On the Computational Complexity of Explaining Graph Neural Networks

Anonymous authors
Paper under double-blind review

Abstract

We study the computational complexity of *local post-hoc explanations* for graph classification using Graph Neural Networks (GNNs). These explanations clarify why a GNN predicts a label on a given graph. Explaining these predictions is key to their safe deployment and can enable scientific knowledge discovery. *Factual explanations* highlight a subgraph that suffices for the prediction, and *counterfactual explanations* identify a minimal edit that changes it. We use computational complexity to analyse these explanations. We make three key contributions. First, we formalise computational problems that capture factual and counterfactual GNN explanations. Second, we fully characterise their computational complexity. Our complexity analysis treats the GNN as fixed, rather than as part of the input, and establishes hardness even in this setting. Third, we show that for fixed bounded GNNs, these explanations are fixed-parameter tractable on bounded-treewidth graphs. Our results give a systematic basis for comparing different types of GNN explanations and establish tractability for practically relevant graph classes.

1 Introduction

We study the computational complexity of explaining predictions made by Graph Neural Networks (GNNs) (Scarselli et al., 2009; Kipf & Welling, 2017) for binary graph classification tasks. GNNs have proven useful for a wide range of applications, including molecular property prediction (Duvenaud et al., 2015; Gilmer et al., 2017), drug-interaction modelling (Zitnik et al., 2018), and large-scale recommender systems (Ying et al., 2018). Understanding their predictions is fundamental to their safe adoption and can enable scientific knowledge discovery. To this end, a variety of *local post-hoc* GNN explainability methods have been proposed (Ying et al., 2019; Luo et al., 2020; Yuan et al., 2021; Lucic et al., 2022; Ma et al., 2022; 2026). These methods take a trained GNN and explain its prediction on a single input graph. Two notions of explanation are standard in this setting: *factual* explanations (Ying et al., 2019; Luo et al., 2020; Yuan et al., 2021), which identify a part of the input graph that suffices for the prediction, and *counterfactual* explanations (Lucic et al., 2022; Ma et al., 2022), which identify a minimal change to the input that flips the prediction.

Our analysis complements previous work, which has taxonomically and empirically analysed GNN explanation methods (Yuan et al., 2023; Li et al., 2025; Villia et al., 2026; Nerini et al., 2026; Longa et al., 2025). Computational complexity has proven to be a powerful lens for studying explainability (Barceló et al., 2020b; Bassan et al., 2024; 2026). These works have formalised factual and counterfactual explanation as computational problems expressed in terms of *explainability queries*. However, these analyses are tailored to models operating on fixed-dimensional spaces such as $\mathbb{R}^d$ or $\{0,1\}^d$, and do not directly carry over to GNNs, for two reasons. First, a single GNN can classify graphs of arbitrary sizes. We therefore study *data complexity*, keeping the GNN fixed while allowing the input graph to vary. Computational complexity results (Barceló et al., 2020b) that take the classifier as part of the input do not characterise this setting. Second, explanations for graphs are expressed in terms of graph sub-structure, such as subgraphs, edge deletions, and node features. The formal notions of feature-based explanations for vector spaces do not directly transfer to this setting. Moreover, graph structure provides an opportunity to identify tractable cases through structural parameters such as treewidth. We make three main contributions:

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Table 1: Summary of our complexity results for bounded AC-GNNs with a final bounded readout layer. Membership holds for all such GNNs, while hardness is established for a fixed GNN. All problems are FPT for a fixed GNN when parameterised by the treewidth of the input graph.

Query	Task	Data complexity
CSR	Decide whether the given partial instance is sufficient.	coNP-complete
MSR	Decide whether there is a sufficient reason of size $\leq k$.	Σ_2^{P}-complete
FMSR	Return any inclusion-minimal sufficient reason.	$\mathsf{FP}^{\mathsf{NP}}$-complete
CC	Count prediction-preserving completions.	#P-complete
MCR	Decide whether there is a counterfactual at distance $\leq k$.	NP-complete

Formal explanation queries. We formalise explanation queries that capture factual and counterfactual explanations for graph classification as used in the GNN literature. These queries transfer the explanations from vector data (Barceló et al., 2020b) to graphs, thereby formally capturing notions of explanation used in GNN explainability.

Complexity characterisation. For each explainability query, we establish completeness for the respective complexity classes. All results are in *data complexity*, that is, they hold even when the GNN is fixed and only the input graph varies. Moreover, our hardness results hold already for GNNs with bounded aggregation and a bounded final readout.

Fixed-parameter tractability. We show that for a fixed GNN with bounded aggregations (Cuenca Grau et al., 2026) and final bounded readout, the explanation queries are fixed-parameter tractable when parameterised by the treewidth of the input graph. Since real-world graphs such as molecules typically have very small treewidth (Horváth & Ramon, 2010; Yamaguchi et al., 2003), this identifies a practically relevant tractable regime.

Our analysis uses logical characterisations of GNN expressivity (Barceló et al., 2020a; Cuenca Grau et al., 2026) combined with algorithmic meta-theorems (Courcelle, 1990; Courcelle et al., 2001). We use the logical expressivity of a simple GNN architecture with only bounded aggregation over its neighbours and a bounded global readout. For hardness, we construct fixed GNNs of this form expressing suitable logical formulas and reduce classical hard problems, such as vertex cover and 4-colourability, to explanation queries on them. Furthermore, building on Cuenca Grau et al. (2026), this restricted GNN architecture *exactly* expresses a fragment of two-variable logic with counting. We use this characterisation to encode the explanation problems as formulas in monadic second-order logic. Courcelle's theorem yields fixed-parameter tractability in the treewidth of the input graph.

Organisation of the paper. We begin by reviewing related work in Section 2 and introducing the necessary preliminaries in Section 3. In Section 4, we formalise factual and counterfactual GNN explanations as computational problems, and analyse their computational complexity in Section 5. In Section 6, we establish fixed-parameter tractability for all the analysed explanation queries. We conclude and discuss limitations of our work in Section 7. Full proofs are provided in the appendix.

2 Related Work

We discuss methods for factual and counterfactual GNN explanations, followed by complexity-theoretic approaches to analysing explainability. We focus on local post-hoc explanations, i.e., explanations at the instance level (local) obtained from a trained GNN (post-hoc). In the following, we discuss the works closest to our analysis; we refer the reader to Yuan et al. (2023); Longa et al. (2025); Li et al. (2025); Ma et al. (2026) for broader surveys on GNN explainability.

Factual GNN explanations. Factual explanations identify parts of an input graph that support its prediction, often selecting nodes, edges, and node features. Ying et al. (2019); Luo et al. (2020) use mutual-information-based objectives to seek compact subgraphs that preserve the GNN's original prediction. Funke et al. (2023) seek sparse selections of nodes and features whose fixed values preserve the prediction with high probability when the unselected features are perturbed. These

approaches pursue different notions of *sufficiency*: the selected information should preserve the prediction when the remaining information is perturbed. Azzolin et al. (2025b) define minimal sufficient subgraphs that must preserve the prediction across every admissible completion within the input graph. We present a more refined formalisation of this notion as *minimal sufficient reason* for graphs, which treats both graph structure and node attributes in the set of allowed perturbations. We also study *Count Completions*, which counts the admissible completions of a candidate explanation within the input graph that preserve the original prediction. Normalising it provides a quantitative measure of sufficiency, analogous to sufficiency-based faithfulness metrics for GNN explanations (Azzolin et al., 2025a; 2026; Zheng et al., 2024).

