From Nodes to Narratives: Explaining Graph Neural Networks with LLMs and Graph Context

Peyman Baghershahi[1]* **Gregoire Fournier**[1]* **Pranav Nyati**[2] **Sourav Medya**[1]
[1]University of Illinois Chicago,
[2]Indian Institute of Technology Kharagpur
[1]{pbaghe2, gfourn2, medya}@uic.edu; [2]pranavnyati26@kgpian.iitkgp.ac.in

Abstract

Graph Neural Networks (GNNs) have emerged as powerful tools for learning over structured data, including text-attributed graphs (TAGs), which are common in domains such as citation networks, social platforms, and knowledge graphs. GNNs are not inherently interpretable and thus, many explanation methods have been proposed. However, existing explanation methods often struggle to generate interpretable, fine-grained rationales, especially when node attributes include rich natural language. In this work, we introduce GSPELL, a lightweight, post-hoc framework that uses large language models (LLMs) to generate faithful and interpretable explanations for GNN predictions. GSPELL projects GNN node embeddings into the LLM embedding space and constructs hybrid prompts that interleave soft prompts with textual inputs from the graph structure. This enables the LLM to reason about GNN internal representations and to produce natural-language explanations, along with concise explanation subgraphs. Our experiments across real-world TAG datasets demonstrate that GSPELL achieves a favorable trade-off between fidelity and sparsity, while improving human-centric metrics such as insightfulness. GSPELL sets a new direction for LLM-based explainability in graph learning by aligning GNN internals with human reasoning.

1 Introduction

Graph neural networks (GNNs) have witnessed rapid adoption across a wide range of critical real-world applications, including healthcare (Zitnik et al., 2018, 2019), drug design (Xiong et al., 2021; Liu et al., 2022; Sun et al., 2020), recommender systems (Chen et al., 2022), and fraud detection (Rao et al., 2021). The high-stakes nature of these domains demands that the predictions made by GNNs be not only accurate but also trustworthy. One powerful approach to building trust in deep learning models is to provide meaningful explanations that justify their predictions (Shneiderman, 2020). Such explanations, which may highlight important substructures in the input (Luo et al., 2020; Schlichtkrull et al., 2021; Yuan et al., 2021; Armgaan et al., 2024), or offer counterfactual scenarios (Lucic et al., 2022a; Tan et al., 2022a; Verma et al., 2024; Huang et al., 2023; Fournier and Medya, 2025). However, generating explanations for GNNs is inherently challenging due to the combinatorial nature of graph data and the joint influence of node attributes and edge connections. These complexities make GNN explainability a non-trivial problem and have led to the development of a wide array of explanation techniques (Kakkad et al., 2023; Guo et al., 2023).

Need for LLM-based GNN explainers. There has been a growing interest in the integration of GNNs and LLMs to enhance GNN task performance on text-attributed graphs (TAGs). Essentially, these methods leverage LLMs in different architectural orders to 1) attain rich representations of the input TAGs, task descriptions, and task-specific few-shot examples for the GNN predictors (Liu et al., 2024a; Fang et al., 2024), or 2) reason over the TAGs directly (Fatemi et al., 2024) or along with GNN-enhanced soft prompts to make predictions (Chen et al., 2024a; Tang et al., 2024). However, there is a lack of focus on using LLMs to explain GNN predictions and clarify the reasoning behind these black-box models.

Traditional GNN explainers aim to identify influential subgraphs or feature subsets that contribute to predictions (Ying et al., 2019a; Luo et al., 2020). While effective for graphs with simple node features, these methods perform poorly on TAGs, where node information is expressed in natural language. Moreover, these explanations are often presented as subgraphs that are not human-interpretable to users. LLMs, on the other hand, ex-

*Equal contribution

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 41975–41994

cel at reasoning over textual content and can generate coherent, human-interpretable rationales. Thus, their integration into GNN explanation pipelines can improve the interpretability of the GNN outputs on TAGs.

Limitations of Existing LLM-based GNN Explainers. The existing frameworks (detailed in Sec. 2) employing LLMs for explaining GNN have the following shortcomings: *(1) Template dependency and alignment complexity:* Aligning the GNN explainer's output with the LLM acceptable input necessitates rigid templates, handcrafted scores, or training/fine-tuning of the GNN explainer. *(2) Missed GNN internals:* Existing approaches fail to directly leverage the rich internal representations of the GNN, making the explanations generic or unfaithful to the internal working of the GNNs. *(3) Bias from suboptimal explainers:* Invoking an external GNN explainer can bias the reasoning and judgment of the LLM. This is critical when the GNN explainers are suboptimal, and they signal noisy information to LLM, while the LLM can potentially better infer the pivotal causes of why the GNN has made specific predictions, particularly for TAGs where the textual attributes carry rich information about the graph entities.

Our Contributions. In this work, we present GSPELL (GNN Soft Prompted Explanation with LLMs), a lightweight, post-hoc explanation framework for TAGs that integrates the representational power of GNNs with the reasoning capability of LLMs. Our contributions are as follows:

- *Novel Method*: We introduce a method that bypasses traditional GNN explainer modules by using LLMs as direct interpreters of model behavior. GSPELL generates fine-grained, natural language explanations (NLEs) unreliant on handcrafted templates or perturbation-based saliency masks, reducing external explainers' bias.
- *Soft Prompt Integration*: GSPELL aligns the internal representations of a trained GNN with the token-level embedding space of an LLM via a novel embedding projector. This projection enables GNN features to be treated as soft prompts in a hybrid prompt that blends structural and textual cues, allowing the LLM to reason over the graph's latent space.
- *Interpretable Explanation Subgraphs*: Unlike prior methods that return subgraph masks, GSPELL produces interpretable NLE subgraphs of node-level support decisions.
- *Training-Agnostic and Plug-and-Play Deployment*: GSPELL does not require fine-tuning the GNN or the LLM, making it easily deployable on pre-trained models without retraining. This makes it adaptable for real-world applications.

2 Related Work

We describe the related studies on the integrated LLM and GNN explainer methods. The explanation capability of LLMs has been applied to graph problems in task-oriented contexts (Li et al., 2025; Ma et al., 2024). There have been attempts to guide GNNs using explanations generated by LLMs; for example, in (He et al., 2024a), an LLM generates node predictions and textual explanations of the reasoning behind them.

LLM explanation to enhance explainers. LLMs have been utilized to enhance the explanations of another GNN explainers, for example (Zhang et al., 2024) uses LLMs to evaluate explanations of an explainer and its scores guide the weighted gradients used for optimizing the explainer through Bayesian Variation Inference to address learning bias. Similarly, to improve the understandability of explanations, (Pan et al., 2024) first trains a pseudo-label generator LLM that takes explanations from an external GNN explainers and generates pseudo-labels for them. The generator is tuned in an expert iteration procedure with tailored objectives for faithfulness and brevity, and the optimized generator generates pseudo-labels to fine-tune an explainer LLM to build end-to-end models.

LLM explanation to enhance understanding. One advantage of LLMs is that they can generate NLE for GNNs, which are more human-interpretable. (He et al., 2024b) proposes a method for generating counterfactual explanations, which uses an autoencoder to construct counterfactual graph topologies from LLM-generated counterfactual text pairs. To mitigate hallucinations, it employs a dynamic feedback mechanism that prompts the LLM to refine its initial outputs. Lately, (Cedro and Martens, 2025) proposed prompting an LLM to narrate the explanation subgraphs and the feature importance matrices generated by a GNN explainer, which describe the relationships between neighboring nodes. Our work produces representation-aware LLM explanations without an external explainer. *We present additional related work in Appendix A.*

3 Problem formulation

We define a graph as $G = (V, E, X)$, where V is the set of nodes, $E = \{(u, v) \mid u, v \in V\}$ is the set of edges, and $X = \{x_v \mid v \in V\}$ is the node feature matrix, with $x_v \in \mathbb{R}^d$ representing the feature vector of node v. We denote by $\mathcal{G}$ the set of graphs in a dataset, and by $\mathcal{V}$ the set of all nodes across the graphs in $\mathcal{G}$.

Definition 1 (Text-Attributed Graph (TAG))**.** *A TAG is defined as a graph*

$$G = (V, E, T)$$

where V is the set of nodes, $E \subseteq V \times V$ is the set of edges (undirected or directed), $T : V \to \mathcal{D}$ is a function that assigns a textual document or sequence to each node, where $\mathcal{D}$ is a corpus of natural language texts (e.g., sentences, paragraphs).

Each node $v \in V$ is thus associated with a text document $T(v)$, and the graph structure E represents semantic, relational, or structural links between nodes. TAGs enable learning from both the textual content and the graph's topology. In this work, we focus on the node classification task.

Node Classification. Given a dataset $\mathcal{G}$ and a graph $G = (V, E, T) \in \mathcal{G}$, where every node $v \in V$ has a label $y_v \in \mathcal{Y}$ belonging to one of C classes ($\mathcal{Y} = \{Y_i\}_{i=1}^{C}$), the node classification task consists in training a GNN Φ such that $\Phi(x_v, G) = y_v \quad \forall G \in \mathcal{G}, \forall v \in V$, and $x_v = f_e(T(v))$, where f_e is a text encoder.

Our method aims to explain the node classification predictions of a GNN using local subgraphs, i.e., small subgraphs around the node being classified (Ying et al., 2019a).

Definition 2 (Local Factual Explainer)**.** *Given a GNN Φ and a graph $G = (V, E, T) \in \mathcal{G}$, a local factual explainer Ψ is a mapping from $(\mathcal{V}, \mathcal{G})$ to $\mathcal{S}_{\mathcal{V}}(\mathcal{G})$ associated to $\mathcal{V}' \subseteq \mathcal{V}$ verifying:*

$$\Phi(x_v, \Psi(v, G)) = \Phi(x_v, G)\ \forall G \in \mathcal{G}, v \in V \cap \mathcal{V}' \tag{1}$$

where $\mathcal{S}_v(G) \in \mathcal{S}_{\mathcal{V}}(\mathcal{G})$ denotes the set of all subgraphs of G that contain v. $\mathcal{V}'$ corresponds to the set of nodes over all graphs for which the equation is verified, and in this case, the explanation $\Psi(v, G)$ is called faithful.

