Natural Language Counterfactual Explanations for Graphs Using Large Language Models

Flavio Giorgi
Sapienza University of Rome
Department of Computer Science
giorgi@di.uniroma1.it

Cesare Campagnano[§]
Pinecone, US
Sapienza University of Rome
Department of Computer, Control and Management Engineering
campagnano@diag.uniroma1.it

Fabrizio Silvestri
Sapienza University of Rome
Department of Computer, Control and Management Engineering
fsilvestri@diag.uniroma1.it

Gabriele Tolomei
Sapienza University of Rome
Department of Computer Science
tolomei@di.uniroma1.it

Abstract

Explainable Artificial Intelligence (XAI) has emerged as a critical area of research to unravel the opaque inner logic of (deep) machine learning models. Among the various XAI techniques proposed in the literature, counterfactual explanations stand out as one of the most promising approaches. However, these "what-if" explanations are frequently complex and technical, making them difficult for non-experts to understand and, more broadly, challenging for humans to interpret. To bridge this gap, in this work, we exploit the power of open-source Large Language Models to generate natural language explanations when prompted with valid counterfactual instances produced by state-of-the-art explainers for graph-based models. Experiments across several graph datasets and counterfactual explainers show that our approach effectively produces accurate natural language representations of counterfactual instances, as demonstrated by key performance metrics.

Proceedings of the 28[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2025, Mai Khao, Thailand. PMLR: Volume 258.

1 INTRODUCTION

In recent years, Machine Learning have become deeply embedded in various facets of our daily lives, influencing decisions ranging from personalized recommendations to critical judgments in healthcare and finance. This pervasive integration has raised significant concerns regarding transparency and accountability, leading to legislative actions such as the European Union's General Data Protection Regulation (GDPR) (Voigt and Von dem Bussche, 2017), which emphasizes the right to explanation for automated decision-making processes. Similarly, the upcoming EU Artificial Intelligence Act aims to regulate AI systems, mandating explainability and interpretability in high-risk applications (European Commission, 2021).

In the realm of eXplainable Artificial Intelligence (XAI), numerous techniques have been proposed to address this need. Among them, *counterfactual explanations* (Wachter et al., 2017) stand out as one of the most promising *post-hoc* methods. The core idea is to explain a model's prediction by identifying a *counterfactual example* – i.e., the minimal change to the input that leads to a different prediction. Counterfactual explanations have been successfully applied to unveil the inner workings of predictive models having various degrees of complexity, ranging from ensembles of decision trees (Tolomei et al., 2017; Tolomei and Silvestri, 2019) and multi-layer perceptrons (Montavon et al., 2018) to more sophisticated models like transformers

[§]Work done before joining Pinecone.

(Chefer et al., 2020) or model-agnostic techniques such as Movin et al. (2024); Chen et al. (2022b,a).

Despite these advancements, a significant challenge remains: translating algorithmically generated explanations into a "format" that is accessible and comprehensible to end-users who may not possess technical expertise. Fredes and Vitria (2024) takes a first step in this direction by leveraging the generative capabilities of Large Language Models (LLMs) to produce user-friendly counterfactual explanations in natural language.

Building on the work above, we tackle the more ambitious challenge of generating natural language explanations from counterfactual examples tailored for graph neural networks (GNNs). Indeed, counterfactual examples derived from graph inputs are inherently more complex than those from tabular data, given the intricate relationships and structures they capture (e.g., nodes, edges, and their dependencies). This complexity presents unique challenges that demand a specialized approach to translate them into universally comprehensible natural language explanations.

Moreover, graph data is prevalent across various domains such as financial decision-making (Cao et al., 2020), fraud detection (Wang et al., 2021), structured text extraction (Campagnano et al., 2022), and biology (Fout et al., 2017), where GNN-based models have shown remarkable predictive performance by encoding high-order structural relationships within their learned graph representations.

Specifically, we consider the counterfactual examples output by a generic graph counterfactual explainer designed for node classification tasks using GNNs. Then, we instruct several open-source LLMs to translate these "raw" counterfactual examples into coherent natural language explanations that are accessible also to non-expert users. To evaluate the quality of the generated explanations, we introduce a set of novel metrics that measure how accurately our method maps the counterfactual examples to their corresponding natural language descriptions. As openly generated text might be hard to evaluate using automatic evaluation metrics (Chiang and Lee, 2023), we also validate this evaluation through human judgment. Extensive experiments conducted using two graph counterfactual explainers for node classification – CF-GNNExplainer (Lucic et al., 2022) and its extension for node features, CF-GNNFeatures (Giorgi et al., 2024) – across several graph datasets and multiple open-source LLMs demonstrate that our method can effectively support decision-making processes in critical domains through the generation of natural language explanations.

To summarize, our main contributions are as follows:

- We present a method for translating counterfactual explanations for graphs into natural language using state-of-the-art open-source LLMs. This is, to our knowledge, the first work proposing the use of LLMs as means for converting the output of a GNN model to a human-readable format.
- We define novel metrics to properly assess the effectiveness of these explanations.
- We perform an extensive evaluation of our approach, which includes varying the sizes of LLMs, datasets, and explanation methods.

This paper is structured as follows: in Section 2, we summarize related work; In Section 4, we describe our method, which is validated through extensive experiments in Section 5. We discuss the practical implications of our method in Section 6. Finally, Section 7 concludes our work and proposes future directions.

The code to reproduce our experiments is available at https://github.com/flaat/llm-graph-cf.