Counterfactual GNN explanations. Counterfactual methods seek changes to an input graph that flip a trained GNN's prediction. Lucic et al. (2022) seek a small set of edge deletions that change a prediction. Bajaj et al. (2021); Tan et al. (2022); Qiu et al. (2024) combine counterfactual objectives with additional sparsity and robustness requirements. Generative approaches pursue prediction-changing edits that remain in-distribution (Ma et al., 2022; Chen et al., 2023; Prado-Romero et al., 2024). Our *Minimum Change Required* query formalises the underlying discrete decision problem: Can we flip the GNN prediction by applying k edits, where each one either deletes one edge or flips one attribute value? This models the sparsity criteria used by practical counterfactual methods.

Formalisation and complexity of explainability. Recent work has analysed explanations for classifiers operating on vector data. This includes Bayesian networks (Shih et al., 2018; Darwiche & Hirth, 2020; 2023), decision trees (Audemard et al., 2022), linear classifiers (Marques-Silva et al., 2020), and multi-layer perceptrons (Barceló et al., 2020b; Bassan et al., 2024).

Our formal view is closest to that of Barceló et al. (2020b), who study similar explainability queries for vector data. In their setting, the classifier is part of the input, allowing hardness reductions to construct a different classifier for each instance of a hard problem. We formalise graph analogues of these queries and study their *data complexity*, fixing the GNN while allowing the input graph to vary. Consequently, hardness proofs that rely on varying the classifier do not transfer directly to our setting. Moreover, the graph structure allows us to exploit structural parameters, such as treewidth, to establish fixed-parameter tractability results.

3 Preliminaries

This section introduces the background and notation for graphs, logic, and complexity used throughout the paper. We concentrate on the problem of binary graph classification.

Graphs, partial instances, and completions. We work with finite, simple, undirected graphs with binary node attributes. A graph is a triple $G = (V, E, \lambda)$, where $V(G) = V$ is a finite set of nodes, $E(G) = E$ is a set of undirected edges, and $\lambda = (\lambda_1, \ldots, \lambda_m)$ with $\lambda_i : V \to \{0, 1\}$ specifies the node attributes. We also write $\lambda(v) = (\lambda_1(v), \ldots, \lambda_m(v))$, and omit λ from $G = (V, E)$ whenever G has no attributes. We denote by $N_G(v)$ the neighbours of v in G.

A *partial instance* of G, written $S \preceq G$, is a tuple $S = (E_S, \lambda_S)$, where $E_S \subseteq E$ fixes a set of edges and $\lambda_S = (\lambda_1^S, \ldots, \lambda_m^S)$ is obtained from λ by replacing zero or more attribute values with $\perp$. That is, λ_S fixes some attributes from G, while leaving others unspecified ($\perp$). The *size* of a partial instance S is

$$|S| := |E_S| + \sum_{i=1}^{m} |\{v \in V : \lambda_i^S(v) \neq \perp\}|.$$

A *completion* of S relative to G is a graph $\bar{G} = (V, \bar{E}, \bar{\lambda})$ with $S \preceq \bar{G}$ and $\bar{E} \subseteq E$. That is, a completion replaces all undefined attributes with 0 or 1, and potentially adds edges from G. Thus, the underlying graph $(V, \bar{E})$ of every completion is a spanning subgraph of (V, E).

GNNs and logic. We use a logical characterisation of GNN expressivity. Barceló et al. (2020a) show that *Aggregate-Combine GNNs* (AC-GNNs) can uniformly express node classifiers expressible in *Graded Modal logic* (GML). That is, for each node classifier in GML, there exists an AC-GNN computing the same function. In the following, we briefly revisit AC-GNNs and GML. In an AC-GNN with L layers, each layer $l \in \{0, 1, \ldots, L-1\}$ computes node features using a combination

and aggregation function as follows:

$$h^{(l+1)}(v) = \mathrm{comb}^{(l)}\big(h^{(l)}(v), \mathrm{agg}^{(l)}(\{\{h^{(l)}(u) \mid u \in N_G(v)\}\})\big),$$

where the initial features are defined as $h^{(0)}(v) = (\lambda_1(v), \ldots, \lambda_m(v))$. We denote the computed node features on graph G by $h^G(v) = h^{(L)}(v)$.

Graded modal logic (GML) is a local fragment of two-variable first-order logic with counting in which each formula has one free variable and counting quantifiers range over neighbours. Formally, GML formulas $\psi(x)$ are constructed as

$$\psi(x) ::= P_i(x) \mid \neg\psi(x) \mid \psi(x) \wedge \psi(x) \mid \exists^{\geq p} y\,(E(x,y) \wedge \psi(y)),$$

where P_i with $i \in \{1, \ldots, m\}$ is a predicate symbol and $p \in \mathbb{N}$. We will also use the equivalent notation $G = (V, E, P_1, \ldots, P_m)$ for the graph with predicates $P_i = \{v \in V : \lambda_i(v) = 1\}$. For a GML formula $\psi(x)$, a graph G, and a vertex v we write $G \models \psi(v)$ to denote that G satisfies ψ if x is interpreted by v. Graded modal logic with global counting (GMLC) extends GML with global counting quantifiers of the form $\exists^{\geq p} y\,\varphi(y)$ where φ is a GMLC formula. Unlike graded modal quantifiers, which count neighbours of the current vertex, global counting quantifiers range over all vertices of the graph. Formally, $G \models \exists^{\geq p} y\,\varphi(y)$ iff $|\{v \in V : G \models \varphi(v)\}| \geq p$. Since $\varphi(y)$ has at most one free variable, the global counting quantifier always produces a closed sentence. Thus, GMLC can combine local node properties with global graph properties.

While every node classifier in GML is captured by an AC-GNN, the converse does not hold: Not every AC-GNN can be expressed by a fixed formula in GML. This is because AC-GNNs can process unbounded multiplicities in the aggregation function. To prevent this, Cuenca Grau et al. (2026) proposed *bounded GNNs*. A function f is K-bounded if $f(M) = f(M_K)$ for all multisets M, where M_K is the multiset obtained from M by capping all multiplicities at K. An AC-GNN is K-bounded if all used aggregation functions are K-bounded. They show that bounded AC-GNNs are exactly as expressive as GML node classifiers.

Since we consider graph classification, we transfer their result to GMLC graph classifiers. To obtain equivalent GNNs, we apply a final bounded readout layer followed by a classification head. More formally, we define

$$h^G := \mathrm{read}(\{\{h^G(v) : v \in V\}\})$$

and consider the GNN $\mathcal{M}(G) = \mathrm{cls}(h^G)$ for an arbitrary classification head. Note that we only introduce a readout in the final layer. We call $\mathcal{M}$ a *bounded AC-GNN with bounded readout* if the node features are computed by a bounded AC-GNN and the final readout is bounded. The following result shows that these GNNs are precisely as expressive as GMLC graph classifiers.

Theorem 1. *A Boolean graph classifier is computable by a bounded AC-GNN with final bounded readout if and only if it is definable by a* GMLC *sentence.*

This conversion is the basis for all hardness and fixed-parameter tractability results. Throughout the paper, we denote by $\mathcal{M}_\varphi$ a bounded AC-GNN with bounded readout that computes the same function as the GMLC sentence φ.

All hardness proofs throughout the paper use GMLC sentences and, hence, apply to bounded AC-GNNs with a final bounded readout and all more expressive architectures. The matching upper bounds only require that the GNN can be evaluated in polynomial time. That is, all completeness results in Section 5 apply to all GNN architectures that can be evaluated in polynomial time and are at least as expressive as bounded AC-GNNs. The results in Section 6 establish fixed-parameter tractability for fixed GNNs expressible by GMLC graph classifiers, covering bounded AC-GNNs with final bounded readout. In the remainder of this paper, by GNNs we mean bounded AC-GNNs with a final bounded readout.