Notice that a complete local factual explainer always exists, as $\Psi = \mathrm{Id}$ (identity function) verifies Equation 1 for every node. However, such explainers are rarely useful in practice because they do not produce concise, interpretable subgraphs. Therefore, the explanation subgraph size is an important metric, as formally defined in Section 5.1.

Although Definition 2 necessitates faithfulness of the explanation graphs and the existing methods that build explanations for GNNs mostly follow this definition, there are two major problems:

- It does not necessarily generate concise explanations. Generally, a good factual explanation subgraph is sparse (Kakkad et al., 2023; Yuan et al., 2023), containing only the most important graph elements: nodes, edges, and node features.
- More importantly, this problem definition does not account for human-level interpretability. For example, retrieving an explanation subgraph is common (Ying et al., 2019a; Luo et al., 2020), but it lacks sufficient information for understanding, which limits its usefulness in practice.

These criteria align with the goals of Explainable AI (XAI). Consequently, we aim to develop a method that enhances interpretability while maintaining conciseness.

4 Our Proposed Method: GSPELL

To explain a trained GNN's predictions, our method projects node embeddings from the GNN's latent space into the LLM's embedding space. By injecting these embeddings as soft prompts interleaved with natural language, we enable the LLM to generate context-aware explanations grounded in the model's internal representations. Our approach consists of three parts: training the projector, constructing the hybrid prompt, and generating the explanation subgraph (Figure 1).

4.1 Projection from the GNN

To help the LLM interpret the GNN, we developed a framework that aligns the LLM with the GNN's view of the computation tree. Specifically, we use a projector to map GNN node embeddings directly into the LLM's token space.

Given a GNN Φ trained on $\mathcal{G}$ for node classification on the node set $\mathcal{V}$, for a given graph $G = (V, E, T) \in \mathcal{G}$ and node $v \in V \subseteq \mathcal{V}$, we define $f_\Phi(v)$ to be the GNN embedding of node v in graph G, i.e the feature vector of v before the last prediction layer. The projector aligns GNN embeddings with natural language context, effectively bridging two distinct embedding spaces.

Generally speaking, this approach has similarities with other multi-modal alignment frameworks.

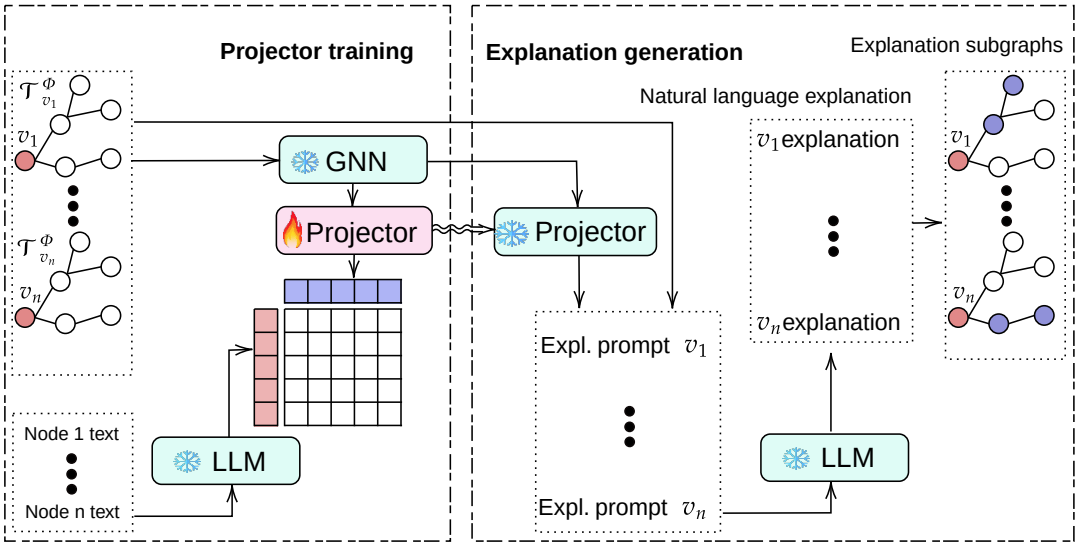

Figure 1: Illustration of GSPELL's framework. First, the projector is trained to align GNN node embeddings with the LLM's embedding space. Next, hybrid prompts are constructed by interleaving projected embeddings (as soft prompts) with natural language tokens. These prompts are then fed to the LLM to produce natural language explanations, which are converted into explanation subgraphs.

For instance, in (Radford et al., 2021), the authors use contrastive training to align image and text embeddings in a shared space, while in (Tsimpoukelli et al., 2021), images are projected directly into a frozen language model's embedding space to enable coherent language understanding. Similarly, our designed projector learns to map GNN embeddings into soft prompt tokens that serve as input to an LLM, effectively bridging the GNN's latent space and the LLM's token-level semantics.

Formally, we define a projector as a function $\Pi : \mathbb{R}^m \to \mathbb{R}^{k \times h}$, where m is the dimension of the GNN embedding $f_\Phi(v) \in \mathbb{R}^m$, k is the number of soft-prompt tokens, and h is the hidden dimension of the LLM's token embeddings. We write $\Pi(f_\Phi(v)) = Z_v \in \mathbb{R}^{k \times h}$. Π optimizes two losses:

1. **Context alignment loss:** This encourages the average soft token representation to align with the LLM embedding of the natural language text associated with the node:

$$\mathcal{L}_{\text{context}} = -\mathbb{E}_{v \in \mathcal{V}} \left[\cos\left(\bar{Z}_v,\ LLM(v) \right) \right]$$

where $\bar{Z}_v = \frac{1}{k} \sum_{i=1}^{k} Z_v^{(i)}$ is the mean-pooled projector output, $\cos(.,.)$ denotes cosine similarity, and $LLM(v) \in \mathbb{R}^h$ is the normalized embedding of the text associated with node v obtained from the LLM embedding space.

2. **Contrastive loss with GNN embeddings:** This encourages soft prompt representations to preserve the similarity structure of the GNN embeddings as follows:

$$\mathcal{L}_{\text{contrast}} = -\frac{1}{|\mathcal{V}|} \sum_{v \in \mathcal{V}} \sum_{u \in \mathcal{V}} p_{vu}^{\Phi} \log p_{vu}^{\Pi}$$

where

$$p_{vu}^{\Phi} = \frac{\exp\left(\cos\left(f_\Phi(v), f_\Phi(u)\right)/\tau\right)}{\sum_{w \in \mathcal{V}} \exp\left(\cos\left(f_\Phi(v), f_\Phi(w)\right)/\tau\right)}$$

$$p_{vu}^{\Pi} = \frac{\exp\left(\cos\left(\bar{Z}_v, \bar{Z}_u\right)\right)}{\sum_{w \in \mathcal{V}} \exp\left(\cos\left(\bar{Z}_v, \bar{Z}_w\right)\right)}$$

and $\tau > 0$ is a temperature hyperparameter. The overall training objective becomes: $\mathcal{L} = \beta\, \mathcal{L}_{\text{context}} + (1 - \beta)\mathcal{L}_{\text{contrast}}$ where $1 \geq \beta \geq 0$.

4.2 Prompt Construction

Given a node u, the projector outputs a matrix $\Pi(f_\Phi(u)) = Z_u \in \mathbb{R}^{k \times h}$, which we refer to as a soft prompt embedding. This matrix is directly injected into the LLM's embedding space and interleaved with natural language token embeddings.

For a node v that we wish to explain, we consider its computation tree for GNN Φ, denoted by $\mathcal{T}_v^\phi$. $\mathcal{T}_v^\phi$ is a tree with root node v and depth L (the number of layers of Φ) composed of all paths of length L starting at v, concatenated to the root v. We construct the LLM input as a sequence of embeddings consisting of: (i) Standard token embeddings for the system prompt and user instruction; (ii) The target node's k-token embedding Z_v—generated after enclosing the node's text descriptor by text markers; (iii) An enumeration of the nodes in $\mathcal{T}_v^\phi$, where each node is enclosed by text markers and its

own soft-prompt embeddings; and (iv) Final query instructions. By treating the GNN embeddings as native tokens, the LLM reasons across both GNN representations and textual features (see Figure 2 and Appendix D.2).

4.3 Explanation Generation

First, the LLM predicts whether each node in the computation tree supports or does not support the target node's classification. This partitions the nodes of the computation tree into three categories: (i) Supporting nodes, (ii) Opposing nodes, and (iii) Neutral nodes, which do not appear in either category. Formally, the LLM generates a natural language explanation E_v based on $\mathcal{T}_v^\Phi$ and the classification of v. Given the parsing of E_v we apply a label attribution function $\chi : \mathcal{T}_v^\phi \to \{-1, 0, 1\}$ for each node $u \in \mathcal{T}_v^\phi$ such that:

- $\chi(u) = 1$ if u is cited in E_v as supporting the classification of v,
- $\chi(u) = -1$ if u is cited as not supporting the classification of v,
- $\chi(u) = 0$ if u is not mentioned.

We consider the induced partition of $\mathcal{T}_v^\phi$:

$$S_v^+ = \{u \in \mathcal{T}_v^\phi \mid \chi(u) = 1\}$$
$$S_v^- = \{u \in \mathcal{T}_v^\phi \mid \chi(u) = -1\}$$
$$S_v^0 = \{u \in \mathcal{T}_v^\phi \mid \chi(u) = 0\}$$

The final explanation subgraph is $S_v = S_v^+$.

