutional net.

[117] Chengcheng Sun, Chenhao Li, Xiang Lin, Tianji Zheng, Fanrong Meng, Xiaobin Rui, and Zhixiao Wang. 2023. Attention-based graph neural networks: A survey. *Artificial intelligence review* 56 (2023), 2263–2310. https://doi.org/10.1007/s10462-023-10577-2

[118] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *International conference on machine learning*. PMLR, ., 3319–3328.

[119] Juntao Tan, Shijie Geng, Zuohui Fu, Yingqiang Ge, Shuyuan Xu, Yunqi Li, and Yongfeng Zhang. 2022. Learning and evaluating graph neural network explanations based on counterfactual and factual reasoning. In *ACM web conference*. ACM, ., 1018–1027. https://doi.org/10.1145/3485447.3511948

[120] Ilaria Tiddi, Mathieu d'Aquin, and Enrico Motta. 2014. Dedalo: Looking for clusters explanations in a labyrinth of linked data. In *European semantic web conference*. Springer, Cham, ., 333–348. https://doi.org/10.1007/978-3-319-07443-6_23

[121] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, Vol. 30. Curran Associates Inc., ., 5998–6008.

[122] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. In *International conference on learning representations*, Vol. 1050. ICLR, ., 20.

[123] Minh Vu and My T Thai. 2020. PGM-Explainer: Probabilistic graphical model explanations for graph neural networks. In *Advances in neural information processing systems*, Vol. 33. Curran Associates Inc., ., 12225–12235.

[124] Hongwei Wang, Fuzheng Zhang, Jialin Wang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. 2018. RippleNet: Propagating user preferences on the knowledge graph for recommender systems. In *ACM international conference on information and knowledge management*. ACM, ., 417–426. https://doi.org/10.1145/3269206.3271739

[125] Jihong Wang, Minnan Luo, Jundong Li, Yun Lin, Yushun Dong, Jin Song Dong, and Qinghua Zheng. 2023. Empower post-hoc graph explanations with information bottleneck: A pre-training and fine-tuning perspective. In *ACM SIGKDD conference on knowledge discovery and data mining*. ACM, ., 2349–2360. https://doi.org/10.1145/3580305.3599330

[126] Xiang Wang, Xiangnan He, Fuli Feng, Liqiang Nie, and Tat-Seng Chua. 2018. TEM: Tree-enhanced embedding model for explainable recommendation. In *World wide web conference*. ACM, ., 1543–1552. https://doi.org/10.1145/3178876.3186066

[127] Xiting Wang, Kunpeng Liu, Dongjie Wang, Le Wu, Yanjie Fu, and Xing Xie. 2022. Multi-level recommendation reasoning over knowledge graphs with reinforcement learning. In *ACM web conference*. ACM, ., 2098–2108. https://doi.org/10.1145/3485447.3512083

[128] Xiang Wang, Dingxian Wang, Canran Xu, Xiangnan He, Yixin Cao, and Tat-Seng Chua. 2019. Explainable reasoning over knowledge graphs for recommendation. In *AAAI conference on artificial intelligence*, Vol. 33. AAAI Press, ., 5329–5336. https://doi.org/10.1609/aaai.v33i01.33015329

[129] Xiang Wang, Yingxin Wu, An Zhang, Xiangnan He, and Tat-Seng Chua. 2021. Towards multi-grained explainability for graph neural networks. In *Advances in neural information processing systems*, Vol. 34. Curran Associates Inc., ., 1–10.

[130] Xiang Wang, Yingxin Wu, An Zhang, Xiangnan He, and Tat seng Chua. 2021. Causal screening to interpret graph neural networks.

[131] Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. 2016. Attention-based LSTM for aspect-level sentiment classification. In *Conference on empirical methods in natural language processing*. ACL, ., 606–615. https://doi.org/10.18653/v1/D16-1058

[132] Sarah Wiegreffe and Yuval Pinter. 2019. Attention is not not explanation. In *Conference on empirical methods in natural language processing and the International joint conference on natural language processing*. ACL, ., 11–20. https://doi.org/10.18653/v1/D19-1002

[133] Haoran Wu, Wei Chen, Shuang Xu, and Bo Xu. 2021. Counterfactual supporting facts extraction for explainable medical record based diagnosis with graph network. In *Conference of the North American chapter of the association for computational linguistics: Human language technologies*. ACL, ., 1942–1955. https://doi.org/10.18653/v1/2021.naacl-main.156

[134] Yikun Xian, Zuohui Fu, Shan Muthukrishnan, Gerard De Melo, and Yongfeng Zhang. 2019. Reinforcement knowledge graph reasoning for explainable recommendation. In *ACM SIGIR conference on research and development in information retrieval*. ACM, ., 285–294. https://doi.org/10.1145/3331184.3331203

[135] Yikun Xian, Zuohui Fu, Handong Zhao, Yingqiang Ge, Xu Chen, Qiaoying Huang, Shijie Geng, Zhou Qin, Gerard De Melo, Shan Muthukrishnan, et al. 2020. CAFE: Coarse-to-fine neural symbolic reasoning for explainable recommendation. In *ACM international conference on information & knowledge management*. ACM, ., 1645–1654. https://doi.org/10.1145/3340531.3412038

[136] Yikun Xian, Handong Zhao, Tak Yeon Lee, Sungchul Kim, Ryan Rossi, Zuohui Fu, Gerard De Melo, and Shan Muthukrishnan. 2021. EXACTA: Explainable column annotation. In *ACM SIGKDD conference on knowledge discovery & data mining*. ACM, ., 3775–3785. https://doi.org/10.1145/3447548.3467211

[137] Shangsheng Xie and Mingming Lu. 2019. Interpreting and understanding graph convolutional neural network using gradient-based attribution method.