Computational complexity. We study decision, search, and counting problems. For background on computational complexity, we refer the reader to Arora & Barak (2009). The *decision* complexity classes relevant to this paper are P, NP, coNP, and Σ_2^{P}. For all decision problems, we use polynomial-time many-one reductions. The task in *search problems* is to output an arbitrary valid solution (or state that there is none) rather than decide whether a solution exists. We write FP for the class of search problems solvable in deterministic polynomial time and $\mathsf{FP}^{\mathsf{NP}}$ for the class of search

problems solvable in deterministic polynomial time with access to an NP oracle. The task in *counting problems* is to count the number of valid solutions to a given problem. We are interested in the class #P consisting of all counting problems whose solution is the number of accepting paths of a nondeterministic Turing machine running in polynomial time. For both search and counting problems, we use polynomial-time Turing reductions.

For all three types of problems, we also study the *parametrised complexity*. A parameterisation assigns to every instance I a parameter $\kappa(I) \in \mathbb{N}$. A problem is *fixed-parameter tractable* (FPT) with respect to κ if every instance I of size n can be solved in time $f(\kappa(I)) \cdot n^{O(1)}$ for some computable function f. We are particularly interested in the *treewidth* of a graph (Cygan et al., 2015). Intuitively, this parameter measures how similar the graph is to a tree. Throughout the paper, we investigate the *data complexity* of the problems under consideration. That is, the GNN $\mathcal{M}$ is fixed and not part of the input.

4 Formal Explanations for Graph Neural Networks

This section formalises the GNN explainability queries we analyse. Throughout, we fix a GNN $\mathcal{M}$.

Sufficient reasons. Let $\mathcal{M}$ be a fixed GNN and G be a graph. A *sufficient reason* for G with respect to $\mathcal{M}$ is a partial instance $S \preceq G$ such that $\mathcal{M}(G) = \mathcal{M}(\bar{G})$ for all completions $\bar{G}$ of S.

This definition formalises the requirement that the information fixed by S determines the prediction on every admissible completion within G. Similar criteria or approximate versions thereof are required by many practical GNN explainability methods (Ying et al., 2019; Luo et al., 2020; Azzolin et al., 2025b; 2026). We observe that every graph has a sufficient reason with respect to every GNN, as the graph itself is the trivial sufficient reason. We first define the problem of checking whether a given partial instance is sufficient.

> CheckSufficientReason (CSR)
> **Instance:** Graph G, partial instance $S \preceq G$.
> **Task:** Decide whether S is a sufficient reason for G with respect to $\mathcal{M}$.

In practice, it is often beneficial for explanations to be small, as small explanations are naturally more interpretable and informative. We formalise this idea as a decision problem that asks whether a sufficient reason of size at most k exists. Note that hardness of the decision problem directly implies hardness of finding a sufficient reason of minimum cardinality.

> MinimumSufficientReason (MSR)
> **Instance:** Graph G, and $k \in \mathbb{N}$.
> **Task:** Decide whether there is a sufficient reason for G w.r.t. $\mathcal{M}$ of size $\leq k$.

Requiring a sufficient reason to have minimum cardinality can be too restrictive. We therefore also consider *inclusion-minimal sufficient reasons* (Azzolin et al., 2025b). A sufficient reason S for G is *minimal* if there is no other sufficient reason $S' \preceq G$, $S' \neq S$ such that $E(S') \subseteq E(S)$, and $\lambda_{S'}$ specifies only node attributes specified by λ_S. Every sufficient reason contains a minimal sufficient reason. Thus, for any k, a sufficient reason of size at most k exists if and only if a minimal sufficient reason of size at most k exists. The size-bounded decision problem therefore coincides with MSR. We thus study the corresponding search problem of finding a minimal sufficient reason.

> FindMinimalSufficientReason (FMSR)
> **Instance:** Graph G.
> **Task:** Find a minimal sufficient reason S for G with respect to $\mathcal{M}$.

Throughout the paper, *minimum sufficient reasons* are cardinality-minimal, while *minimal sufficient reasons* are inclusion-minimal.

Count completions and probabilistic sufficiency. Many practical approaches aim for a notion of probabilistic sufficiency (Funke et al., 2023; Zheng et al., 2024). They search for or check a partial instance whose completions preserve the GNN prediction with high probability. For uniformly sampled completions, we formalise this by counting the prediction-preserving completions.

CountCompletions (CC)
Instance: Graph G, partial instance $S \preceq G$.
Task: Count the completions $\bar{G}$ of S relative to G satisfying $\mathcal{M}(\bar{G}) = \mathcal{M}(G)$.

Since we can easily compute the number N of overall completions, the fraction $\mathrm{CC}_{\mathcal{M}}(G, S)/N$ gives us the probability that a completion sampled uniformly at random preserves the GNN prediction.

Counterfactual explanations. Counterfactual explanations are graphs that flip the GNN prediction, while being similar to the original graph. To measure this similarity, we usually fix a set of allowed changes. In this paper, we consider *attribute flips* (Tan et al., 2022) and *edge deletions* (Lucic et al., 2022). Thus, for graphs G and G' with $V(G') = V(G)$ and $E(G') \subseteq E(G)$ we define

$$d(G, G') := |E(G) \setminus E(G')| + \sum_{i=1}^{m} |\{v \in V : \lambda_i^G(v) \neq \lambda_i^{G'}(v)\}|.$$

For a GNN $\mathcal{M}$ and a graph G, we say that G' is a *counterfactual at distance* k if $d(G, G') = k$ and $\mathcal{M}(G') \neq \mathcal{M}(G)$.

MinimumChangeRequired (MCR)
Instance: Graph G, and $k \in \mathbb{N}$.
Task: Decide whether there is a counterfactual at distance at most k.

5 Computational Complexity of GNN Explanations

This section presents our main complexity results for the explainability queries defined in Section 4. Theorem 1 shows that bounded AC-GNNs with only one final bounded readout can express all sentences in GMLC. By showing hardness for specific GMLC graph classifiers, we establish hardness of explanations already for restricted GNN architectures. We provide detailed proofs for all theorems in the Appendix. We begin with the analysis of checking (or verifying) sufficient reasons. As the following theorem shows, this problem is already intractable in general.

Theorem 2. *CheckSufficientReason is* coNP*-complete in data complexity.*

Proof. For membership, fix a GNN $\mathcal{M}$. If S is not a sufficient reason for G, there exists a completion $\bar{G}$ of S relative to G such that $\mathcal{M}(\bar{G}) \neq \mathcal{M}(G)$. Since this completion has polynomial size and can be verified in polynomial time, this shows that $\mathrm{CSR}_{\mathcal{M}} \in$ coNP.

For hardness, we reduce from Non-4-Colourability. Let $H = (V, E)$ be a graph for which we want to decide Non-4-Colourability. Without loss of generality, assume that E is not empty. We define the graph $G = (V, E, \lambda_G)$ that keeps the nodes and edges from H and adds two node attributes $\lambda_G(v) = (1, 1)$ for all $v \in V$. We denote these attributes using unary predicates $(P_1(v), P_2(v))$. The attributes encode four distinct colours, which we define with the following GML formulas:

$$c_1(x) = \neg P_1(x) \wedge \neg P_2(x), \qquad c_3(x) = P_1(x) \wedge \neg P_2(x),$$
$$c_2(x) = \neg P_1(x) \wedge P_2(x), \qquad c_4(x) = P_1(x) \wedge P_2(x).$$

We use these formulas to construct the fixed GMLC sentence

$$\varphi := \exists x \bigvee_{i=1}^{4} \big(c_i(x) \wedge \exists y(E(x, y) \wedge c_i(y))\big).$$

Thus, φ holds precisely when the encoded colours induce a monochromatic edge. By Theorem 1, we fix an equivalent GNN $\mathcal{M}_\varphi$. Since λ_G assigns the same colour to each node and we assumed $E \neq \emptyset$, we have $\mathcal{M}_\varphi(G) = 1$. Let $S = (E, \lambda_S)$ with $\lambda_S(v) = (\bot, \bot)$ for all $v \in V$. Since $E(S) = E$, all completions $\bar{G}$ of S relative to G have edge set $E(\bar{G}) = E$. Since S leaves all attributes undefined, the completions of S correspond to all assignments of four colours to V. S is sufficient if and only if $\mathcal{M}_\varphi(\bar{G}) = 1$ for all completions $\bar{G}$. By the above argument, this holds precisely if each assignment of four colours induces a monochromatic edge, which is equivalent to H not being 4-colourable. The construction is polynomial and $\mathcal{M}_\varphi$ is fixed, establishing coNP-hardness in data complexity. □