2 RELATED WORK

Counterfactual explainability has emerged as a pivotal approach for interpreting complex machine learning models by illustrating how changes in input variables can lead to different outcomes. Despite this, a significant challenge persists: making these explanations accessible and understandable to a broad audience, particularly non-technical users who may struggle with abstract mathematical concepts and technical jargon. The intricacy of counterfactual explanations often hinders their comprehension among laypersons, limiting their practical utility in real-world applications where user trust and transparency are paramount.

The advent of large language models (LLMs), such as GPT-4(Achiam et al., 2023), Qwen2.5 (Qwen Team, 2024), LLama (Touvron et al., 2023), Mistral (Jiang et al., 2023), and theirs successors (Bacciu et al., 2024), has revolutionized the field of natural language processing. These models possess an unparalleled capacity for understanding and generating human-like text, making them invaluable tools for translating complex technical information into plain language. Their widespread availability and ease of integration into various platforms further enhance their appeal for tasks requiring sophisticated language generation and interpretation.

Leveraging the immense capabilities of LLMs presents a promising solution to the accessibility problem in counterfactual explainability. By converting intricate data-driven explanations into natural language narra-

tives, LLMs can make the insights derived from machine learning models more digestible for non-technical users. This translation not only aids in user comprehension but also fosters greater trust in automated systems by promoting transparency.

In this context, the pioneering work done by Fredes and Vitria (2024) marks a significant milestone. Their research laid the foundations for utilizing LLMs to transform data-based counterfactual explanations into coherent, user-friendly language. After generating the counterfactual examples, the researchers aimed to identify the primary causal factors deduced from these examples that led to the user's differing classification. To achieve this, they input both the set of counterfactual examples and the original user data into a Large Language Model (LLM), instructing it to produce a list of the main reasons why the user was classified differently. Once the LLM generated this list, they meticulously verified its accuracy and identified the most relevant causes contributing to the explanation.

Subsequently, they synthesized all the information produced and leveraged the LLM once more to generate a final explanation articulated in plain language. This explanation was crafted to emphasize actionable steps that the user could take to alter their input data or behavior, thereby changing their classification to the desired category. By doing so, they not only provided a transparent rationale behind the classification but also offered practical guidance for the user to achieve a favorable outcome. However, their work focuses solely on the relatively straightforward case of translating counterfactual examples derived from a single, well-known tabular dataset. Furthermore, they used a proprietary LLM model (GPT-4o) making it hard to replicate their approach.

To the best of the authors' knowledge, there are no papers in the State-of-the-Art that address the problem of translating the explanations generated via counterfactual explainers into natural language explanations; for this reason, we firmly believe that our work can be useful to the community.

3 BACKGROUND

Graph Neural Networks (GNNs) have emerged as a powerful class of machine learning models specifically designed to handle graph-structured data. Graphs are a natural representation for many real-world problems, such as social networks, recommendation systems, molecular chemistry, and knowledge graphs, where entities are represented as nodes and their relationships as edges. Unlike traditional neural networks, GNNs explicitly account for the relational structure of data, making them particularly effective for tasks like node classification, link prediction, and graph classification.

At the core of GNNs lies the concept of message passing or neighborhood aggregation, where nodes iteratively aggregate information from their neighbors to learn contextualized and high-order representations. The resulting node embeddings capture both the local structure and node attributes, enabling downstream tasks on graph data. This framework was formalized in seminal works like Graph Convolutional Networks (GCN) by Kipf and Welling (2017), which introduced a spectral perspective on graph convolutions.

Since the introduction of GCNs, several variants have been developed to address specific limitations or enhance capabilities. For example, Graph Attention Networks (GAT), introduced by Veličković et al. (2018), employ an attention mechanism to weigh the contributions of neighboring nodes dynamically. GraphSAGE by Hamilton et al. (2017) enables inductive learning by aggregating sampled neighbor information through mean, LSTM, or pooling operations. Xu et al. (2019) proposed Graph Isomorphism Network (GIN), which focused on designing a more expressive GNN capable of distinguishing graph structures that previous architectures could not. Finally, extensions like R-GCN by Schlichtkrull et al. (2018), address edge types and heterogeneous graph structures, which are common in knowledge graphs and multi-relational data.

The versatility of GNNs has led to their application across a range of domains, such as community detection and friend recommendations (Qiu et al. (2018)) in social networks, personalized suggestions based on complex user-item interaction graphs (Wang et al. (2019b)) in recommender systems, molecular property prediction, and drug discovery (Gilmer et al. (2017)) in biochemistry, and entity linking and relation extraction (Wang et al. (2019a)) from knowledge graphs. For a comprehensive survey on GNNs, we invite the reader to refer to Wu et al. (2021).

4 FROM COUNTERFACTUAL EXAMPLES TO NATURAL LANGUAGE EXPLANATIONS

This section begins by formalizing the counterfactual explanation problem in a classification task, with a particular focus on graph-structured inputs. It then introduces our proposed method, which generates counterfactual examples using a dedicated explainer and subsequently translates these examples into natural language explanations via a pre-trained large language model (LLM). Finally, the section presents a new evaluation framework that defines a suite of met-

rics—such as target node identification, counterfactual class identification, and target node feature and neighbor extraction—to quantitatively assess the quality and coherence of the generated natural language explanations.