Mitigating Hallucination. LLM-generated explanations may hallucinate, especially when multiple node embeddings are projected within a single prompt. We address this issue through two complementary mechanisms within our framework:

- **Prompt templating with constrained node sets:** To reduce hallucinations, we partition $\mathcal{T}_v^\phi$ into fixed-size batches and sequentially prompt them to the LLM, instructing it to generate explanations that reference only the nodes in each batch.
- **Post hoc filtering:** After the LLM generates the explanations, any hallucinated nodes—the nodes absent in $\mathcal{T}_v^\phi$—are filtered out in a post-processing step. This filtering ensures that the final explanation remains relevant and grounded in the actual GNN model's predictions, as detailed in Section 4.3.

These strategies work together to minimize hallucination and enhance the accuracy and reliability of the generated explanations.

4.4 Faithfulness to GNN Predictions

Beyond constraining hallucinations, we also aim to ensure that the generated explanations remain faithful to the GNN's predictions. Accounting for context aligns the GNN embeddings with those of the LLM; however, this alignment alone does not guarantee that the GNN's decisions are faithful to the LLM's. We address faithfulness in two ways.

First, we explicitly include the GNN's predicted label for the target node in the input prompt (Sec. 4.2), so that the LLM conditions its reasoning on the model's actual decision. This provides the LLM with a clear natural-language interpretation of the prediction target.

Second, we introduce GSPELL$^+$, which directly optimizes soft prompts to ensure faithfulness to the GNN's predictions. Assume the GNN is a predictive model $\Phi = h \circ g$ where $g : G \to \mathcal{E}$ is a message-passing encoder and $h : \mathcal{E} \to Y$ is a decoder s.t. $\mathcal{E} \in \mathbb{R}^m$ and $Y \in \mathbb{R}^C$ are the latent and output spaces, and C is the number of classes. If Φ is well optimized, the conditional entropy $H(Y \mid \mathcal{E})$ is low due to the pretraining stage, which is equivalent to increasing the mutual information $I(Y; \mathcal{E})$. Therefore, a good projector Π has to not only align the GNN embeddings with the LLM embeddings, but also have the projections share high mutual information with the output of the GNN.

Along with the previous alignment objectives $\mathcal{L}_{\text{context}}$ and $\mathcal{L}_{\text{contrast}}$, we further consider increasing the mutual information $I(\hat{y}_v; Z_v)$, where $\hat{y}_v = h(g(v)) \in \mathbb{R}^C$ by minimizing the InfoNCE loss (van den Oord et al., 2019). Since Z_v and $\hat{y}_v$ do not live in the same space, we use another projector $\Pi' : \mathbb{R}^{k \times h} \to \mathbb{R}^C$ to back-propagate through the original projector $\Pi : \mathbb{R}^m \to \mathbb{R}^{k \times h}$ the faithfulness to the prediction. Therefore, writing $Z'_v = \Pi'(Z_v)$, the final objective for GSPELL$^+$ becomes:

$$\mathcal{L} = \beta_1 \, \mathcal{L}_{\text{context}} + \beta_2 \mathcal{L}_{\text{contrast}} + \beta_3 \mathcal{L}_{\text{InfoNCE}} \text{ where}$$
$$\mathcal{L}_{\text{InfoNCE}} =$$
$$-\mathbb{E}\left[\log \frac{\exp\left(\frac{\cos(Z'_v, \hat{y}_v)}{\tau}\right)}{\exp\left(\frac{\cos(Z'_v, \hat{y}_v)}{\tau}\right) + \sum_{u \neq v} \exp\left(\frac{\cos(Z'_v, \hat{y}_u)}{\tau}\right)}\right]$$

5 Experiments

We demonstrate the effectiveness of GSPELL in various settings. Our code is publicly available at: https://github.com/pbaghershahi/GSPELL.git.

	AMAZON		CORA		LIAR		WIKICS	
	Fidelity (%)	Size	Fidelity (%)	Size	Fidelity (%)	Size	Fidelity (%)	Size
GNNEXPLAINER	94.5 ± 1.5	5.71 ± 0.01	96.5 ± 0.6	17.91 ± 0.01	100.0 ± 0.00	474.01 ± 0.00	99.0 ± 0.0	1930.89 ± 0.0
PGEXPLAINER	73.8 ± 5.6	4.52 ± 0.13	82.0 ± 1.0	8.96 ± 0.40	100.0 ± 0.0	*1.0	90.6 ± 0.0	295.89 ± 0.0
TAGEXPLAINER	65.1 ± 4.4	1.03 ± 0.00	78.5 ± 0.9	1.06 ± 0.00	100.0 ± 0.0	1.31 ± 0.00	76.4 ± 1.1	1.51 ± 0.00
NODE	67.8 ± 2.7	1.00 ± 0.00	77.5 ± 2.6	1.00 ± 0.00	100.0 ± 0.0	1.00 ± 0.00	72.3 ± 0.0	1.00 ± 0.0
RANDOM	91.0 ± 1.1	7.28 ± 0.02	93.7 ± 0.5	18.67 ± 0.01	100.0 ± 0.0	469.36 ± 0.03	93.3 ± 0.0	994.76 ± 0.0
LLM	86.0 ± 1.2	4.90 ± 0.23	94.5 ± 1.0	3.27 ± 0.31	100.0 ± 0.0	2.22 ± 0.04	91.4 ± 1.0	21.19 ± 0.30
GSPELL$^+$	92.0 ± 1.8	4.24 ± 0.50	85.8 ± 1.8	2.10 ± 0.11	100.0 ± 0.0	1.10 ± 0.08	86.3 ± 1.7	12.44 ± 0.27
GSPELL	91.1 ± 2.3	4.25 ± 0.54	86.5 ± 0.0	2.07 ± 0.07	100.0 ± 0.0	1.11 ± 0.08	86.5 ± 0.0	12.80 ± 0.0

Table 1: Performance comparison with baseline models on the node classification task. Higher fidelity and lower size are better. *The explainer returns all-zero masks on LIAR, we consider its output to be the central node only.

5.1 Experimental setup

Evaluation Metrics. Having an explainer Ψ of GNN Φ, a dataset $\mathcal{G}$ of $\mathcal{V}$ nodes, and denoting explanation subgraph by $S_v = \Psi(v, G)$, we use the following metrics in the experiments:

1) Fidelity: Fidelity captures how well an explainable model reproduces the GNN model's logic in its predictions. Mathematically, we have:

$$Fid = \frac{1}{|\mathcal{V}|} \sum_{G \in \mathcal{G}, v \in V} \mathbb{I}[\Phi(v, S_v) = \Phi(v, G)]$$

where $\mathbb{I}[\cdot]$ is the indicator function.

2) Size: Smaller explanation sizes are easier to interpret. The explanation size measures the compactness of the explanation subgraph, i.e., the number of nodes in the explanation, as:

$$Size = \frac{1}{|\mathcal{V}|} \sum_{G \in \mathcal{G}, v \in \mathcal{V}} |S_v|$$

For a more comprehensive evaluation, we adopt the perturbation-based framework of recent natural language explanation literature (Atanasova et al., 2023; Siegel et al., 2024).

3) Comprehensiveness: Comprehensiveness measures the drop in prediction accuracy after removing the explanation subgraph. Let $G \setminus S_v$ denote the graph obtained by removing the nodes in S_v. Then, we define comprehensiveness as:

$$Comp. = \frac{1}{|\mathcal{V}|} \sum_{G \in \mathcal{G}, v \in V} \mathbb{I}[\Phi(x_v, G) = y_v] - \mathbb{I}[\Phi(x_v, G \setminus S_v) = y_v],$$

4) Alignment: Alignment evaluates whether the nodes identified as opposing the prediction, the set S_v^-, are indeed unsupportive. We define alignment as the drop in prediction accuracy after removing these nodes:

$$Align. = \frac{1}{|\mathcal{V}|} \sum_{G \in \mathcal{G}, v \in V} \mathbb{I}[\Phi(v, G) = y_v] - \mathbb{I}\left[\Phi\left(v, G \setminus S_v^-\right) = y_v\right].$$

5) Random Baseline: The random baseline measures the drop in prediction accuracy after removing a random node set, the same size as the explanation subgraph. Let $S_v^{\text{rand}} \subseteq T_v^{\phi}$ be a uniformly sampled subset such that $|S_v^{\text{rand}}| = |S_v|$, we define:

$$Rand. = \frac{1}{|\mathcal{V}|} \sum_{G \in \mathcal{G}, v \in V} \mathbb{I}[\Phi(v, G) = y_v] - \mathbb{I}\left[\Phi\left(v, G \setminus S_v^{\text{rand}}\right) = y_v\right].$$

Overall, higher values are preferred for Fidelity and Comprehensiveness, but lower values are preferred for Size and Alignment. The Random Baseline serves as a reference, where we expect Comprehensiveness to be significantly higher than the Random Baseline.

Datasets. We use real-world TAG datasets to evaluate the performance of our method: CORA (Sen et al., 2008), WIKICS (Mernyei and Cangea, 2022), AMAZON-PRODUCT (Feng et al., 2024), and LIAR (Wang, 2017). Additional details of the datasets are provided in Appendix B.

Models. We consider a GCN (Kipf and Welling, 2017a) as the base GNN model for the node classification task, and the LLM model is Meta's Llama-3.1-8B-Instruct, which remains frozen in our experiments. Full experimental details are provided in Appendix D. We provide experiments with other LLM models in Appendix C.2 and for different GNN architectures in Appendix C.1.