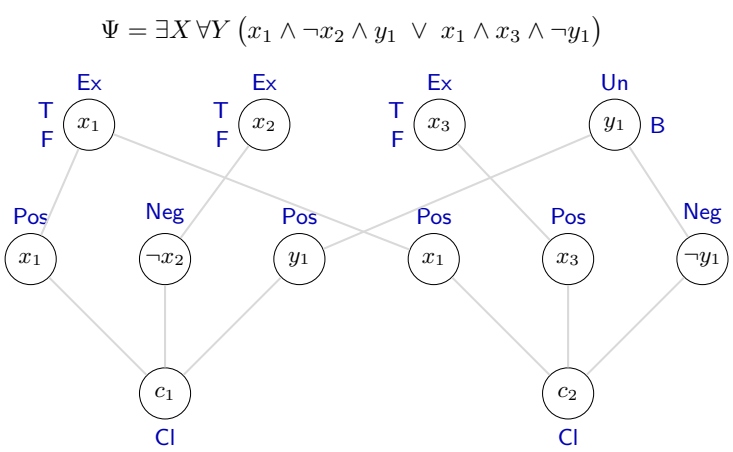

Figure 1: Illustration of the graph encoding used in the Σ_2^{P}-hardness proof for minimum sufficient reasons in Theorem 3. Vertices represent variables, literal occurrences, and clauses. Each literal vertex is connected to its corresponding variable and clause vertices. Node attributes are in blue.

Next, we investigate the minimisation problem. We show that determining whether a sufficient reason of size at most k exists is Σ_2^{P}-complete in general.

Theorem 3. *MinimumSufficientReason is Σ_2^{P}-complete in data complexity.*

Proof sketch. MSR has the $\exists\forall$ structure typical for Σ_2^{P} problems: *There exists* a partial instance S such that *for all* completions of S relative to G, the evaluation of the GNN is the same. For membership, given G and a size bound k, we nondeterministically guess a partial instance $S \preceq G$ of size at most k and check its sufficiency using a $\mathrm{CSR}_{\mathcal{M}}$ oracle. Since this problem belongs to coNP, this establishes membership in $\mathsf{NP}^{\mathsf{coNP}} = \Sigma_2^{\mathsf{P}}$.

For hardness, we reduce from $\exists\forall$-3DNF-SAT. An instance of this problem is a quantified Boolean formula (QBF) $\Psi = \exists X \forall Y \psi(X, Y)$, where $\psi(X, Y)$ is in 3-DNF. The task is to decide whether Ψ is true, that is, whether there exists a truth assignment to the variables in X such that $\psi(X, Y)$ is satisfied for all truth assignments to the variables in Y.

We encode a given QBF using variable vertices, clause vertices, and one vertex per literal occurrence. Further, we add attributes that encode a truth assignment to X and Y. The fixed GNN checks whether G encodes a valid QBF, and whether one clause is satisfied under the encoded variable assignment. We show that every sufficient reason must fix the encoded QBF structure and determine an assignment to the existential variables X. These requirements yield a lower bound k_Ψ on its size. At this bound, the universal assignment remains unrestricted, and the construction ensures that sufficiency is equivalent to ψ holding for every assignment to Y. Consequently, G_Ψ admits a sufficient reason of size at most k_Ψ if and only if Ψ is true. Since only the graph and size bound depend on Ψ, while the GNN is fixed, this establishes hardness in data complexity. □

Next, we consider the problem of finding a minimal sufficient reason. Since this is formulated as a search problem, the complexity class differs from that in standard (decision) complexity. It turns out that the problem falls in the class $\mathsf{FP}^{\mathsf{NP}}$ of search problems that can be solved in polynomial time with access to an NP oracle. This means that (under standard complexity assumptions), although easier than MSR, finding an inclusion-minimal sufficient reason remains intractable in general.

Theorem 4. *FindMinimalSufficientReason is $\mathsf{FP}^{\mathsf{NP}}$-complete in data complexity.*

Proof sketch. For membership, we fix a GNN $\mathcal{M}$. We use a simple greedy deletion algorithm, starting with a fully specified instance S: For each specified edge or node attribute, we use a $\mathrm{CSR}_{\mathcal{M}}$ oracle to decide whether we can remove it from S. After checking each element once, we return the obtained partial instance. By the monotonicity of sufficient reasons, this partial instance is minimal.

For hardness, we show that we can simulate any NP oracle using FMSR. More specifically, we reduce from Non-4-Colourability in such a way that the reduction works for an arbitrary FMSR solution. The construction is similar to the one in the proof of Theorem 2. However, we must additionally ensure that *each* minimal sufficient reason solves the colourability problem. This does not hold for the previous construction, since a single monochromatic edge already certifies that a colouring is invalid. To overcome this, we modify $\mathcal{M}$ such that it additionally requires the instance encoding to be specified. Thus, every sufficient reason contains the entire graph structure and may only leave the colour attributes unspecified. We show that the instance encoding alone is the *unique* minimal sufficient reason if and only if the input graph H is not 4-colourable. □

Theorem 5. *CountCompletions is #P-complete in data complexity.*

Proof sketch. For membership, fix a GNN $\mathcal{M}$. Given a graph G and a partial instance $S \preceq G$, nondeterministically guess a completion $\bar{G}$ of S relative to G. If $\mathcal{M}(\bar{G}) = \mathcal{M}(G)$, accept it. The number of accepting computations equals the number of label-preserving completions.

For hardness, we reduce from #IndependentSet (similar to Kula et al., 2026). We add a single attribute λ to the input graph G that is set to $\lambda_G(v) = 0$ everywhere. Let S be the partial instance that fixes all edges but no attributes. Thus, there is a bijection between subsets of $V(G)$ and completions of S. We construct a fixed GNN $\mathcal{M}$ with $\mathcal{M}(G) = 1$ iff the set of vertices with $\lambda(v) = 1$ is an independent set. Since $\lambda_G(v) = 0$ for all nodes v in G, $\mathcal{M}(G) = 1$. Thus, the prediction-preserving completions $\bar{G}$ of S correspond bijectively to independent sets in G. This shows that counting completions of S is equivalent to counting independent sets. □

Together with Theorem 2, this shows that both deciding whether a partial instance determines the classifier prediction and exactly computing the probability that it implies the decision are intractable.

Finally, we consider counterfactual explanations. We show that, in general, it is intractable to decide whether a counterfactual at distance at most k exists.

Theorem 6. *MinimumChangeRequired is NP-complete in data complexity.*

Proof sketch. For membership, fix a GNN $\mathcal{M}$. A counterfactual for $\mathcal{M}(G)$ is a graph $\bar{G}$ with $\mathcal{M}(\bar{G}) \neq \mathcal{M}(G)$ that is obtained from G by removing edges and flipping node attributes. Guessing at most k changes and verifying $\mathcal{M}(G) \neq \mathcal{M}(\bar{G})$ for the resulting graph $\bar{G}$ shows $\mathrm{MCR}_{\mathcal{M}} \in \mathsf{NP}$.