4.1 The Counterfactual Explanation Problem

The counterfactual explanation problem in a classification task is described as follows. Given a sample x and a predictive model $f_{\boldsymbol{\theta}}$ parametrized by $\boldsymbol{\theta}$ – hereinafter referred to as *oracle* – the goal is to find a sample $x' \neq x$ such that $f_{\boldsymbol{\theta}}(x) \neq f_{\boldsymbol{\theta}}(x')$. This x' is called a *counterfactual example* for x. Among all potential counterfactual examples (if any), we assume the existence of a counterfactual example generator g, which takes as input the original instance x and returns a counterfactual x^*, where the distance $d(x, x^*)$ is minimized, or $\perp$ if no valid counterfactual example exists. The distance function $d(\cdot, \cdot)$ ensures that the counterfactual sample x^* remains as close as possible to the original factual sample x.

Note that, in this work, we focus on graph inputs. Specifically, each sample x and its counterfactual(s) x' can be represented as $G(V, E)$ and $G'(V', E')$, respectively. Therefore, a counterfactual example for an input graph G will be a new graph G', where either the node features, the structural links, or both differ from the original, while still satisfying the counterfactual criterion of altering the model's prediction. This introduces additional challenges as the modifications can affect both the node-level properties and the overall graph topology.

4.2 Proposed Method

We assume to have an oracle $f_{\boldsymbol{\theta}}$ for a node classification task, specifically a graph neural network trained on a given input graph. For any node instance x, we know both its predicted label $\hat{y}$, such that $f_{\boldsymbol{\theta}}(x) = \hat{y}$, and its optimal counterfactual example $x^* = g(x)$, which is generated by the counterfactual explainer g.

To generate the natural language explanation ($e(x^*)$) associated with the generic counterfactual example (x^*), we feed a pre-trained LLM m with the factual (x) and counterfactual (x^*) instances along with a specific prompt p, i.e., $e(x^*) = m(p, x, x^*)$.

One of the critical challenges of this approach is how to validate the quality of the generated natural language explanations through the LLM. In the following section, we introduce the new evaluation framework proposed in this work.

4.3 A New Evaluation Framework

Given the novelty of the problem we are tackling, to the best of our knowledge, no established quality metrics exist in the literature to rigorously evaluate explanations generated by LLMs for graph counterfactuals. The only commonly adopted approach is based on subjective human judgment, which may introduce variability and bias in the assessment process. To address this gap and ensure a more systematic evaluation, we have developed a suite of novel metrics that provide a quantitative and objective assessment of the language model's capability to articulate pairs of factual and counterfactual graphs into coherent and informative natural language explanations. For clarity, we denote the factual graph as G and the corresponding counterfactual graph as G'. To compute these metrics effectively, we structured our prompts to include a request for the language model to populate a predefined dictionary with essential graph information. This dictionary encompasses the target node of the classification task, its original class in the factual graph, its modified class in the counterfactual graph, the neighborhood of the target node in both G and G', and the set of features associated with the target node in each scenario.

By doing so, we aim to evaluate the language model's ability to discern and convey the critical elements of the graph structure and its transformations, thereby assessing the model's comprehension of how the changes in the graph influence the classification outcome for the target node. This framework allows us to objectively measure the performance of the language model in translating structural and attribute-based differences between G and G' into precise and meaningful natural language descriptions.

Target Node Identification (TNI) This metric assesses the ability of the LLM to accurately identify and reference the target node within the graph structure. Given the importance of the target node as the focal point of the classification task, correctly pinpointing it is crucial for generating valid explanations. Formally, given a graph $G(V, E)$, the target node $t \in V$, and a node $v \in V$ predicted by the LLM, we define the Target Node Identification (TNI) metric as:

$$\mathrm{TNI}(v) = \begin{cases} 1 & \text{if } v = t, \\ 0 & \text{otherwise.} \end{cases}$$

Counterfactual Class Identification (CCI) This metric evaluates the capacity of the LLM to correctly comprehend and express the change in class assignment of the target node from the factual graph G to

the counterfactual graph G'. Let $G' = (V, E')$ be the factual, such that $f_{\boldsymbol{\theta}}(G') = c'$ where c' is the counterfactual class of the target node t such that $f_{\boldsymbol{\theta}}(G') \neq f_{\boldsymbol{\theta}}(G)$, and let c_{LLM} be the counterfactual class predicted by the LLM, CCI is defined as:

$$\text{CCI}(c_{\text{LLM}}) = \begin{cases} 1 & \text{if } c_{\text{LLM}} = c', \\ 0 & \text{otherwise.} \end{cases}$$

Factual Target Node Feature (FTNF) This metric examines the LLM's ability to accurately recognize and describe the features associated with the target node in the factual graph G. Correctly identifying these features is essential, as they are pivotal in determining the node's initial classification. Let $G = (V, E)$ be the factual graphs. Let $t \in V$ be the target node, and $\mathbf{x}_t \in \mathbb{R}^d$ denote the feature vectors of node t in the factual graph G. We also define the vector of features predicted by the LLM for the target node t as $\mathbf{x}_{\mathbf{LLM}}$, the metric is computed as:

$$\text{FTNF}(\mathbf{x}_{\mathbf{LLM}}) = \begin{cases} 1 & \text{if } \mathbf{x}_{\mathbf{LLM}} = \mathbf{x}_t, \\ 0 & \text{otherwise.} \end{cases}$$