Baselines. We consider the following *five* baselines: **(1) Node:** This baseline uses only the target node itself as the explanation subgraph. This represents the minimal possible context and serves as a lower bound on explanation size. **(2) Random**: We introduce a randomized baseline that selects a

Dimension	CORA			AMAZON-PRODUCT		
	$M_1\uparrow$	$M_2\uparrow$	Δ	$M_1\uparrow$	$M_2\uparrow$	Δ
Understandability	$2.9_{\pm 1.23}$	$3.2_{\pm 1.45}$	0.3	$3.33_{\pm 1.46}$	$3.25_{\pm 1.33}$	-0.08
Trustworthiness	$2.7_{\pm 1.10}$	$3.1_{\pm 1.06}$	0.4	$3.30_{\pm 1.56}$	$3.05_{\pm 1.41}$	-0.25
Insightfulness	$3.3_{\pm 0.74}$	$3.1_{\pm 0.95}$	-0.2	$2.95_{\pm 1.34}$	$3.25_{\pm 1.17}$	0.30
Satisfaction	$2.2_{\pm 1.16}$	$2.7_{\pm 0.92}$	0.5	$3.08_{\pm 1.46}$	$3.05_{\pm 1.36}$	-0.03
Confidence	$3.0_{\pm 1.49}$	$3.3_{\pm 1.49}$	0.3	$3.13_{\pm 1.47}$	$3.03_{\pm 1.33}$	-0.10
Convincingness	$3.3_{\pm 1.57}$	$3.7_{\pm 1.64}$	0.4	$3.08_{\pm 1.46}$	$2.93_{\pm 1.49}$	-0.15
Communicability	$3.0_{\pm 1.60}$	$3.1_{\pm 1.41}$	0.1	$2.88_{\pm 1.28}$	$2.93_{\pm 1.23}$	0.05
Usability	$3.3_{\pm 1.43}$	$3.4_{\pm 1.42}$	0.1	$2.75_{\pm 1.39}$	$2.80_{\pm 1.22}$	0.05

Table 2: Expert qualitative evaluation of the explanations from GSPELL and GNNExplainer on CORA and AMAZON-PRODUCT. M_1 denotes the mean scores for GNNExplainer and M_2 the mean scores for GSPELL.

	AMAZON					CORA					WIKICS				
	F (%) ↑	C (%) ↑	R (%)	A (%) ↓	S ↓	F (%) ↑	C (%) ↑	R (%)	A (%) ↓	S ↓	F (%) ↑	C (%) ↑	R (%)	A (%) ↓	S ↓
GNNEXPLAINER	94.5	14.7	6.9	-	5.71	96.5	15.8	7.8	-	17.91	99.0	13.1	13.1	-	1930.89
GSPELL	91.1	8.2	4.8	3.0	4.25	86.5	5.8	1.2	7.8	2.07	86.5	4.20	0.0	4.7	12.80
GSPELL$^+$	92.0	8.3	3.5	5.0	4.24	85.8	6.2	1.5	8.2	2.10	86.3	3.8	0.2	5.8	12.44

Table 3: Perturbation-based evaluation of faithfulness. GSPELL shows a superior trade-off between faithfulness and conciseness, achieved by isolating critical nodes for human-centered explanations in natural language.

subgraph around each target node as its explanation subgraph. Specifically, for each node, we sample a subgraph of half the size of its computation tree. The results are averaged over 5 samples. **(3-5) Graph Explainers**: We compare GSPELL with GNNExplainer (Ying et al., 2019a), PGExplainer (Luo et al., 2020) as subgraph-based explainers, which do not involve any LLM-based component. We also include the recent LLM-based graph explanation method, TAGExplainer (Pan et al., 2024).

5.2 Quantitative Results

Table 1 shows that the Node baseline is relatively strong, highlighting the critical role of textual attributes in GNN predictions. While GNNExplainer often reaches the highest fidelity, this metric alone can be misleading. Since GNNs aggregate neighborhood data, high fidelity can be trivially achieved by simply outputting the entire induced subgraph.

In homophilic graphs, where neighbors often share the same category, selecting a broad neighborhood effectively replicates the GNN's original context. This is evidenced by the Random baseline, which achieves fidelity scores comparable to GNNExplainer. In these settings, attaining high fidelity is trivial if the explanation subgraph is large.

Across all datasets, GSPELL achieves a superior fidelity-size trade-off: despite marginally lower fidelity than GNNExplainer and the LLM baseline, it produces significantly smaller subgraphs. Unlike Random or Node baselines, which either inflate size or restrict context, GSPELL leverages projected GNN embeddings to distill rich textual semantics into concise, interpretable explanations.

Comparing GSPELL$^+$ to GSPELL highlights the impact of the InfoNCE objective: while it further reduces explanation size, it occasionally lowers fidelity. The objective's effectiveness scales with the number of negative samples; for instance, it outperforms on AMAZON-PRODUCT compared to CORA because a higher class count provides a tighter mutual information bound.

Perturbation-based Evaluation. To further assess the faithfulness of the generated explanations, we adopt perturbation-based evaluation.

Table 3 demonstrates that GSPELL optimizes the trade-off between conciseness and faithfulness. While GNNExplainer shows high fidelity, it relies on brute-force coverage—often selecting nearly entire neighborhoods. For instance, on Cora, GNNExplainer averages 18 nodes per explanation, whereas GSPELL averages 2. This distinction is further highlighted on WikiCS: GNNExplainer produces excessively large, unreadable subgraphs, whereas GSPELL identifies high-impact nodes within significantly smaller, interpretable structures. Moreover, GSPELL achieves a comprehensiveness of 2–5$\times$ greater than the random drop baseline, outperforming GNNExplainer's 2–3$\times$ margin. This demonstrates that GSPELL maintains a superior signal-to-noise ratio, making human-readable narratives.

Unlike baselines that provide binary-mask explanations, GSPELL 's 3-way classification—Support, Oppose, and Neutral—uniquely enables the Align-

LLM Prompt

You are analyzing Amazon product reviews and their co-purchase neighborhoods to understand why it has been classified under a specific category. Your task is to determine which of the neighboring products does (not) support the classification of the target product.
Target Product ID: 160
Predicted Category: Sports & Outdoors

Product Embedding Representation:
«BEGIN TARGET KEYWORDS» PROJECTED EMBEDDING «END TARGET KEYWORDS»

Neighboring Products:
Each Product below is described by keywords.

Product 3: «BEGIN KEYWORDS» PROJECTED EMBEDDING «END KEYWORDS»
Product 4: «BEGIN KEYWORDS» PROJECTED EMBEDDING «END KEYWORDS»
Product 5: «BEGIN KEYWORDS» PROJECTED EMBEDDING «END KEYWORDS»
Product 10: «BEGIN KEYWORDS» PROJECTED EMBEDDING «END KEYWORDS»
Product 12: «BEGIN KEYWORDS» PROJECTED EMBEDDING «END KEYWORDS»

LLM Response

P3	Summary: This product seems to be a fitness accessory, possibly a yoga mat or a resistance band, given its mention of "workout" and "stretching".	YES
P4	Summary: This product appears to be a nutrition supplement, possibly a protein powder or a vitamin, given its mention of "muscle" and "energy".	NO
P5	Summary: This product seems to be a fitness tracker or a smartwatch, given its mention of "fitness" and "activity tracking".	YES
P10	Summary: This product seems to be a health and wellness product, possibly a meditation app or a mindfulness journal, given its mention of "relax" and "meditate".	NO
P12	Summary: This product appears to be a sports accessory, possibly a water bottle or a towel, given its mention of "hydrate" and "dry off".	YES

Instructions

For each neighboring product:
1. Summarize the similarity between supporting products in *one sentence*.
2. Clearly state whether this product supports the classification of the Target Product into category Sports & Outdoors.

Format:
```
Product <ID>:
Summary: <One sentence>.
Support: YES or NO — Does this product support
classification into Sports & Outdoors?
```
Base reasoning only on the keywords and proximity to the target product.

Figure 2: Left: prompt with category and embeddings highlighted. Right: full model response with summaries in the middle and YES/NO verdicts aligned on the right.

ment metric to differentiate between contradicting evidence and irrelevant noise to detect hallucinations. Across all datasets, GSPELL's Alignment scores are low, particularly on Amazon, confirming successful detection of irrelevant nodes.

5.3 Qualitative Evaluation

Expert Assessment. We asked experts to evaluate the explanations generated by GNNExplainer and GSPELL on the AMAZON-PRODUCT and CORA datasets over 10 node classification tasks. We follow the qualitative evaluation framework provided in (Cedro and Martens, 2025), and refer to Appendix E for a full description of the evaluation dimensions. The evaluation framework and setup are presented in Appendix F.

As Table 2 shows, GSPELL's explanations on CORA were more trustworthy and convincing, likely due to the LLM component's semantic understanding. However, these explanations were seen as less insightful, suggesting more limited exploration beyond the article's main subject. On AMAZON-PRODUCT, GSPELL provided more insightful explanations than GNNExplainer by capturing richer semantic links across product categories, but it was rated lower in trustworthiness and convincingness, indicating less consistency and reliability.

Overall, our method performs better on CORA, where its explanations were viewed as more trustworthy, convincing, and communicable. On the more challenging AMAZON-PRODUCT dataset, GNNExplainer retained an edge in clarity and user confidence, suggesting that dataset complexity and semantic richness shape the relative strengths of LLM-based explanations.

Real Example. Figure 2 illustrates a prompting example of GSPELL on the Amazon dataset to emphasize the narrative quality and interpretability of the generated explanation. In this instance, the target node corresponds to a product (ID 160), predicted to belong to the "Sports & Outdoors" category. GSPELL leverages latent GNN representations via soft prompts to benefit from the LLM's language-generation capabilities while accounting for the local neighborhood of the target node.

GSPELL generates node-level rationales from the target node's computation tree, providing for each neighbor a binary support judgment (YES/NO) indicating whether it supports the target's classification. Also, it provides a general summary that highlights common patterns across the target's context, helping to explain the model's prediction.

6 Ablation Study and Runtime Analysis

6.1 Effect of the projection tokens

Without accounting for GNN internals (embeddings), asking an LLM to generate explanations only based on the input and the predictions of the GNN is theoretically wrong. The main reason is that one can directly substitute the predictive GNN with any arbitrary function that makes predictions based on different reasoning (logic) than the GNN, while still making the same predictions. Therefore, we use the soft prompt generated by the projector for the faithfulness of the explanations.