For hardness, we reduce from VertexCover similar to Barceló et al. (2020b). Given an input graph $H = (V, E)$ with $E \neq \emptyset$, we construct a graph G with the same structure and one attribute λ_G that is 0 everywhere. We construct a fixed GNN $\mathcal{M}$ that predicts one if and only if the set of nodes v with attribute $\lambda(v) = 1$ forms a vertex cover. Since E is not empty, this implies $\mathcal{M}(G) = 0$. We then prove that H has a vertex cover X of size at most k iff G has a counterfactual at distance at most k. The first direction is clear since flipping $\lambda(v)$ for all nodes $v \in X$ yields a counterfactual at distance $\leq k$. Conversely, we show that a counterfactual at distance k with edge changes can always be transformed into one at distance $\leq k$ without edge changes by setting λ to 1 for one endpoint per deleted edge. This gives a vertex cover of size at most k and proves the theorem. □

6 Explanations for Bounded-treewidth Graphs

The results in Section 5 show that all considered explainability queries are hard in general. In this section, we show that all queries become fixed-parameter tractable when parameterised by treewidth. Intuitively, the treewidth $\mathrm{tw}(G)$ of a graph G measures how closely G resembles a tree. Our approach is to express the explainability queries in *monadic second-order logic* (MSO_2). This logic extends first-order logic with variables ranging over sets of vertices or edges, allowing us to quantify over partial graphs and their completions. By Courcelle (1990)'s theorem, model checking a fixed MSO_2 formula is FPT, parameterised by $\mathrm{tw}(G)$. Extensions of this theorem also support counting satisfying assignments and finding satisfying assignments of minimum weight (Arnborg et al., 1991; Courcelle et al., 2001). By expressing our explainability queries in MSO_2 and applying these meta-theorems, we establish that all considered problems are FPT, parameterised by $\mathrm{tw}(G)$.

Theorem 7. *For every fixed bounded AC-GNN with final bounded readout, all five explainability queries are FPT parameterised by* $\mathrm{tw}(G)$.

Proof sketch. By Theorem 1, the fixed GNN $\mathcal{M}$ is equivalent to a fixed GMLC sentence φ. We express the explainability queries for φ in MSO_2.

We represent a graph G_X by a tuple $X = (X_E, X_1, \ldots, X_m)$, where X_E specifies its edges and X_i specifies the nodes with attribute P_i. Similarly, a tuple $Y = (Y_E, Y_1, \ldots, Y_m)$ represents a partial instance S_Y, where Y_E contains the fixed edges and Y_i contains the nodes whose i-th attribute is fixed. We express completions using the MSO_2 formula

$$\mathrm{Comp}(X, Y) := Y_E \subseteq X_E \wedge \bigwedge_{i=1}^{m} (X_i \cap Y_i = P_i \cap Y_i).$$

Since edge-set variables range over $E(G)$, this captures precisely the completions relative to G.

We translate φ into a fixed MSO_2 formula $\hat{\varphi}(X)$ that holds exactly when $G_X \models \varphi$. We can then express sufficient reasons as

$$\mathrm{SR}_\varphi(Y) := \forall X \left(\mathrm{Comp}(X, Y) \rightarrow (\varphi \leftrightarrow \hat{\varphi}(X))\right).$$

This formula allows us to check whether a given partial instance is sufficient (CSR), or to find a sufficient reason of minimum size by minimising $|Y_E| + \sum_i |Y_i|$ (MSR). We obtain an inclusion-minimal sufficient reason (FMSR) using the greedy deletion algorithm, with sufficiency checked after each deletion. For CC, we fix the assignment Y_S and count the assignments to X satisfying

$$\mathrm{Comp}(X, Y_S) \wedge (\varphi \leftrightarrow \hat{\varphi}(X)).$$

Finally, for MCR, observe that every counterfactual at distance k induces a nonsufficient partial instance of size $N - k$, where $N = |E(G)| + m|V(G)|$. Conversely, every nonsufficient partial instance S_Y admits a counterfactual at distance at most $N - |S_Y|$. Thus, the minimum change required is obtained by maximising $|Y_E| + \sum_i |Y_i|$ subject to $\neg\,\mathrm{SR}_\varphi(Y)$, or determining that no such partial instance exists.

Since all formulas are fixed, the model-checking (Courcelle, 1990), optimisation (Arnborg et al., 1991), and counting (Courcelle et al., 2001) versions of Courcelle's theorem establish fixed-parameter tractability for all five queries, parameterised by $\mathrm{tw}(G)$. □

The proof extends to any fixed graph classifier expressible in MSO_2. Indeed, the only property of bounded AC-GNNs used in the proof is that they can be expressed by an MSO_2 formula.

7 Conclusion, Limitations and Future Work

We formalised fundamental GNN explainability queries and characterised their computational complexity. All five queries are intractable in general under standard complexity assumptions, with hardness already arising for fixed bounded AC-GNNs with a final bounded readout. Thus, fixing the classifier does not eliminate the computational difficulty of explaining its predictions. Nevertheless, for every fixed GNN in this class, all five queries are fixed-parameter tractable when parameterised by the treewidth of the input graph. These results identify graph structure as a source of tractability and motivate a more fine-grained analysis of GNN explainability.

Our framework focuses on binary graph classification with Boolean node attributes. Completions and counterfactuals preserve the node set and use only edges present in the original graph. We therefore do not analyse perturbations involving node additions or deletions, edge additions, or continuous attributes. Extending the framework to such perturbations would allow us to study explanations whose admissible perturbations reflect the application domain more closely.

Further directions include investigating parameterisations by explanation size or counterfactual distance, and determining which additional structural restrictions yield tractability. Our treewidth results also leave open the development of practical algorithms: the general logical meta-theorems establish FPT guarantees but do not by themselves ensure useful running times. Designing specialised algorithms for these cases, as well as approximation methods for counting prediction-preserving completions, would help connect the complexity classification to practical explanation methods.

AI USE STATEMENT

In this work, we used generative AI tools to support research ideation, assist in developing and refining mathematical formulations and proofs, and identify relevant literature. We also used these tools to write, edit, and improve parts of the manuscript, such as individual paragraphs of proof text. Our use was aimed at improving the paper's structure and readability and identifying potential errors and inconsistencies. We have critically reviewed all AI-assisted content and independently checked the mathematical arguments and cited references. We take responsibility for the final content of this work, including the text, claims, and formal results presented in the paper.

REPRODUCIBILITY STATEMENT

All assumptions and definitions are clearly presented in the main body. Detailed proofs for all the claims are provided in the Appendix, along with the proof sketches in the main body.

REFERENCES

Stefan Arnborg, Jens Lagergren, and Detlef Seese. Easy problems for tree-decomposable graphs. *J. Algorithms*, 12(2):308–340, 1991.

Sanjeev Arora and Boaz Barak. *Computational Complexity - A Modern Approach*. Cambridge University Press, 2009. ISBN 978-0-521-42426-4. URL http://www.cambridge.org/catalogue/catalogue.asp?isbn=9780521424264.

Gilles Audemard, Steve Bellart, Louenas Bounia, Frédéric Koriche, Jean-Marie Lagniez, and Pierre Marquis. On the explanatory power of boolean decision trees. *Data Knowl. Eng.*, 142:102088, 2022.

Steve Azzolin, Antonio Longa, Stefano Teso, and Andrea Passerini. Reconsidering faithfulness in regular, self-explainable and domain invariant gnns. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025a. URL https://openreview.net/forum?id=kiOxNsrpQy.

Steve Azzolin, Sagar Malhotra, Andrea Passerini, and Stefano Teso. Beyond topological self-explainable gnns: A formal explainability perspective. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*, pp. 2045–2080. PMLR / OpenReview.net, 2025b. URL https://proceedings.mlr.press/v267/azzolin25a.html.

Steve Azzolin, Stefano Teso, Bruno Lepri, Andrea Passerini, and Sagar Malhotra. GNN explanations that do not explain and how to find them. *CoRR*, abs/2601.20815, 2026. doi: 10.48550/ARXIV.2601.20815. URL https://doi.org/10.48550/arXiv.2601.20815.