Counterfactual Target Node Feature (CFTNF) Similar to FTNF, this metric evaluates the LLM's capability to identify and articulate the features associated with the target node in the counterfactual graph G'. This metric is critical for assessing whether the LLM captures the differences in node attributes that lead to a shift in classification. Let $G' = (V', E')$ be the counterfactual graphs. Let $t \in V'$ be the target node, and $\mathbf{x}'_t \in \mathbb{R}^d$ denote the feature vectors of node t in the counterfactual graph G'. We also define the vector of features predicted by the LLM for the target node t as $\mathbf{x}_{\mathbf{LLM}}$, the metric is computed as:

$$\text{CFTNF}(\mathbf{x}_{\mathbf{LLM}}) = \begin{cases} 1 & \text{if } \mathbf{x}_{\mathbf{LLM}} = \mathbf{x}'_t, \\ 0 & \text{otherwise.} \end{cases}$$

Counterfactual Target Node Neighbors (CFTNN) This metric measures the LLM's capacity to correctly identify the set of neighboring nodes for the target node in the counterfactual graph G'. Understanding these neighbors in the counterfactual context is essential, as changes in the neighborhood structure may directly influence the target node's classification shift. Accurately capturing the neighbors in G' helps the LLM articulate how the local connectivity of the target node has been modified and how this alteration affects the node's role and classification within the network. Let $G'(V', E')$ be a counterfactual graph, let $t' \in V'$ be the target node in the graph G', and let $\mathcal{N}(t') = u \in V' \mid (t', u) \in E'$ denote the set of neighbors of the target node t. Given a set of neighbors $\mathcal{N}_{\text{LLM}}(t')$ predicted by the LLM for the target node t', we define the Counterfactual Target Node Neighbors (CFTNN) metric as:

$$\text{CFTNN}(\mathcal{N}_{\text{LLM}}(t')) = \begin{cases} 1 & \text{if } \mathcal{N}_{\text{LLM}}(t') = \mathcal{N}(t'), \\ 0 & \text{otherwise.} \end{cases}$$

Overall, these metrics provide a comprehensive framework for evaluating the language model's capacity to interpret and describe the structural and feature-based transformations between factual and counterfactual graphs. By assessing these distinct aspects of graph understanding, we can quantitatively measure the model's ability to generate coherent and insightful explanations that accurately reflect the graph dynamics and their implications on classification outcomes. We decided to select the explanations that score 1 on at least 5 out of 6 metrics and test them to human judgment.

5 EXPERIMENTS

Given the general framework described in Section 4, this section presents the experimental setup and evaluation of our approach for generating counterfactual explanations in graph neural networks using large language models (LLMs). Specifically, we investigate the problem of translating counterfactual modifications into human-interpretable natural language descriptions.

To comprehensively assess the effectiveness and generalizability of our method, we conduct experiments on two distinct tasks: node classification and graph classification. In the node classification setting, the goal is to generate counterfactual explanations that describe changes necessary to alter the classification of an individual node within a larger graph. In contrast, in the graph classification task, counterfactuals are generated at the graph level, identifying modifications that would change the overall label assigned to the entire graph. To ensure a robust evaluation, we employ real-world datasets widely used in graph-based machine learning.

5.1 Node Classification Graph Counterfactual Explainers

As outlined in Section 4, our proposed pipeline requires the integration of a counterfactual explainer for node classification, denoted as g. In this study, we employed two state-of-the-art explainers, namely, CF-GNNExplainer and CF-GNNFeatures, allowing us to investigate the impact of distinct graph modifications on the LLM's ability to generate natural language explanations. Specifically, CF-GNNExplainer modifies

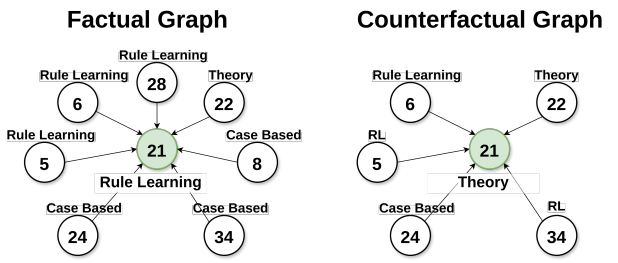

Figure 1: On the left, the factual graph. On the right, the counterfactual graph computed using CF-GNNExplainer. Each graph contains nodes ids and classes

the graph structure by perturbing the adjacency matrix, while CF-GNNFeatures focuses on altering the node attributes. This dual approach enables us to assess how variations in both structural and feature-based properties influence the quality and coherence of the generated explanations.

CF-GNNExplainer is a perturbation-based counterfactual explainer. It defines $\bar{\mathbf{A}}_v = \mathbf{P} \odot \mathbf{A}_v$, where $\mathbf{P}$ is a binary perturbation matrix that sparsifies $\mathbf{A}_v$. The goal is to find $\mathbf{P}$ for a given node v such that $f_{\boldsymbol{\theta}}(\mathbf{A}_v, x) \neq f_{\boldsymbol{\theta}}(\mathbf{P} \odot \mathbf{A}_v, x)$. To find $\mathbf{P}$, CF-GNNExplainer exploits a technique to train sparse neural networks to zero out entries in the adjacency matrix (i.e., removing edges). This results in the deletion of the edge between node i and node j.

CF-GNNFeatures is a node features perturbation-based counterfactual explainer. Given a graph $G(V, E)$, two matrices are defined, namely: $\mathbf{V}_x$, the node features matrix representing the features for every node in G and $\mathbf{P}_x$, the feature perturbation matrix. Initially, the matrix $\mathbf{P}_x$ is filled with ones to maintain the original sets of attributes. Given an oracle $f_{\boldsymbol{\theta}}$, parameterized by θ, we fix all the weights and train $\mathbf{P}_x$ to change the attribute matrix $\mathbf{V}_x$ that is fed into the oracle multiplying the current feature matrix with the perturbation matrix. In Figure 1 you can see an example of a factual and its counterfactual graph computed using CF-GNNExplainer.