		$k=1$		$k=5$		$k=10$		$k=20$		$k=50$	
		Fidelity (%)	Size	Fidelity (%)	Size	Fidelity (%)	Size	Fidelity (%)	Size	Fidelity (%)	Size
Amazon	GSPELL	92.0	4.22	91.1	4.25	88.0	1.71	85.0	1.17	82.5	1.00
	GSPELL$^+$	90.5	4.45	92.0	4.24	86.0	1.45	85.0	1.13	81.0	1.01
Liar	GSPELL	100.0	1.13	100.0	1.11	100.0	1.11	100.0	1.00	OOM	OOM
	GSPELL$^+$	100.0	1.14	100.0	1.10	98.4	1.10	100.0	1.00	OOM	OOM
Cora	GSPELL	86.2	1.98	86.5	2.07	87.8	2.04	85.4	1.81	84.4	1.33
	GSPELL$^+$	85.2	1.87	85.8	2.10	88.0	2.17	87.4	2.17	84.8	1.30

Table 4: Effect of the number of tokens (k) on the performance of GSPELL and GSPELL$^+$.

Soft prompts channel information from the GNN to the LLM, with the token count, k, being a critical factor here. Insufficient tokens constrain the information channel, causing the LLM to ignore GNN-derived context in favor of textual priors, thereby increasing hallucination rates. Conversely, a large number of tokens can disrupt coherence during generation. If the soft prompt distribution aligns poorly with the LLM's embedding space—often due to a suboptimal projection function–increasing k injects out-of-distribution noise into the model. This results in degraded generation quality and increases computational overhead.

We perform an ablation study over different token counts (k), shown in Table 4. We observe that k acts as a sparsity regulator. On the complex Amazon dataset, low k yields high fidelity with larger explanations, while high k significantly reduces size at the cost of fidelity. This supports our hypothesis that excessive soft tokens inject noise into the LLM. We find an optimal balance at $k = 5$, achieving high fidelity with reduced explanation size. In contrast, Liar, a more heterogeneous dataset, benefits from a higher k as it likely requires more tokens to encode complex relationships than standard homogeneous graphs. Conversely, Cora exhibits high robustness in both fidelity and sparsity across all k values, likely due to its higher homophily.

6.2 Runtime

We provide the details of running times in Table 5, comparing our method with all baselines. Although GSPELL is more time-consuming than other baselines, we note that this is justified by the fact that GSPELL, unlike the fast baselines, e.g., PGExplainer and TAGExplainer, generates descriptive, informative explanations in natural language that are understandable by humans.

7 Conclusions

In this work, we propose GSPELL, a lightweight, post-hoc explanation framework that uses LLMs to generate faithful and human-interpretable explanations for GNN predictions on text-attributed graphs. By projecting GNN embeddings into the LLM space, GSPELL enables context-aware, natural language rationales without relying on an external GNN explainer. Our approach achieves a superior trade-off between conciseness and fidelity, as well as greater semantic clarity, compared to prior methods. GSPELL demonstrates that LLMs can serve as effective interpreters for GNNs and offers a scalable and intuitive path toward explainability in graph-based applications.

	AMAZON	CORA	LIAR	WIKICS
GNNEXPLAINER	1.45	1.69	1.79	3.76
PGEXPLAINER	0.02	0.03	0.79	0.80
TAGEXPLAINER	0.05	0.13	0.52	1.57
LLM	33.74	18.21	12.80	61.67
GSPELL (Ours)	39.65	30.77	42.41	76.76

Table 5: Inference runtime per node for explanation generation measured in seconds.

Future Work. Although GSPELL$^+$ incorporates an InfoNCE loss to enhance faithfulness, it does not consistently outperform GSPELL. Moreover, the other projector objectives focus on aligning GNN and LLM embedding spaces rather than directly optimizing explanation quality. Designing more principled, explanation-centered objectives is an important direction for future work.

Additionally, as the computation tree grows, the LLM struggles to capture the full graph context. While batch-wise prompting mitigates this issue, it limits global reasoning. Developing prompt compression techniques that preserve structural information could improve the quality of the generated explanations.

Acknowledgment

The authors acknowledge the National Artificial Intelligence Research Resource (NAIRR) Pilot and the Texas Advanced Computing Center (TACC) Vista for contributing to this research.

8 Ethical considerations

In this work, we have built methods for generating explanations for graph neural network predictions. We do not foresee any ethical issues from our study.

Potential Risks. While GSPELL aims to improve the interpretability of GNN predictions, the generated explanations rely on LLM outputs and may not always faithfully reflect the underlying model's true decision process. This could lead to over-trust in the explanations. Also, the use of LLMs introduces the risk of hallucinated or misleading reasoning, despite our mitigation strategies. We encourage careful human oversight when deploying such explanation systems in practice.

9 Limitations

We present some limitations of our work:

- Our method combines projected GNN embeddings with text in hybrid prompts. When the computation tree is large, processing the full context can lead to hallucinations, as the LLM struggles to distinguish among multiple soft prompt representations, while batch-wise prompting limits its ability to capture a global view.
- The explanations generated by GSPELL are produced by an LLM conditioned on projected embeddings rather than explicitly tracing the GNN's computation. Consequently, they may not reflect the true causal factors underlying the model's prediction, so we do not provide formal guarantees of causal faithfulness.
- Although the projector aligns GNN embeddings with the LLM embedding space, its training objectives are not directly tailored for explanation. While we explore an InfoNCE-based objective to improve faithfulness, it does not consistently yield gains, suggesting that more effective explanation-specific objectives are needed.

References

Carlo Abrate and Francesco Bonchi. 2021. Counterfactual graphs for explainable classification of brain networks. In *KDD*, page 2495–2504.

Burouj Armgaan, Manthan Dalmia, Sourav Medya, and Sayan Ranu. 2024. Graphtrail: Translating gnn predictions into human-interpretable logical rules. *Advances in Neural Information Processing Systems*, 37:123443–123470.

Pepa Atanasova, Oana-Maria Camburu, Christina Lioma, Thomas Lukasiewicz, Jakob Grue Simonsen, and Isabelle Augenstein. 2023. Faithfulness tests for natural language explanations. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 283–294, Toronto, Canada. Association for Computational Linguistics.

Steve Azzolin, Antonio Longa, Pietro Barbiero, Pietro Lio, and Andrea Passerini. 2023. Global explainability of GNNs via logic combination of learned concepts. In *The Eleventh International Conference on Learning Representations*.

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*.

Mateusz Cedro and David Martens. 2025. Graphxain: Narratives to explain graph neural networks. *Preprint*, arXiv:2411.02540.

Runjin Chen, Tong Zhao, Ajay Kumar Jaiswal, Neil Shah, and Zhangyang Wang. 2024a. LLaGA: Large language and graph assistant. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pages 7809–7823.

Zhikai Chen, Haitao Mao, Hang Li, Wei Jin, Hongzhi Wen, Xiaochi Wei, Shuaiqiang Wang, Dawei Yin, Wenqi Fan, Hui Liu, and Jiliang Tang. 2024b. Exploring the potential of large language models (llms)in learning on graphs. *SIGKDD Explor. Newsl.*, 25(2):42–61.

Ziheng Chen, Fabrizio Silvestri, Jia Wang, Yongfeng Zhang, Zhenhua Huang, Hongshik Ahn, and Gabriele Tolomei. 2022. Grease: Generate factual and counterfactual explanations for gnn-based recommendations. *arXiv preprint arXiv:2208.04222*.

Kewei Cheng, Nesreen K. Ahmed, Theodore L. Willke, and Yizhou Sun. 2024. Structure guided prompt: Instructing large language model in multi-step reasoning by exploring graph structure of the text. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 9407–9430.

Chirag Chhablani, Sarthak Jain, Akshay Channesh, Ian A Kash, and Sourav Medya. 2024. Game-theoretic counterfactual explanation for graph neural networks. In *Proceedings of the ACM on Web Conference 2024*, pages 503–514.

Yi Fang, Dongzhe Fan, Daochen Zha, and Qiaoyu Tan. 2024. Gaugllm: Improving graph contrastive learning for text-attributed graphs with large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, page 747–758.

Bahare Fatemi, Jonathan Halcrow, and Bryan Perozzi. 2024. Talk like a graph: Encoding graphs for large language models. In *The Twelfth International Conference on Learning Representations*.

Jiarui Feng, Hao Liu, Lecheng Kong, Mingfang Zhu, Yixin Chen, and Muhan Zhang. 2024. Taglas: An atlas of text-attributed graph datasets in the era of large graph and language models. *Preprint*, arXiv:2406.14683.

Gregoire Fournier and Sourav Medya. 2025. Comrecgc: Global graph counterfactual explainer through common recourse. *arXiv preprint arXiv:2505.07081*.

Zhimeng Guo, Teng Xiao, Charu Aggarwal, Hui Liu, and Suhang Wang. 2023. Counterfactual learning on graphs: A survey. *arXiv preprint arXiv:2304.01391*.

Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, volume 30.

Xiaoxin He, Xavier Bresson, Thomas Laurent, Adam Perold, Yann LeCun, and Bryan Hooi. 2024a. Harnessing explanations: LLM-to-LM interpreter for enhanced text-attributed graph representation learning. In *The Twelfth International Conference on Learning Representations*.

Yinhan He, Zaiyi Zheng, Patrick Soga, Yaochen Zhu, Yushun Dong, and Jundong Li. 2024b. Explaining graph neural networks with large language models: A counterfactual perspective on molecule graphs. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 7079–7096.

Zhengyu Hu, Yichuan Li, Zhengyu Chen, Jingang Wang, Han Liu, Kyumin Lee, and Kaize Ding. 2024. Let's ask GNN: Empowering large language model for graph in-context learning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 1396–1409.

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*.

Zexi Huang, Mert Kosan, Sourav Medya, Sayan Ranu, and Ambuj Singh. 2023. Global counterfactual explainer for graph neural networks. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, pages 141–149.

Jaykumar Kakkad, Jaspal Jannu, Kartik Sharma, Charu Aggarwal, and Sourav Medya. 2023. A survey on explainability of graph neural networks. *IEEE Data Eng. Bull.*, 47(2):35–63.