Mohit Bajaj, Lingyang Chu, Zi Yu Xue, Jian Pei, Lanjun Wang, Peter Cho-Ho Lam, and Yong Zhang. Robust counterfactual explanations on graph neural networks. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 5644–5655, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/2c8c3a57383c63caef6724343eb62257-Abstract.html.

Pablo Barceló, Egor V. Kostylev, Mikaël Monet, Jorge Pérez, Juan L. Reutter, and Juan Pablo Silva. The logical expressiveness of graph neural networks. In *ICLR*. OpenReview.net, 2020a.

Pablo Barceló, Mikaël Monet, Jorge Pérez, and Bernardo Subercaseaux. Model interpretability through the lens of computational complexity. In *NeurIPS*, 2020b.

Shahaf Bassan, Guy Amir, and Guy Katz. Local vs. global interpretability: A computational complexity perspective. In *ICML*, Proceedings of Machine Learning Research, pp. 3133–3167. PMLR / OpenReview.net, 2024.

Shahaf Bassan, Xuanxiang Huang, and Guy Katz. Unifying formal explanations: A complexity-theoretic perspective. *CoRR*, abs/2602.18160, 2026. doi: 10.48550/ARXIV.2602.18160. URL https://doi.org/10.48550/arXiv.2602.18160.

Jialin Chen, Shirley Wu, Abhijit Gupta, and Rex Ying. D4explainer: In-distribution explanations of graph neural network via discrete denoising diffusion. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/f978c8f3b5f399cae464e85f72e28503-Abstract-Conference.html.

Bruno Courcelle. The monadic second-order logic of graphs. i. recognizable sets of finite graphs. *Inf. Comput.*, 85(1):12–75, 1990.

Bruno Courcelle, Johann A. Makowsky, and Udi Rotics. On the fixed parameter complexity of graph enumeration problems definable in monadic second-order logic. *Discret. Appl. Math.*, 108(1-2): 23–52, 2001.

Bernardo Cuenca Grau, Eva Feng, and Przemyslaw Andrzej Walega. The correspondence between bounded graph neural networks and fragments of first-order logic. In *AAAI*, pp. 19135–19142. AAAI Press, 2026.

Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015.

Adnan Darwiche and Auguste Hirth. On the reasons behind decisions. In *ECAI*, Frontiers in Artificial Intelligence and Applications, pp. 712–720. IOS Press, 2020.

Adnan Darwiche and Auguste Hirth. On the (complete) reasons behind decisions. *J. Log. Lang. Inf.*, 32(1):63–88, 2023. doi: 10.1007/S10849-022-09377-8. URL https://doi.org/10.1007/s10849-022-09377-8.

David Duvenaud, Dougal Maclaurin, Jorge Aguilera-Iparraguirre, Rafael Gómez-Bombarelli, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P. Adams. Convolutional networks on graphs for learning molecular fingerprints. In Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pp. 2224–2232, 2015. URL https://proceedings.neurips.cc/paper/2015/hash/f9be311e65d81a9ad8150a60844bb94c-Abstract.html.

Thorben Funke, Megha Khosla, Mandeep Rathee, and Avishek Anand. Zorro: Valid, sparse, and stable explanations in graph neural networks. *IEEE Trans. Knowl. Data Eng.*, 35(8):8687–8698, 2023. doi: 10.1109/TKDE.2022.3201170. URL https://doi.org/10.1109/TKDE.2022.3201170.

Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1263–1272. PMLR, 2017. URL http://proceedings.mlr.press/v70/gilmer17a.html.

Tamás Horváth and Jan Ramon. Efficient frequent connected subgraph mining in graphs of bounded tree-width. *Theor. Comput. Sci.*, 411(31-33):2784–2797, 2010. doi: 10.1016/J.TCS.2010.03.030.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=SJU4ayYgl.

Václav Kula, Qipeng Kuang, Yuyi Wang, Yuanhong Wang, and Ondrej Kuzelka. Tractable weighted first-order model counting with bounded treewidth binary evidence. In *AAAI*, pp. 19198–19207. AAAI Press, 2026.

Xuyan Li, Jie Wang, and Zheng Yan. Can graph neural networks be adequately explained? A survey. *ACM Comput. Surv.*, 57(5):131:1–131:36, 2025. doi: 10.1145/3711122. URL https://doi.org/10.1145/3711122.

Antonio Longa, Steve Azzolin, Gabriele Santin, Giulia Cencetti, Pietro Lio, Bruno Lepri, and Andrea Passerini. Explaining the explainers in graph neural networks: a comparative study. *ACM Comput. Surv.*, 57(5):120:1–120:37, 2025.

Ana Lucic, Maartje A. ter Hoeve, Gabriele Tolomei, Maarten de Rijke, and Fabrizio Silvestri. Cf-gnnexplainer: Counterfactual explanations for graph neural networks. In *AISTATS*, volume 151 of *Proceedings of Machine Learning Research*, pp. 4499–4511. PMLR, 2022.

Dongsheng Luo, Wei Cheng, Dongkuan Xu, Wenchao Yu, Bo Zong, Haifeng Chen, and Xiang Zhang. Parameterized explainer for graph neural network. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/e37b08dd3015330dcbb5d6663667b8b8-Abstract.html.

Jing Ma, Ruocheng Guo, Saumitra Mishra, Aidong Zhang, and Jundong Li. CLEAR: generative counterfactual explanations on graphs. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/a69d7f3a1340d55c720e572742439eaf-Abstract-Conference.html.

Wenzheng Ma, Xiaofeng Liu, Yihu Liu, and Yinglong Ma. Post-hoc explainability of graph neural networks: A comprehensive survey. *Inf. Sci.*, 740:123202, 2026. doi: 10.1016/J.INS.2026.123202. URL https://doi.org/10.1016/j.ins.2026.123202.

João Marques-Silva, Thomas Gerspacher, Martin C. Cooper, Alexey Ignatiev, and Nina Narodytska. Explaining naive bayes and other linear classifiers with polynomial time and delay. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/eccd2a86bae4728b38627162ba297828-Abstract.html.

Francesco Paolo Nerini, Mirko Zaffaroni, Paolo Baracco, Gabriele Ciravegna, and Alan Perotti. Measuring what matters: A unified evaluation framework for GNN explainability. *CoRR*, abs/2607.04600, 2026.

Mario Alfonso Prado-Romero, Bardh Prenkaj, and Giovanni Stilo. Robust stochastic graph generator for counterfactual explanations. In Michael J. Wooldridge, Jennifer G. Dy, and Sriraam Natarajan (eds.), *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2024, February 20-27, 2024, Vancouver, Canada*, pp. 21518–21526. AAAI Press, 2024. doi: 10.1609/AAAI.V38I19.30149. URL https://doi.org/10.1609/aaai.v38i19.30149.

Dazhuo Qiu, Mengying Wang, Arijit Khan, and Yinghui Wu. Generating robust counterfactual witnesses for graph neural networks. In *40th IEEE International Conference on Data Engineering, ICDE 2024, Utrecht, The Netherlands, May 13-16, 2024*, pp. 3351–3363. IEEE, 2024. doi: 10.1109/ICDE60146.2024.00259. URL https://doi.org/10.1109/ICDE60146.2024.00259.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The Graph Neural Network Model. *IEEE Transactions on Neural Networks (TNN)*, 20(1):61–80, 2009. ISSN 1045-9227. doi: 10.1109/TNN.2008.2005605. URL http://ieeexplore.ieee.org/document/4700287/.

Andy Shih, Arthur Choi, and Adnan Darwiche. A symbolic approach to explaining bayesian network classifiers. In Jérôme Lang (ed.), *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, pp. 5103–5111. ijcai.org, 2018. doi: 10.24963/IJCAI.2018/708. URL https://doi.org/10.24963/ijcai.2018/708.

Larry J. Stockmeyer. The polynomial-time hierarchy. *Theor. Comput. Sci.*, 3(1):1–22, 1976.