5.2 Large Language Models

In this study, we utilized a family of state-of-the-art LLMs, namely Qwen2.5 (Qwen Team, 2024), to generate natural language explanations. We experimented with different model sizes in terms of parameters, specifically employing the 0.5B, 1.5B, 3B, 7B, and 14B variants.

The model configuration used in this work includes several hyperparameters that influence its behavior during text generation:

- temperature: 0.1,
- top-p: 0.8,
- top-k: 30,
- repetition penalty: 1.05,
- max output tokens: 2048,
- top-k: 10.

In order to reduce the memory footprint of the models, we used GPTQ (Frantar et al., 2023) quantization with 4-bit integer precision (`int4`).

Overall, the chosen configuration allows the model to generate meaningful counterfactual explanations for complex graph structures while maintaining computational efficiency. The setup leverages the power of modern LLMs along with advanced quantization techniques to produce high-quality outputs, making it an ideal choice for applications requiring detailed textual descriptions of graph perturbations.

Prompting The design and structure of the prompt is critical, as the performances of LLMs are highly influenced by how the graph data is presented. To this end, we adopted the incident representation framework proposed by Fatemi et al. (2024), with some adjustments to better suit the requirements of our task. To provide all the information needed for the LLM, we use an initial system prompt providing the essential background information on the challenges of counterfactual explainability, particularly in the context of graph-based data. Subsequently, the counterfactual prompt is introduced (Figure 2), which instructs the LLM to generate a coherent and contextually appropriate natural language explanation based on both the factual and counterfactual graph samples.

5.3 Experimental Setup

The experiments have been carried out on a machine equipped with 64GB of RAM, an Nvidia RTX 4090, and a AMD Ryzen 9 7900 processor. As oracle we used a 2-layer GCN trained for 500 epochs with a learning rate of 0.001.

5.4 Datasets

To ensure a comprehensive evaluation, we employed two well-known citation network datasets: Cora and CiteSeer. The CiteSeer dataset comprises 3,312 scientific publications categorized into six distinct classes,

# Parameters	TNI ↑	CCI ↑	FTNF ↑	CFTNF ↑	FTNN ↑	CFTNN ↑
0.5B	0.000	0.000	0.000	0.000	0.000	0.000
1.5B	0.048	0.048	0.022	0.007	0.030	0.030
3B	0.391	0.336	0.092	0.074	0.221	0.258
7B	0.292	0.284	0.162	0.085	0.007	0.007
14B	**0.720**	**0.720**	**0.668**	**0.565**	**0.528**	**0.524**

Table 1: Evaluation metrics using CF-GNNFeatures and Cora dataset for graph understanding.

# Parameters	TNI ↑	CCI ↑	FTNF ↑	CFTNF ↑	FTNN ↑	CFTNN ↑
0.5B	0.000	0.000	0.000	0.000	0.000	0.000
1.5B	0.053	0.053	0.039	0.026	0.000	0.013
3B	0.237	0.197	0.105	0.132	0.066	0.105
7B	0.171	0.118	0.053	0.118	0.026	0.000
14B	**0.618**	**0.618**	**0.579**	**0.579**	**0.539**	**0.500**

Table 2: Evaluation metrics using CF-GNNExplainer and Cora dataset for graph understanding.

Given the factual graph: ***$factual graph description$*** and given the counterfactual example: ***$counterfactual graph description$*** and given the knowledge base about the dataset: ***$dataset knowledge$***, fill the dictionary and provide an explanation about the change in classification for the target node, please evaluate also the influences of neighbors nodes.

Figure 2: Prompt example to get the explanations

connected by 4,732 citation links. Each publication is represented by a binary word vector, where each entry indicates the absence or presence of a specific word from a dictionary of 3,703 unique terms. Similarly, the Cora dataset contains 2,708 scientific publications classified into seven distinct classes, with 5,429 citation links. Each publication is also represented by a binary word vector corresponding to a dictionary of 1,433 unique terms.

After the counterfactuals had been found, we translated the original node feature vocabulary using actual words rather than binary vectors to facilitate the language model's understanding of the relationship between a node's classification and its features. Since predefined vocabularies for node features are not provided for these datasets, we adhered to the original feature extraction instructions to generate the vocabularies for both datasets.

5.5 Results

To provide an appropriate evaluation, we tested multiple counterfactual methods and LLMs on different graph datasets. As shown in Tables from 1 to 4 the results for the graph understanding as defined in Section 4 are generally good for models with more parameters. In particular, a clear trend emerges across all tables: as the number of parameters in the LLM increases, performance improves significantly across all metrics, as expected. The smallest model, with 0.5 billion parameters, consistently scores zero across all metrics and datasets, indicating its inability to generate meaningful explanations. With an increase to 1.5 billion parameters, there is a minimal improvement, but performance remains substantially low.

Notable improvements are observed with the 3-billion-parameter model, especially in metrics like Target Node Identification (TNI) and Counterfactual Class Identification (CCI). However, it's the largest model, with 14 billion parameters, that achieves the highest scores across all metrics and datasets. This demonstrates the importance of model size when dealing with complex tasks such as generating natural language explanations from graph counterfactuals.