[138] Yaochen Xie, Sumeet Katariya, Xianfeng Tang, Edward Huang, Nikhil Rao, Karthik Subbian, and Shuiwang Ji. 2022. Task-agnostic graph explanations. In *Advances in neural information processing systems*, Vol. 35. Curran Associates Inc., ., 12027–12039.

[139] Ping Xiong, Thomas Schnake, Michael Gastegger, Grégoire Montavon, Klaus Robert Muller, and Shinichi Nakajima. 2023. Relevant walk search for explaining graph neural networks. In *International conference on machine learning*. PMLR, ., 38301–38324.

[140] Hao Xu, Shengqi Sang, Herbert Yao, Alexandra I Herghelegiu, Haiping Lu, James T Yurkovich, and Laurence Yang. 2021. APRILE: Exploring the molecular mechanisms of drug side effects with explainable graph neural networks. *bioRxiv* 1 (2021). https://doi.org/10.1101/2021.07.02.450937

[141] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*. PMLR, ., 2048–2057.

[142] Makoto Yamada, Wittawat Jitkrittum, Leonid Sigal, Eric P Xing, and Masashi Sugiyama. 2014. High-dimensional feature selection by feature-wise kernelized Lasso. *Neural computation* 26, 1 (2014), 185–207. https://doi.org/10.1162/NECO_a_00537

[143] Zhitao Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. 2019. GNNExplainer: Generating explanations for graph neural networks. In *Advances in neural information processing systems*, Vol. 32. Curran Associates Inc., ., 9244–9255.

[144] Zhaoning Yu and Hongyang Gao. 2022. MotifExplainer: A motif-based graph neural network explainer.

[145] Hao Yuan, Jiliang Tang, Xia Hu, and Shuiwang Ji. 2020. XGNN: Towards model-level explanations of graph neural networks. In *ACM SIGKDD international conference on knowledge discovery & data mining*. ACM, ., 430–438. https://doi.org/10.1145/3394486.3403085

[146] Hao Yuan, Haiyang Yu, Shurui Gui, and Shuiwang Ji. 2022. Explainability in graph neural networks: A taxonomic survey. *IEEE transactions on pattern analysis and machine intelligence* 45, 5 (2022), 5782–5799. https://doi.org/10.1109/TPAMI.2022.3204236

[147] Hao Yuan, Haiyang Yu, Jie Wang, Kang Li, and Shuiwang Ji. 2021. On explainability of graph neural networks via subgraph explorations. In *International conference on machine learning*. PMLR, ., 12241–12252.

[148] Xunlin Zhan, Yinya Huang, Xiao Dong, Qingxing Cao, and Xiaodan Liang. 2022. PathReasoner: Explainable reasoning paths for commonsense question answering. *Knowledge-based systems* 235 (2022), 107612. https://doi.org/10.1016/j.knosys.2021.107612

[149] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. 2019. Self-attention generative adversarial networks. In *International conference on machine learning*. PMLR, ., 7354–7363.

[150] Shichang Zhang, Yozen Liu, Neil Shah, and Yizhou Sun. 2022. GStarX: Explaining graph neural networks with structure-aware cooperative games. In *Advances in neural information processing systems*, Vol. 35. Curran Associates Inc., ., 19810–19823.

[151] Yue Zhang, David Defazio, and Arti Ramesh. 2021. RelEx: A model-agnostic relational model explainer. In *AAAI/ACM conference on AI, ethics, and society*. ACM, ., 1042–1049. https://doi.org/10.1145/3461702.3462562

[152] Yuan Zhang, Xiaoran Xu, Hanning Zhou, and Yan Zhang. 2020. Distilling structured knowledge into embeddings for explainable and accurate recommendation. In *International conference on web search and data mining*. ACM, ., 735–743. https://doi.org/10.1145/3336191.3371790

[153] Xianrui Zheng, Chao Zhang, and Philip C Woodland. 2021. Adapting GPT, GPT-2 and BERT language models for speech recognition. In *IEEE automatic speech recognition and understanding workshop*. IEEE, ., 162–168. https://doi.org/10.1109/ASRU51503.2021.9688232

[154] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. 2016. Learning deep features for discriminative localization. In *IEEE conference on computer vision and pattern recognition*. IEEE, ., 2921–2929. https://doi.org/10.1109/CVPR.2016.319

[155] Yaxin Zhu, Yikun Xian, Zuohui Fu, Gerard De Melo, and Yongfeng Zhang. 2021. Faithfully explainable recommendation via neural logic reasoning. In *Conference of the North American chapter of the association for computational linguistics: Human language technologies*. ACL, ., 3083–3090. https://doi.org/10.18653/v1/2021.naacl-main.245

[156] Jacek M Zurada, Aleksander Malinowski, and Ian Cloete. 1994. Sensitivity analysis for minimization of input data dimension for feedforward neural network. In *IEEE international symposium on circuits and systems*, Vol. 6. IEEE, ., 447–450. https://doi.org/10.1109/ISCAS.1994.409622