Juntao Tan, Shijie Geng, Zuohui Fu, Yingqiang Ge, Shuyuan Xu, Yunqi Li, and Yongfeng Zhang. Learning and evaluating graph neural network explanations based on counterfactual and factual reasoning. In Frédérique Laforest, Raphaël Troncy, Elena Simperl, Deepak Agarwal, Aristides Gionis, Ivan Herman, and Lionel Médini (eds.), *WWW '22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022*, pp. 1018–1027. ACM, 2022. doi: 10.1145/3485447.3511948. URL https://doi.org/10.1145/3485447.3511948.

Maria Myrto Villia, Filippos Gouidis, Theodore Patkos, and Panos Trahanias. A completion-aware framework for impactful counterfactual explainability in graph neural networks. In *ECML PKDD (3)*, volume 16943 of *Lecture Notes in Computer Science*, pp. 291–309. Springer, 2026.

Atsuko Yamaguchi, Kiyoko F. Aoki, and Hiroshi Mamitsuka. Graph complexity of chemical compounds in biological pathways. *Genome Informatics*, 14:376–377, 2003. URL https://www.jstage.jst.go.jp/article/gi1990/14/0/14_0_376/_pdf/-char/ja.

Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L. Hamilton, and Jure Leskovec. Graph convolutional neural networks for web-scale recommender systems. In Yike Guo and Faisal Farooq (eds.), *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*, pp. 974–983. ACM, 2018. doi: 10.1145/3219819.3219890. URL https://doi.org/10.1145/3219819.3219890.

Zhitao Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. Gnnexplainer: Generating explanations for graph neural networks. In *NeurIPS*, pp. 9240–9251, 2019.

Hao Yuan, Haiyang Yu, Jie Wang, Kang Li, and Shuiwang Ji. On explainability of graph neural networks via subgraph explorations. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 12241–12252. PMLR, 2021. URL http://proceedings.mlr.press/v139/yuan21c.html.

Hao Yuan, Haiyang Yu, Shurui Gui, and Shuiwang Ji. Explainability in graph neural networks: A taxonomic survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(5):5782–5799, 2023.

Xu Zheng, Farhad Shirani, Tianchun Wang, Wei Cheng, Zhuomin Chen, Haifeng Chen, Hua Wei, and Dongsheng Luo. Towards robust fidelity for evaluating explainability of graph neural networks. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=up6hr4hIQH.

Marinka Zitnik, Monica Agrawal, and Jure Leskovec. Modeling polypharmacy side effects with graph convolutional networks. *Bioinform.*, 34(13):i457–i466, 2018. doi: 10.1093/BIOINFORMATICS/BTY294. URL https://doi.org/10.1093/bioinformatics/bty294.

A Expressiveness of Bounded AC-GNN Graph Classifiers

Lemma 1. *Every* GMLC *sentence is equivalent to a Boolean combination of global counting sentences of the form* $\exists^{\geq p} x\, \psi(x)$ *with* $\psi(x) \in$ GML.

Proof. Fix a GMLC sentence φ. Every global counting subformula is closed by the syntax of GMLC. We can iteratively reduce the depth of global quantifiers as follows: Choose an innermost global counting subformula $\phi = \exists^{\geq p} y\, \psi(y)$ nested inside another global counting quantifier. Since ϕ has no free variables, it is either true or false in the graph. Let $\varphi^{\top}$ and $\varphi^{\bot}$ be obtained by replacing this occurrence of ϕ with the corresponding truth value. Thus, $\varphi \equiv (\phi \wedge \varphi^{\top}) \vee (\neg\phi \wedge \varphi^{\bot})$. Repeating this process eliminates all nested global counting quantifiers. Thus, we obtain an equivalent sentence of the claimed form. □

Theorem 1. *A Boolean graph classifier is computable by a bounded AC-GNN with final bounded readout if and only if it is definable by a* GMLC *sentence.*

Proof. Fix a bounded AC-GNN $\mathcal{M}$ with B-bounded readout. By Proposition 5 of Cuenca Grau et al. (2026), each bounded AC-GNN can only compute finitely many distinct final node feature vectors. Let $A = \{a_1, \ldots, a_r\}$ be those of $\mathcal{M}$. For each a_i, choosing a classification head that predicts 1 only for a_i yields a GML formula $\psi_i(x)$ with $G \models \psi_i(v)$ iff $h^G(v) = a_i$ by their Corollary 9. Since the readout is B-bounded, the final classification of $\mathcal{M}$ is only based on the tuple $(\min\{B, |\{v \in V : h^G(v) = a_i\}|)_{i=1}^r \in \{0, \ldots, B\}^r$. For each $p \in \{0, \ldots, B-1\}$ and $i \in \{1, \ldots, r\}$ we construct the GMLC sentence $\exists^{=p} x\, \psi_i(x)$, and for $p = B$ the sentence $\exists^{\geq p} x\, \psi_i(x)$. By choosing a suitable Boolean combination of these sentences, we obtain a GMLC graph classifier that is equivalent to $\mathcal{M}$.

Conversely, fix a GMLC graph classifier φ. Since φ has no free variable, it is equivalent to a Boolean combination of sentences $\exists^{\geq p_i} x\, \psi_i(x)$ for $p_i \in \mathbb{N}$ and $\psi_i(x) \in$ GML by Lemma 1. Let B be the maximum of these counts p_i. As before, we use the correspondence between GML and bounded AC-GNNs to compute node features $h_i(v)$ with $h_i(v) = 1$ iff $G \models \psi_i(v)$ for each of these formulas ψ_i. Now, we apply a bounded sum readout to obtain the graph feature

$$(h^G)_i = \min\{B, |\{v \in V : h_i(v) = 1\}|\}.$$

Since B is the maximum count used within φ and $(h^G)_i \geq p_i$ iff $G \models \exists^{\geq p_i} x\, \psi_i(x)$, a suitable classification head yields a bounded AC-GNN $\mathcal{M}$ with bounded readout and $\mathcal{M}(G) = 1$ if and only if $G \models \varphi$. This GNN depends only on the fixed sentence, not on G, since we can compute B directly from φ. □

B Completeness Proofs

This section contains the completeness proofs for all investigated explainability queries.

Theorem 2. *CheckSufficientReason is* coNP*-complete in data complexity.*

Proof. For membership, fix a GNN $\mathcal{M}$. If S is not a sufficient reason for G, there exists a completion $\bar{G}$ of S relative to G such that $\mathcal{M}(\bar{G}) \neq \mathcal{M}(G)$. Since this completion has polynomial size and can be verified in polynomial time, this shows that $\mathrm{CSR}_{\mathcal{M}} \in$ coNP.

For hardness, we reduce from Non-4-Colourability. Let $H = (V, E)$ be a graph for which we want to decide Non-4-Colourability. Without loss of generality, assume that E is not empty. We define the graph $G = (V, E, \lambda_G)$ that keeps the nodes and edges from H and adds two node attributes $\lambda_G(v) = (1, 1)$ for all $v \in V$. We denote these attributes using unary predicates $(P_1(v), P_2(v))$. The attributes encode four distinct colours, which we define with the following GML formulas:

$$\begin{aligned} c_1(x) &= \neg P_1(x) \wedge \neg P_2(x), \\ c_2(x) &= \neg P_1(x) \wedge P_2(x), \\ c_3(x) &= P_1(x) \wedge \neg P_2(x), \\ c_4(x) &= P_1(x) \wedge P_2(x). \end{aligned} \tag{1}$$

We use these formulas to construct the fixed GMLC sentence

$$\varphi := \exists x \bigvee_{i=1}^{4} \big(c_i(x) \wedge \exists y(E(x,y) \wedge c_i(y))\big).$$

Thus, φ holds precisely when the encoded colours induce a monochromatic edge. By Theorem 1, we fix an equivalent GNN $\mathcal{M}_\varphi$. Since λ_G assigns the same colour to each node and we assumed $E \neq \emptyset$, we have $\mathcal{M}_\varphi(G) = 1$. Let $S = (E, \lambda_S)$ with $\lambda_S(v) = (\bot, \bot)$ for all $v \in V$. Since $E(S) = E$, all completions $\bar{G}$ of S relative to G have edge set $E(\bar{G}) = E$. Since S leaves all attributes undefined, the completions of S correspond precisely to all assignments of four colours to V. S is sufficient if and only if $\mathcal{M}_\varphi(\bar{G}) = 1$ for all completions $\bar{G}$. By the above argument, this holds precisely if each assignment of four colours induces a monochromatic edge, which is equivalent to H not being 4-colourable.