In Table 1, results show that using CF-GNNFeatures on the Cora dataset, the TNI metric increases from 0.000 with the smallest model to 0.720 with the largest model. Similarly, in Table 3, using the CiteSeer dataset, the TNI metric reaches 0.879 with the 14-billion-parameter model. These improvements indicate that larger models can understand and generate accurate descriptions of complex graph structures.

# Parameters	TNI ↑	CCI ↑	FTNF ↑	CFTNF ↑	FTNN ↑	CFTNN ↑
0.5B	0.000	0.000	0.000	0.000	0.000	0.000
1.5B	0.273	0.280	0.096	0.003	0.174	0.174
3B	0.488	0.394	0.112	0.037	0.320	0.348
7B	0.450	0.447	0.180	0.171	0.314	0.314
14B	**0.879**	**0.873**	**0.705**	**0.481**	**0.758**	**0.758**

Table 3: Evaluation metrics using CF-GNNFeatures and Citeseer dataset for graph understanding

# Parameters	TNI ↑	CCI ↑	FTNF ↑	CFTNF ↑	FTNN ↑	CFTNN ↑
0.5B	0.000	0.000	0.000	0.000	0.000	0.000
1.5B	0.000	0.038	0.000	0.000	0.000	0.000
3B	0.385	0.192	0.038	0.077	0.115	0.115
7B	0.385	0.346	0.154	0.269	0.077	0.038
14B	**0.615**	**0.615**	**0.346**	**0.346**	**0.577**	**0.538**

Table 4: Evaluation metrics using CF-GNNExplainer and Citeseer dataset for graph understanding.

Comparing performances across datasets The LLMs generally perform better on the CiteSeer dataset, especially when using the CF-GNNFeatures explainer. For example, in Table 3 (CF-GNNFeatures on CiteSeer), the 14-billion-parameter model achieves a TNI of 0.879, higher than the 0.720 achieved on the Cora dataset in Table 1.

Several factors contribute to this difference in performance. The datasets have different levels of complexity, differences in feature distributions, or inherent properties that make one more amenable to the LLM's processing capabilities. CiteSeer have more straightforward or more distinctive features that the LLM can more readily associate with classification outcomes, aiding in generating coherent explanations.

Comparison between explainers The two counterfactual explainers used in the study, CF-GNNFeatures and CF-GNNExplainer, focus on different aspects of the graph. CF-GNNFeatures modifies node features, while CF-GNNExplainer modifies the graph structure.

For instance, in Table 1 (CF-GNNFeatures on Cora), the FTNF (Factual Target Node Feature) and CFTNF (Counterfactual Target Node Feature) scores at 14 billion parameters are 0.668 and 0.565, respectively. In contrast, in Table 2 (CF-GNNExplainer on Cora), these scores are lower, at 0.579 for both metrics at the same model size.

This suggests that LLMs find it easier to generate explanations when the modifications involve changes in node features rather than structural changes in the graph. Explaining structural changes may require a deeper understanding of the graph's topology and how it influences the classification, which appears more challenging for the LLMs.

Question	CF-GNNE.	CF-GNNF.
Q1	3.00 ± 1.60	4.44 ± 0.80
Q2	2.87 ± 1.47	4.28 ± 0.83
Q3	2.90 ± 1.51	3.56 ± 1.23
Q4	2.92 ± 1.52	3.68 ± 1.20
Q5	3.00 ± 1.55	3.80 ± 1.11

Table 5: Average scores (1 to 5) from the human evaluation for the explanations generated using Qwen2.5-14B using counterfactuals from CF-GNNExplainer (CF-GNNE.) and CF-GNNFeatures (CF-GNNF) on the Cora dataset

Human Evaluation In order to appropriately assess the explanations, we conducted a human evaluation study involving a sample of 15 graduate students with varied skills and backgrounds. Participants were asked to complete a questionnaire designed to evaluate the quality and clarity of the counterfactual explanations. The questionnaire consisted of the following five questions, each rated on a scale from 1 to 5:

- Q1: Is the terminology and language used in the explanation appropriate and easy to understand?
- Q2: How clear and easy to understand is the provided counterfactual explanation?
- Q3: How clearly does the explanation describe the changes in node connections (graph structure) that led to the counterfactual outcome?
- Q4: Are the changes in features and structure

easy to interpret and make sense in the context of the original graph?

- Q5: What is your overall assessment of the clarity and coherence of the counterfactual explanation?

The target node (node 25) was originally classified as **Genetic Algorithms** with a rich set of attributes. By removing certain attributes like *'adapt', 'constrain', 'audio', 'interoper', 'dataflow', 'visibl', 'see', 'gateway', 'realtim', 'exploratori', 'skill', 'almost', 'adversari', 'confid', and 'chapter'*, the node's representation became simpler and more aligned with the characteristics of nodes classified as **Neural Networks**. Additionally, the neighboring nodes (24 and 41) also changed their classification from **Genetic Algorithms** to **Neural Networks**, further influencing the reclassification of node 25.
The original target node features were: *['schedul', 'adapt', 'constrain', 'audio', 'interoper', 'account', 'dataflow', 'gestur', 'light', 'visibl', 'see', 'intrus', 'gateway', 'realtim', 'exploratori', 'skill', 'almost', 'adversari', 'confid', 'oracl', 'gigabit', 'chapter']*

Figure 3: Response generated using Qwen2.5-14B using as counterfactual generator CF-GNNFeatures.

The results of the human evaluation can be seen in Table 5. An example of natural language trnaslation can be seen in Figure 3. The findings indicate that explanations generated by the LLM that focus on node features are more effective and meaningful to human users than those based on adjacency matrix perturbations. Participants found feature-based explanations to be more accessible in terms of language, clearer in conveying the reasoning, and more interpretable within the context of the graph.