Thomas N. Kipf and Max Welling. 2017a. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*.

Thomas N. Kipf and Max Welling. 2017b. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*.

Yuhan Li, Peisong Wang, Zhixun Li, Jeffrey Xu Yu, and Jia Li. 2024. Zerog: Investigating cross-dataset zero-shot transferability in graphs. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, page 1725–1735.

Yuhan Li, Xinni Zhang, Linhao Luo, Heng Chang, Yuxiang Ren, Irwin King, and Jia Li. 2025. G-refer: Graph retrieval-augmented large language model for explainable recommendation. In *Proceedings of the ACM on Web Conference 2025, WWW 2025, Sydney, NSW, Australia, 28 April 2025- 2 May 2025*, pages 240–251. ACM.

Wanyu Lin, Hao Lan, and Baochun Li. 2021. Generative causal explanations for graph neural networks. In *International Conference on Machine Learning*, pages 6666–6679. PMLR.

Hao Liu, Jiarui Feng, Lecheng Kong, Ningyue Liang, Dacheng Tao, Yixin Chen, and Muhan Zhang. 2024a. One for all: Towards training one graph model for all classification tasks. In *The Twelfth International Conference on Learning Representations*.

Yunchao Liu, Yu Wang, Oanh T Vu, Rocco Moretti, Bobby Bodenheimer, Jens Meiler, and Tyler Derr. 2022. Interpretable chirality-aware graph neural network for quantitative structure activity relationship modeling in drug discovery. *bioRxiv*, pages 2022–08.

Zheyuan Liu, Xiaoxin He, Yijun Tian, and Nitesh V. Chawla. 2024b. Can we soft prompt llms for graph learning tasks? In *Companion Proceedings of the ACM Web Conference 2024*, WWW '24, page 481–484. ACM.

Shengyao Lu, Keith G Mills, Jiao He, Bang Liu, and Di Niu. 2024. GOAt: Explaining graph neural networks via graph output attribution. In *The Twelfth International Conference on Learning Representations*.

Ana Lucic, Maartje A Ter Hoeve, Gabriele Tolomei, Maarten De Rijke, and Fabrizio Silvestri. 2022a. Cf-gnnexplainer: Counterfactual explanations for graph neural networks. In *International Conference on Artificial Intelligence and Statistics*, pages 4499–4511. PMLR.

Ana Lucic, Maartje A Ter Hoeve, Gabriele Tolomei, Maarten De Rijke, and Fabrizio Silvestri. 2022b. Cf-gnnexplainer: Counterfactual explanations for graph neural networks. In *AISTATS*, pages 4499–4511.

Dongsheng Luo, Wei Cheng, Dongkuan Xu, Wenchao Yu, Bo Zong, Haifeng Chen, and Xiang Zhang. 2020. Parameterized explainer for graph neural network. *Advances in neural information processing systems*, 33:19620–19631.

Qiyao Ma, Xubin Ren, and Chao Huang. 2024. XRec: Large language models for explainable recommendation. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 391–402. Association for Computational Linguistics.

Péter Mernyei and Cătălina Cangea. 2022. Wiki-cs: A wikipedia-based benchmark for graph neural networks. *Preprint*, arXiv:2007.02901.

Bo Pan, Zhen Xiong, Guanchen Wu, Zheng Zhang, Yifei Zhang, and Liang Zhao. 2024. Tagexplainer: Narrating graph explanations for text-attributed graph learning models. *Preprint*, arXiv:2410.15268.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. *Preprint*, arXiv:2103.00020.

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. *Proceedings of the VLDB Endowment*, (3):427–436.

Xubin Ren, Jiabin Tang, Dawei Yin, Nitesh Chawla, and Chao Huang. 2024. A survey of large language models for graphs. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, page 6616–6626.

Michael Sejr Schlichtkrull, Nicola De Cao, and Ivan Titov. 2021. Interpreting graph neural networks for nlp with differentiable edge masking. *International Conference on Learning Representations*.

Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Gallagher, and Tina Eliassi-Rad. 2008. Collective classification in network data. *AI Mag.*, 29(3):93–106.

Caihua Shan, Yifei Shen, Yao Zhang, Xiang Li, and Dongsheng Li. 2021. Reinforcement learning enhanced explainer for graph neural networks. In *NeurIPS 2021*.

Ben Shneiderman. 2020. Bridging the gap between ethics and practice: guidelines for reliable, safe, and trustworthy human-centered ai systems. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 10(4):1–31.

Noah Siegel, Oana-Maria Camburu, Nicolas Heess, and Maria Perez-Ortiz. 2024. The probabilities also matter: A more faithful metric for faithfulness of free-text explanations in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 530–546, Bangkok, Thailand. Association for Computational Linguistics.

Mengying Sun, Sendong Zhao, Coryandar Gilvary, Olivier Elemento, Jiayu Zhou, and Fei Wang. 2020. Graph convolutional networks for computational drug development and discovery. *Briefings in bioinformatics*, 21(3):919–935.

Juntao Tan, Shijie Geng, Zuohui Fu, Yingqiang Ge, Shuyuan Xu, Yunqi Li, and Yongfeng Zhang. 2022a. Learning and evaluating graph neural network explanations based on counterfactual and factual reasoning. In *Proceedings of the ACM Web Conference 2022*, pages 1018–1027.

Juntao Tan, Shijie Geng, Zuohui Fu, Yingqiang Ge, Shuyuan Xu, Yunqi Li, and Yongfeng Zhang. 2022b. Learning and evaluating graph neural network explanations based on counterfactual and factual reasoning. In *Proceedings of the ACM Web Conference 2022*, WWW '22, page 1018–1027.

Jiabin Tang, Yuhao Yang, Wei Wei, Lei Shi, Lixin Su, Suqi Cheng, Dawei Yin, and Chao Huang. 2024. Graphgpt: Graph instruction tuning for large language models. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 491–500.

Yijun Tian, Huan Song, Zichen Wang, Haozhu Wang, Ziqing Hu, Fang Wang, Nitesh V. Chawla, and Panpan Xu. 2023. Graph neural prompting with large language models. *Preprint*, arXiv:2309.15427.

Maria Tsimpoukelli, Jacob Menick, Serkan Cabi, S. M. Ali Eslami, Oriol Vinyals, and Felix Hill. 2021. Multimodal few-shot learning with frozen language models. *Preprint*, arXiv:2106.13884.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2019. Representation learning with contrastive predictive coding. *Preprint*, arXiv:1807.03748.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

Samidha Verma, Burouj Armgaan, Sourav Medya, and Sayan Ranu. 2024. InduCE: Inductive counterfactual explanations for graph neural networks. *Transactions on Machine Learning Research*.

Duo Wang, Yuan Zuo, Fengzhi Li, and Junjie Wu. 2024. Llms as zero-shot graph learners: Alignment of gnn representations with llm token embeddings. *Preprint*, arXiv:2408.14512.

Heng Wang, Shangbin Feng, Tianxing He, Zhaoxuan Tan, Xiaochuang Han, and Yulia Tsvetkov. 2023. Can language models solve graph problems in natural language? In *Thirty-seventh Conference on Neural Information Processing Systems*.

William Yang Wang. 2017. "liar, liar pants on fire": A new benchmark dataset for fake news detection. *Preprint*, arXiv:1705.00648.

Xiaoqi Wang and Han Wei Shen. 2023. GNNInterpreter: A probabilistic generative model-level explanation for graph neural networks. In *The Eleventh International Conference on Learning Representations*.

Geemi P Wellawatte, Aditi Seshadri, and Andrew D White. 2022. Model agnostic generation of counterfactual explanations for molecules. *Chemical science*, 13(13):3697–3705.

Jiacheng Xiong, Zhaoping Xiong, Kaixian Chen, Hualiang Jiang, and Mingyue Zheng. 2021. Graph neural networks for automated de novo drug design. *Drug Discovery Today*, 26(6):1382–1393.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *7th International Conference on Learning Representations, ICLR 2019*.

Han Xuanyuan, Pietro Barbiero, Dobrik Georgiev, Lucie Charlotte Magister, and Pietro Lió. 2023. Global concept-based interpretability for graph neural networks via neuron analysis. In *AAAI*.

Haotong Yang, Xiyuan Wang, Qian Tao, Shuxian Hu, Zhouchen Lin, and Muhan Zhang. 2025. GL-fusion: Rethinking the combination of graph neural network and large language model.

Rex Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. 2019a. *GNNExplainer: generating explanations for graph neural networks*. Curran Associates Inc., Red Hook, NY, USA.

Zhitao Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. 2019b. Gnnexplainer: Generating explanations for graph neural networks. *Advances in neural information processing systems*, 32.

Hao Yuan, Jiliang Tang, Xia Hu, and Shuiwang Ji. 2020. Xgnn: Towards model-level explanations of graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 430–438.

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*.

Hao Yuan, Haiyang Yu, Shurui Gui, and Shuiwang Ji. 2023. Explainability in graph neural networks: A taxonomic survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5):5782–5799.

Hao Yuan, Haiyang Yu, Jie Wang, Kang Li, and Shuiwang Ji. 2021. On explainability of graph neural networks via subgraph explorations. In *ICML*, pages 12241–12252. PMLR.

Jiaxing Zhang, Jiayi Liu, Dongsheng Luo, Jennifer Neville, and Hua Wei. 2024. Llmexplainer: Large language model based bayesian inference for graph explanation generation. *Preprint*, arXiv:2407.15351.

Marinka Zitnik, Monica Agrawal, and Jure Leskovec. 2018. Modeling polypharmacy side effects with graph convolutional networks. *Bioinformatics*, 34(13):i457–i466.

Marinka Zitnik, Francis Nguyen, Bo Wang, Jure Leskovec, Anna Goldenberg, and Michael M Hoffman. 2019. Machine learning for integrating data in biology and medicine: Principles, practice, and opportunities. *Information Fusion*, 50:71–91.