6 PRACTICAL IMPLICATIONS

The proposed framework for generating natural language counterfactual explanations using LLMs holds significant promise for enhancing interpretability and transparency in graph-based machine learning models. This approach has several practical implications across domains where complex graph structures are used, such as financial analysis, healthcare, cybersecurity, and social network analysis. By translating complex counterfactual explanations into natural language, our method makes these explanations more accessible to non-technical stakeholders, including business managers, policymakers, and end-users. Moreover, with the European Union's General Data Protection Regulation (GDPR) and the EU Artificial Intelligence Act, explainability and transparency of AI systems are becoming mandatory, particularly in high-stakes applications. Our method addresses this requirement by ensuring that counterfactual explanations are technically sound and easily interpretable, thereby aiding compliance with legal and ethical standards. In summary, our framework provides a comprehensive solution for bridging the gap between technical counterfactual explanations and human comprehension, enabling broader adoption of AI technologies in high-stakes, complex domains. The method promotes transparency and trust and enhances the utility of counterfactual explanations as a tool for model evaluation, debugging, and refinement.

7 CONCLUSION AND FUTURE WORK

In this paper, we presented a novel framework for generating natural language counterfactual explanations for graph-based models using state-of-the-art LLMs. Our approach leverages the inherent generative capabilities of LLMs to transform complex and technical counterfactual examples into coherent and accessible natural language descriptions. By doing so, we address a critical gap in the literature: the lack of intuitive, user-friendly counterfactual explanations for GNNs. We validated our method across several GNN-based counterfactual explainers and multiple graph datasets, demonstrating its effectiveness through a suite of newly proposed metrics and human evaluation. Our findings show that our framework not only captures the underlying structural and attribute-based transformations within the graphs but also produces explanations that are easily interpretable by non-expert users, thereby enhancing the transparency and usability of graph-based machine learning models.

Our framework does not depend upon any particular counterfactual explainer, so it can be used with any approach. We developed additional metrics that quantify the quality of the natural language output—such as graph understanding. Finally, our experiments were conducted using open-source LLMs without task-specific fine-tuning. Future research could explore the impact of fine-tuning the LLMs on graph-specific tasks or developing domain-specific LLMs to further enhance the quality and relevance of the generated explanations.

Overall, we believe that our framework represents a significant step toward bridging the gap between complex algorithmic explanations and human comprehension in the domain of graph-based machine learning.

8 Acknowledgments

This work was partially supported by the following projects: FAIR (PE0000013), SERICS (PE00000014), and SoBigData.it (IR0000013) under the National Recovery and Resilience Plan funded by the European Union NextGenerationEU; HyperKG – Hybrid Prediction and Explanation with Knowledge Graphs (2022Y34XNM) and NEREO – Neural Reasoning Over Open Data (2022AEFHA) funded by the Italian Ministry of University and Research under the PRIN 2022 program; GHOST – Protecting User Privacy from Community Detection in Social Networks (RG124190FD55EB57) funded by Sapienza University of Rome - "Progetti di Ricerca Grandi"; SEEDS – Sustainable Ecosystems in Evolving Digital Societies funded by Sapienza University of Rome - "Progetti Dipartimentali".

References

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al. (2023). Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Bacciu, A., Campagnano, C., Trappolini, G., and Silvestri, F. (2024). DanteLLM: Let's push Italian LLM research forward! In Calzolari, N., Kan, M.-Y., Hoste, V., Lenci, A., Sakti, S., and Xue, N., editors, *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 4343–4355, Torino, Italia. ELRA and ICCL.

Campagnano, C., Conia, S., and Navigli, R. (2022). SRL4E – Semantic Role Labeling for Emotions: A unified evaluation framework. In Muresan, S., Nakov, P., and Villavicencio, A., editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4586–4601, Dublin, Ireland. Association for Computational Linguistics.

Cao, D., Wang, Y., Duan, J., Zhang, C., Zhu, X., Huang, C., Tong, Y., Xu, B., Bai, J., Tong, J., et al. (2020). Spectral temporal graph neural network for multivariate time-series forecasting. *Advances in neural information processing systems*, 33:17766–17778.

Chefer, H., Gur, S., and Wolf, L. (2020). Transformer interpretability beyond attention visualization. arxiv.

Chen, Z., Silvestri, F., Tolomei, G., Wang, J., Zhu, H., and Ahn, H. (2022a). Explain the explainer: Interpreting model-agnostic counterfactual explanations of a deep reinforcement learning agent. *IEEE Transactions on Artificial Intelligence*, 5(4):1443–1457.

Chen, Z., Silvestri, F., Wang, J., Zhu, H., Ahn, H., and Tolomei, G. (2022b). Relax: Reinforcement learning agent explainer for arbitrary predictive models. In *Proceedings of the 31st ACM international conference on information & knowledge management*, pages 252–261.

Chiang, C.-H. and Lee, H.-y. (2023). Can large language models be an alternative to human evaluations? In Rogers, A., Boyd-Graber, J., and Okazaki, N., editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15607–15631, Toronto, Canada. Association for Computational Linguistics.

European Commission (2021). Proposal for a regulation of the european parliament and of the council laying down harmonised rules on artificial intelligence (Artificial Intelligence Act). COM/2021/206 final.

Fatemi, B., Halcrow, J., and Perozzi, B. (2024). Talk like a graph: Encoding graphs for large language models. In *The Twelfth International Conference on Learning Representations*.

Fout, A., Byrd, J., Shariat, B., and Ben-Hur, A. (2017). Protein interface prediction using graph convolutional networks. *Advances in neural information processing systems*, 30.

Frantar, E., Ashkboos, S., Hoefler, T., and Alistarh, D. (2023). Gptq: Accurate post-training quantization for generative pre-trained transformers.

Fredes, A. and Vitria, J. (2024). Using llms for explaining sets of counterfactual examples to final users. *arXiv preprint arXiv:2408.15133*.

Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and Dahl, G. E. (2017). Neural message passing for quantum chemistry. In *International Conference on Machine Learning (ICML)*, pages 1263–1272.

Giorgi, F., Silvestri, F., and Tolomei, G. (2024). Generate counterfactual explanations for graph neural networks from node feature perturbations. *TechRxiv preprint*.

Hamilton, W. L., Ying, R., and Leskovec, J. (2017). Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Jiang, A., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D., de las Casas, D., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., et al. (2023). Mistral 7b (2023). *arXiv preprint arXiv:2310.06825*.

Kipf, T. N. and Welling, M. (2017). Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*.

Lucic, A., Ter Hoeve, M. A., Tolomei, G., De Rijke, M., and Silvestri, F. (2022). Cf-gnnexplainer: Counterfactual explanations for graph neural networks. In *International Conference on Artificial Intelligence and Statistics*, pages 4499–4511. PMLR.

Montavon, G., Samek, W., and Müller, K.-R. (2018). Methods for interpreting and understanding deep neural networks. *Digital signal processing*, 73:1–15.

Movin, M., Siciliano, F., Ferreira, R., Silvestri, F., and Tolomei, G. (2024). Consistent counterfactual explanations via anomaly control and data coherence. *IEEE Transactions on Artificial Intelligence*.

Qiu, J., Tang, J., Ma, H., Dong, Y., Wang, K., and Tang, J. (2018). Deepinf: Social influence prediction with deep learning. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*, pages 2110–2119. ACM.

Qwen Team, Q. T. (2024). Qwen2.5: A party of foundation models.

Schlichtkrull, M., Kipf, T. N., Bloem, P., van den Berg, R., Titov, I., and Welling, M. (2018). Modeling relational data with graph convolutional networks. In *European Semantic Web Conference (ESWC)*, pages 593–607. Springer.

Tolomei, G. and Silvestri, F. (2019). Generating actionable interpretations from ensembles of decision trees. *IEEE Transactions on Knowledge and Data Engineering*, 33(4):1540–1553.

Tolomei, G., Silvestri, F., Haines, A., and Lalmas, M. (2017). Interpretable predictions of tree-based ensembles via actionable feature tweaking. In *Proc. of KDD '17*, pages 465–474. ACM.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. (2023). Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., and Bengio, Y. (2018). Graph attention networks. In *International Conference on Learning Representations (ICLR)*.

Voigt, P. and Von dem Bussche, A. (2017). The eu general data protection regulation (gdpr). *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, 10(3152676):10–5555.

Wachter, S., Mittelstadt, B., and Russell, C. (2017). Counterfactual explanations without opening the black box: Automated decisions and the GDPR. *Harvard Journal of Law & Technology*, 31(2):841–888.

Wang, J., Zhang, S., Xiao, Y., and Song, R. (2021). A review on graph neural network methods in financial applications. *arXiv preprint arXiv:2111.15367*.

Wang, X., He, X., Cao, Y., Liu, M., and Chua, T.-S. (2019a). Kgat: Knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*, pages 353–361. ACM.

Wang, X., Zhang, F., Zhao, M., Li, W., Xie, X., and Guo, M. (2019b). Neural graph collaborative filtering. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 165–174. ACM.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Yu, P. S. (2021). A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1):4–24.

Xu, K., Hu, W., Leskovec, J., and Jegelka, S. (2019). How powerful are graph neural networks? In *International Conference on Learning Representations (ICLR)*.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model.
   Yes, we described the framework in Section 3
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm.
   Not Applicable, the paper does not propose any kind of new algorithm or model
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries.
   Yes, we provide the anonymized code in Section 4 under the subsection 4.3 Experimental Setup

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results.
   Not Applicable
   (b) Complete proofs of all theoretical results.
   Not Applicable

(c) Clear explanations of any assumptions.
Not Applicable

3. For all figures and tables that present empirical results, check if you include:

(a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL).
Yes, we provide the anonymized code in Section 4 under the subsection 4.3 Experimental Setup to reproduce the results

(b) All the training details (e.g., data splits, hyperparameters, how they were chosen).
Yes, we provide models hyperparameters

(c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times).
Yes, we provide details about it in section 4

(d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider).
Yes, we provide the hardware se used to conduct the experiments

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:

(a) Citations of the creator If your work uses existing assets.
Yes, we cited all the creators of the code we used

(b) The license information of the assets, if applicable.
Not Applicable

(c) New assets either in the supplemental material or as a URL, if applicable.
Not Applicable

(d) Information about consent from data providers/curators.
Not Applicable

(e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content.
Not Applicable

5. If you used crowdsourcing or conducted research with human subjects, check if you include:

(a) The full text of instructions given to participants and screenshots.
Yes, we provided the questions we fed to the human subjects during the human evaluation

(b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable.
Not Applicable

(c) The estimated hourly wage paid to participants and the total amount spent on participant compensation.
Not Applicable