A Additional Related Work

Here we provide additional related works that are on the integrated methods of LLMs and GNNs. The integration of LLMs with GNNs aims to leverage the advanced reasoning abilities of LLMs and the extensive background knowledge they acquire, alongside the GNNs' proven capacity to enhance the exploitation of graph structures via message-passing mechanisms (Kipf and Welling, 2017b; Hamilton et al., 2017). The approaches for integrating LLMs and GNNs primarily vary in the sequence of component application (Ren et al., 2024; Chen et al., 2024b).

LLMs as prefix and postfix. First, there are methods where LLMs are used in the prefix modules for task and feature representation (Liu et al., 2024a; Fang et al., 2024; He et al., 2024a; Yang et al., 2025). This architecture provides greater flexibility, as various tasks can be specified in natural language with text-based graph representations, and LLM modules can remain fixed. On the other hand, the second class of methods uses GNNs as the prefix (Wang et al., 2023; Chen et al., 2024a; Hu et al., 2024; Tang et al., 2024), which is better suited for capturing the graph topology and its intricate relationships and uses LLMs for prediction. This approach also has some flexibility, as the LLM can output embeddings, prediction scores, or textual sequences to address various downstream problems specifically for generation tasks (Wang et al., 2023; Chen et al., 2024a; Hu et al., 2024; Tang et al., 2024).

Independent LLMs and joint LLM-GNNs. Additionally, there is a body of work that employs LLMs independently to tackle graph-related tasks (Cheng et al., 2024; Fatemi et al., 2024; Li et al., 2024). Approaches following this paradigm benefit from LLMs' generalization capabilities at both the input and output levels and can potentially avoid training learnable parameters by fixing the LLM component and relying on input prompts. Finally, there is a group of works which employ GNNs and LLMs jointly through soft prompting, which necessitates alignments between their embeddings while enhancing LLMs reasoning by the GNN captured graph structural features (Liu et al., 2024b; Wang et al., 2024; Tian et al., 2023). However, these works do not address the GNN explanation problem.

Explainers for GNNs. Several algorithms have been proposed in the literature to enhance the interpretability of GNNs. The majority of existing explainers concentrate on instance-level explanations. Instance-level (or local) explainers operate on individual graphs, identifying influential components—such as subgraphs—that most strongly impact the model's prediction (Ying et al., 2019b; Luo et al., 2020; Shan et al., 2021; Huang et al., 2022; Yuan et al., 2022; Lucic et al., 2022b; Tan et al., 2022b; Lin et al., 2021; Bajaj et al., 2021; Abrate and Bonchi, 2021; Chhablani et al., 2024; Wellawatte et al., 2022; Verma et al., 2024; Lu et al., 2024). However, this localized perspective constrains their ability to uncover global patterns leveraged by GNNs across multiple graphs, as well as how these patterns integrate into a unified decision-making process. Research on global GNN explainers remains relatively limited (Yuan et al., 2020; Huang et al., 2023; Xuanyuan et al., 2023; Azzolin et al., 2023; Armgaan et al., 2024; Fournier and Medya, 2025). For instance, XGNN (Yuan et al., 2020) and GNNInterpreter (Wang and Shen, 2023) adopt generative modeling approaches, producing graphs that most strongly correspond to a given class label. For more details, please refer to this survey on GNN explainers (Kakkad et al., 2023).

Dataset	# Nodes	# Edges	# Categories	# Features
CORA	2,708	5,429	7	2000
WIKICS	11701	148555	10	300
AMAZON	1000	1397	47	2000
LIAR	13293	32443	7	387

Table 6: Statistics of the graph datasets.

B Datasets

B.1 Datasets Details

We describe in detail the datasets that we used in our evaluation below. The basic statistics about these datasets are provided in Table 6.

1. CORA: It is a citation network, in which nodes represent computer science research papers, and each edge between two nodes represents a research paper citing another. The nodes are classified into one of seven categories. Though a citation network is a directed network, the dataset is widely used as an undirected network in the message-passing-based graph machine learning, especially for the node classification task, in which the task is to predict each node's category based on its own text features and text features of its neighbors. Each node's feature is a 2000-dimensional bag-of-words representation of the

Dataset	Description & Relevance	Why It Fits Our Work
CORA	Citation network of CS papers with bag-of-words features.	Standard benchmark for node classification on text-rich graphs.
WIKICS	Wikipedia CS articles linked by hyperlinks, with GloVe embeddings.	Large, semantic-rich graph ideal for explanation evaluation.
LIAR	Fake news detection graph combining statements, speakers, topics.	Challenging, heterogeneous graph testing method adaptability.
AMAZON	Product co-purchase network with 47 categories.	Real-world, diverse e-commerce graph to test scalability.

Table 7: Dataset Descriptions and Relevance to Our Work

keywords in the paper that it corresponds to.

2. WIKICS: It is a text-attributed graph dataset, derived from the Wikipedia platform, widely used for node classification tasks. The nodes correspond to Wikipedia page descriptions of different computer science topics, and the edges represent hyperlinks from one article to another. Each node in the dataset belongs to one of 10 categories. The node classification task is to correctly predict a node's label, and edges are undirected. Each node's feature vector is a 300-dimensional GloVe embedding computed from the text associated with the node.
3. LIAR: It is a fake-news detection dataset that is often represented as a knowledge graph, with nodes corresponding to statements, speakers, and topics, and edges encoding typed relations such as *spoken_by* and *about*. To adapt LIAR into a homogeneous graph suitable for standard GNN pipelines, we merge the three node types—statements, speakers, and topics—into a single unified node set. Each node is embedded with a 384-dimensional BERT representation and augmented with a 3-dimensional one-hot vector indicating its type. Edges representing the original heterogeneous relations (*spoken_by* and *about*) are converted to undirected edges and unified into a single edge type in the homogeneous graph. Statement nodes retain their ground-truth labels for fake news classification (ranging from *pants-fire* to *true*), while speaker and topic nodes are assigned a dummy label. We also preserve a node-level string attribute containing the raw statement, speaker name, or topic for use in explanations and visualizations.
4. AMAZON-PRODUCT: It is a network with nodes representing different product categories on Amazon and edges connecting co-purchased products. The nodes belong to one of 47 product categories. We use only a subset of the AMAZON-PRODUCT dataset, consisting of the first 1000 products and their co-purchase edges, as the entire dataset is very large, and call it AMAZON in our work. The node classification task is to predict a product's category based on its own features and those of its neighbors. For nodes, bag-of-words representations are derived from node textual attributes.

B.2 Relevance of datasets

The datasets used in our work are selected for their relevance to graph explainability tasks, enabling us to assess the effectiveness and robustness of our methods across a diverse set of graph structures and features. The chosen datasets span a variety of domains, including citation networks, text-attributed graphs, fake news detection, and e-commerce, and reflect both real-world complexity and common challenges in explainability. The characteristics of each dataset make them suitable for evaluating the performance of our explanation method. Table 7 summarizes the key features of the datasets and explains their relevance.

C Extended Experiments

C.1 GNN Architecture

We extend Table 1 of the paper with additional experiments for AMAZON using two other standard GNN architectures: GIN (Xu et al.) and GAT (Veličković et al., 2017). The results are reported in Table 8.

Method	Model	FIDELITY (%)	SIZE
GNNEXPLAINER	GAT	100.0	10.57
PGEXPLAINER		71.8	4.45
TAGEXPLAINER		63.1	1.03
RANDOM		89.9	7.28
LLM		96.5	4.92
GSPELL		86.8	3.79
GNNEXPLAINER	GIN	100.0	10.57
PGEXPLAINER		87.2	5.15
TAGEXPLAINER		30.9	1.03
RANDOM		84.7	7.28
LLM		86.0	4.90
GSPELL		73.7	3.06

Table 8: Performance comparison on the Products dataset using different GAT and GIN as two other architectures (using "Llama 3.1 8B Instruct"). Higher fidelity and lower size are better.

With GCN architecture, we observed that our method produces the most compact explanations (close to size 1) across all architectures, achieving

a strong balance. However, while explanation size remains minimal for the GAT and GIN models, fidelity drops more noticeably, particularly with GIN (73.7%), suggesting that our projection-based mechanism may be less effective when the model relies heavily on structural information or deep aggregation. In contrast, GNNExplainer maintains the highest fidelity across all architectures, but at the cost of much larger explanations, while PGExplainer and TAGExplainer underperform in fidelity despite moderate sizes. The LLM baseline shows robust fidelity, especially on GAT, but its explanations are substantially less concise. These results suggest that our method offers an excellent fidelity/interpretability trade-off on GCN, highlighting the need for architecture-aware extensions when applied to more structure-based models.

C.2 LLM Backbone

We also evaluate our method across different LLM models and present the results in Table 9. The baseline LLM method, which directly leverages the language model without projecting GNN embeddings, generally achieves higher fidelity scores but produces substantially larger explanations, two to three times larger. In contrast, our method consistently generates more compact explanations—typically around size 1 —while maintaining comparable fidelity. Notably, Llama 3.1 and Mistral backbones show strong fidelity across both methods, whereas smaller models such as GPT-Neo, Phi-3 Mini, and Pythia exhibit lower fidelity overall. These results highlight our method's effectiveness at generating precise explanations while maintaining reasonable alignment with the GNN's predictions and demonstrate its compatibility with different LLM models.

C.3 Framework Component Analysis

To assess the effectiveness of the different components of our approach, we measure the performance of our method with and without the projector and the post-processing. The results are in Table 10. We observe that both fidelity and explanation size decrease across all datasets when we replace the prompt words with the projected GNN embeddings. This might be due to the LLM having to extract knowledge from more abstract representations, making it more selective in its reasoning. We also observe that adding the post-processing step increases fidelity at a lower cost in size than the method without a projector on the WIKICS dataset. From this analysis, we suggest that our method offers a good compromise: achieving high fidelity (slightly lower than directly prompting the LLM with node-associated text) while maintaining a compact explanation size (slightly larger than projected GNN embeddings without post-processing).

D Additional Implementation Details

We provide additional details of the experiments as follows:

D.1 Experimental Setup

Datasets. For all datasets, we use the graph topology as provided, without any modification or filtering of nodes or classes. For AMAZON-PRODUCT, we first randomly subsample 1000 nodes from the full dataset. For all datasets, we use a consistent split protocol with 60% of nodes for training, 10% for validation, and 200 held-out nodes for testing. Across all datasets, we restrict computation trees to at most 2 hops from each node.

Models. We use a GCN (Kipf and Welling, 2017a) as the base GNN model for all datasets. The GCN consists of 3 convolutional layers, with the final layer used for classification. The model is trained for 100 to 400 epochs (tune based on validation set for each dataset) using the Adam optimizer with a batch size of 64. For the hidden dimension, we choose either 64 or 512, depending on the nodes' feature dimension.

We provide the GNN's accuracy where GCN is used as the model architecture for the different folds in Table 11.

For main experiments, we use Meta's Llama-3.1-8B-Instruct model from the HuggingFace library as a frozen LLM.

Projector. The projector is implemented as a two-layer MLP with ReLU activation. The hidden dimension is set to 4x the GNN embedding dimension. The output layer maps the GNN embedding to a vector of size $k \cdot h$, where k is the number of soft prompt tokens and h is the LLM embedding dimension. The mean-pooled representation is used for alignment objectives. We fix $k = 5$ across all datasets.

The projector is trained separately for each dataset using full-batch gradient descent. We use the Adam optimizer, with the learning rate selected via random search over $\{0.001, 0.0005\}$ based on validation performance. The number of training epochs is selected from $\{100, 200, 400\}$ using validation. The loss weights are selected via random

	Llama 3.1 8B Instruct		Mistral 7B Instruct v0.2		GPT-Neo 2.7B		Phi-3 Mini 4k Instruct		Pythia 2.8B	
	Fidelity (%)	Size	Fidelity (%)	Size	Fidelity (%)	Size	Fidelity (%)	Size	Fidelity (%)	Size
LLM	86.0	4.90	88.0	5.14	84.0	2.33	88.0	5.17	84.0	2.34
GSPELL	91.1	4.25	83.0	1.63	80.0	1.32	80.0	1.47	80.0	1.12

Table 9: Results on Products dataset using different pretrained LLMs (for pretrained GCN). Higher fidelity and lower size are better.

	AMAZON		CORA		LIAR		WIKICS	
	Fidelity	Size	Fidelity	Size	Fidelity	Size	Fidelity	Size
LLM	86.0	4.90	94.5	3.27	100.0	2.22	91.4	21.14
LLM+PR	85.8	3.2	94.5	3.27	100.0	1.10	80.2	3.7
LLM+PR+PO	91.1	4.25	86.5	2.07	100.0	1.11	86.5	12.8

Table 10: Performance comparison for the ablation study. LLM denotes prompting the LLM with words only and no projected embeddings. LLM+Pr is LLM with a projector. LLM with projector and post-processing (denoted by LLM+Pr+Po) is our method.

Dataset	Training	Validation	Testing
CORA	0.88	0.81	0.84
WIKICS	0.89	0.82	0.81
AMAZON	0.87	0.73	0.76
LIAR	0.42	0.30	0.29

Table 11: Accuracy of GCN for the node classification task on four datasets.

search over $\{0.1, 0.5, 0.9\}$ on the validation set. The contrastive temperature is fixed to $\tau = 0.2$.

Explanation construction. For each node, we construct explanation subgraphs by including all supporting nodes and excluding all opposing nodes.

LLM inference. We use the default HuggingFace generation settings for the LLM, without modifying temperature or sampling parameters, and set the maximum number of generated tokens to 1024. Prompts are formatted using the chat template when supported by the model, and a fixed prompt template is used across all datasets.

Baselines and evaluation protocol. We compare GSPELL against standard GNN explainers including GNNExplainer, PGExplainer, and TAGExplainer. We use official implementations and PyTorch Geometric implementations where applicable, without additional hyperparameter tuning.

All methods are evaluated under the same GNN backbone for fairness. Experiments are repeated over 5 random seeds, and we report the mean and standard deviation of all metrics.

We discuss additional details related to the training of the base GNN.

D.2 Additional example on AMAZON-PRODUCT

A key advantage of GSPELL is its ability to increase human understandability and trustworthiness by providing *explanation of explanations*. We give an additional example on AMAZON-PRODUCT in Figure 3. The "Reasoning" section of the LLM response is where the LLM provides a descriptive reasoning on how the supporting neighbors can explain the prediction of the target node. On the other hand, the previous GNN explainers simply generate a subgraph that is more abstract and harder to understand without some level of background knowledge.

E Definition of the qualitative metrics

We adapt the different dimensions of the framework in (Cedro and Martens, 2025) to the context of CORA to clarify their definitions, and provide the definition as given to the expert.

1. **Understandability.** Explanations should clearly convey why a given paper was assigned to a particular research topic.
2. **Trustworthiness.** Explanations should help users assess whether the model's classification of a paper can be trusted.
3. **Insightfulness.** Explanations should reveal insights about the applications or connections that might play a role in the classification.
4. **Satisfaction.** Explanations should feel complete and meaningful to users evaluating the model's behavior.
5. **Confidence.** Explanations should help users gain confidence in the correctness of the classification.
6. **Convincingness.** Explanations should be persuasive in justifying the model's decision for a given paper.
7. **Communicability.** Explanations should be expressed in a way that aligns with the user's background knowledge and expectations.
8. **Usability.** Explanations should support practical tasks such as interpreting predictions, or

LLM Prompt

You are analyzing Amazon product reviews and their co-purchase neighborhoods to understand why it has been classified under a specific category. Your task is to determine which of the neighboring products does (not) support the classification of the target product.
Target Product ID: 160
Predicted Category: Health & Personal Care

Product Embedding Representation:
«BEGIN TARGET KEYWORDS» PROJECTED EMBEDDING «END TARGET KEYWORDS»

Neighboring Products:
Each Product below is described by keywords.

Product 200: «BEGIN KEYWORDS» PROJECTED EMBEDDING «END KEYWORDS»
Product 201: «BEGIN KEYWORDS» PROJECTED EMBEDDING «END KEYWORDS»
Product 202: «BEGIN KEYWORDS» PROJECTED EMBEDDING «END KEYWORDS»
Product 203: «BEGIN KEYWORDS» PROJECTED EMBEDDING «END KEYWORDS»
Product 204: «BEGIN KEYWORDS» PROJECTED EMBEDDING «END KEYWORDS»
Product 205: «BEGIN KEYWORDS» PROJECTED EMBEDDING «END KEYWORDS»
Product 206: «BEGIN KEYWORDS» PROJECTED EMBEDDING «END KEYWORDS»
Product 207: «BEGIN KEYWORDS» PROJECTED EMBEDDING «END KEYWORDS»
Product 208: «BEGIN KEYWORDS» PROJECTED EMBEDDING «END KEYWORDS»
Product 209: «BEGIN KEYWORDS» PROJECTED EMBEDDING «END KEYWORDS»
Product 210: «BEGIN KEYWORDS» PROJECTED EMBEDDING «END KEYWORDS»

LLM Response

P200	Summary: Home care and hygiene products.	NO
P201	Summary: Personal grooming and beauty tools.	NO
P202	Summary: Health and wellness supplements.	YES
P203	Summary: Fitness and exercise equipment.	NO
P204	Summary: Skincare and beauty products.	YES
P205	Summary: Health and wellness journals.	YES
P206	Summary: Personal care and hygiene for specific needs.	YES
P207	Summary: Health and wellness supplements.	YES
P208	Summary: Fitness and exercise accessories.	NO
P209	Summary: Health and wellness products for specific needs.	YES
P210	Summary: Health and wellness journals.	YES

Reasoning: The common pattern among the supporting Products is that they are all related to health and wellness, skincare, and personal care, which are all categories within the Health & Personal Care category.

Instructions

For each neighboring product:
1. Summarize the similarity between supporting products in *one sentence*.
2. Clearly state whether this product supports the classification of the Target Product into category Health & Personal Care.

Format:
```
Product <ID>:
Summary: <One sentence>.
Support: YES or NO — Does this product support classification into Health & Personal Care?
```
Base reasoning only on the keywords and proximity to the target product.

Figure 3: Left: prompt with category and embeddings highlighted. Right: model response with summaries, YES/NO verdicts, and reasoning. Below: instructions shown separately.

improving model performance.

F Experimental Setup for the human evaluation of M1 and M2 Scores

The human evaluation was conducted by the authors, who served as domain experts. For the qualitative experiments, one expert reviewed the CORA dataset, and four experts reviewed the AMAZON-PRODUCT data, evaluating the explanations generated by GNNexplainer and GSPELL over 10 node classification explanation tasks. We clarify the experimental setup used to obtain the M1 and M2 scores reported in Table 2.

A) Data Preparation

We randomly selected 10 articles from the Cora dataset, on which a GNN had been trained. For each article, we generated explanations using both **GSPELL** and **GNNExplainer**. To mitigate potential bias, explanations from these methods were randomized and presented in a blind manner, without revealing their source. Each expert was provided with:

- The original article text and the GNN's predicted label.
- Two explanation subgraphs (one per method), anonymized and presented in random order known only to the experimenters.
- For each subgraph, we listed the texts of the nodes it contains, sorted by increasing node index.

B) Evaluation Procedure

- **Before the evaluation**, we provided 3 experts with the evaluation metrics' definitions and asked them to consider how to apply each metric to content from the Cora dataset.
- **During the evaluation**, each expert received

the materials described above and independently rated each explanation on a scale from 1 to 5 for each metric.

C) Compilation of Results

The experimenters collected and aggregated the scores. **Method 1 (M1)** corresponds to GNNExplainer, and **Method 2 (M2)** corresponds to GSPELL. These results are reported in Table 2.