The construction is polynomial and $\mathcal{M}_\varphi$ is fixed, establishing coNP-hardness in data complexity. □

Theorem 3. *MinimumSufficientReason is Σ_2^{P}-complete in data complexity.*

Proof. For membership, fix a GNN $\mathcal{M}$. Let (G, k) be the given $\mathrm{MSR}_{\mathcal{M}}$ instance. Recall that $\Sigma_2^{\mathsf{P}} = \mathsf{NP}^{\mathsf{NP}}$ is the class of problems that can be decided in nondeterministic polynomial time, given access to an NP (or, equivalently, coNP) oracle. To decide $\mathrm{MSR}_{\mathcal{M}}$, non-deterministically guess a partial instance $S \preceq G$ of size at most k, and use a $\mathrm{CSR}_{\mathcal{M}}$ oracle to check whether S is sufficient. Since $\mathrm{CSR}_{\mathcal{M}} \in \mathsf{coNP}$ and the algorithm runs in nondeterministic polynomial time, this proves $\mathrm{MSR}_{\mathcal{M}} \in \Sigma_2^{\mathsf{P}}$.

For hardness, we reduce from the Σ_2^{P}-complete problem $\exists\forall$-3DNF-SAT, whose instances are quantified Boolean formulas $\Psi = \exists X \forall Y\, \psi(X, Y)$, with exactly three literal occurrences per clause (Stockmeyer, 1976). The task is to determine whether Ψ is true, that is, whether there exists an assignment α to X such that for all assignments β to Y, $\psi(\alpha, \beta)$ is true.

Let $\Psi = \exists X \forall Y \psi(X, Y)$ be the input QBF with ψ in 3-DNF. We construct $G_\Psi = (V, E, \lambda)$ as illustrated in Figure 2. We create one vertex for each variable in $X \cup Y$, each clause in ψ, and each literal occurrence. Then, we connect every literal vertex to its variable vertex and its clause vertex. We set the attributes according to the following table; all unlisted attribute values are 0.

Table 2: Definition of the node attributes of G_Ψ

Vertex	Attributes set to 1
Existential variable	Ex, T, F
Universal variable	Un, B
Clause	Cl
Positive literal occurrence	Pos
Negative literal occurrence	Neg

The predicates in $\mathcal{T} = \{\mathrm{Ex}, \mathrm{Un}, \mathrm{Pos}, \mathrm{Neg}, \mathrm{Cl}\}$ encode the QBF structure, while the remaining predicates T, F, and B encode the truth assignment to X and Y, respectively.

By adding a fresh universal variable z and the clause $z \wedge z \wedge z$, we may assume that ψ contains a clause that is true under the assignment $\beta = (1, \ldots, 1)$. This preserves the truth of Ψ, since z is universal and setting $z = 0$ recovers the original formula. This will be important to ensure that the constructed MSR instance satisfies $G \models \varphi$.

Before we define the fixed sentence, we construct formulas for later use. We introduce the shortcuts $\mathrm{Var}(x) := \mathrm{Ex}(x) \vee \mathrm{Un}(x)$ describing variable vertices, and $\mathrm{Lit}(x) := \mathrm{Pos}(x) \vee \mathrm{Neg}(x)$ describing literal vertices. We further define the formula

$$\begin{aligned}\mathrm{SatLit}(x) := \mathrm{Pos}(x) \wedge \exists y(E(x,y) \wedge (\mathrm{Ex}(y) \wedge \mathrm{T}(y) \vee \mathrm{Un}(y) \wedge \mathrm{B}(y))) \vee \\ \mathrm{Neg}(x) \wedge \exists y(E(x,y) \wedge (\mathrm{Ex}(y) \wedge \mathrm{F}(y) \vee \mathrm{Un}(y) \wedge \neg\mathrm{B}(y)))\end{aligned}$$

expressing that node x represents a literal that is satisfied under the given variable assignment.

We now construct the fixed GMLC sentence $\varphi := \varphi_1 \wedge \varphi_2 \wedge \varphi_3 \wedge \varphi_4$ as follows:

$$\varphi_1 := \forall x \left(\Big(\bigvee_{P \in \mathcal{T}} P(x) \Big) \wedge \Big(\bigwedge_{P \neq Q \in \mathcal{T}} (\neg P(x) \vee \neg Q(x)) \Big) \right)$$
$$\varphi_2 := \forall x \, (\mathrm{Lit}(x) \rightarrow \exists y \, (E(x,y) \wedge \mathrm{Var}(y)) \wedge \exists y \, (E(x,y) \wedge \mathrm{Cl}(y)))$$
$$\varphi_3 := \forall x \, (\mathrm{Ex}(x) \rightarrow \mathrm{T}(x) \vee \mathrm{F}(x))$$
$$\varphi_4 := \exists x \, (\mathrm{Cl}(x) \wedge \exists^{\geq 3} y \, (E(x,y) \wedge \mathrm{SatLit}(y))).$$

Intuitively, the sentences express the following: φ_1 ensures that each node has exactly one of the structure attributes in $\mathcal{T}$. φ_2 ensures that each literal-node is both connected to a variable-node and to a clause-node. φ_3 ensures that each existential node must satisfy T or F, and thereby enforces that each sufficient reason assigns a truth value to each existential variable. Finally, φ_4 ensures that there exists one clause-node that is connected to three satisfied literals, i.e., there is one satisfied clause in the DNF.

Let $S_0 = (E, \lambda_0)$ be the partial instance that fully specifies all attributes in $\mathcal{T}$ at every vertex, including their zero values. The remaining attributes T, F, and B are unspecified. Every sufficient reason must contain all specifications of λ_0, as leaving any value for an attribute in $\mathcal{T}$ unspecified allows a completion that contradicts φ_1. Analogously, every sufficient reason must specify the entire edge set E; otherwise, we would have a completion contradicting φ_2. Moreover, φ_3 requires every sufficient reason to specify at least one of T, F to 1 at each existential vertex. Consequently, every sufficient reason has size at least $k := |S_0| + |X|$, and one of size k consists exactly of S_0 and one such choice per existential variable. In particular, a sufficient reason of size k fixes no B-values. We will now show that there is a sufficient reason of size k iff Ψ is true.

Suppose that S is a sufficient reason of size k. Its choices for T and F define an assignment α to X. For any assignment β to Y, consider the completion $\bar{G}$ of S that sets all unspecified T/F values to 0, the B attribute of universal nodes according to β, and the B attribute of all other nodes to 0. Since S is sufficient, $\bar{G} \models \varphi$. In this completion, SatLit holds exactly at the literal vertices whose literals are true under (α, β). As each clause vertex has exactly three literal neighbours, φ_4 implies $\psi(\alpha, \beta)$. Since β was arbitrary, Ψ is true.

For the other direction, suppose that Ψ is true with assignment α to X. Extend S_0 by fixing T to 1 for all existential vertices v with $\alpha(v) = 1$ and fixing F to 1 for all existential vertices with $\alpha(v) = 0$. The resulting partial instance S has size k. Let $\bar{G}$ be any completion of S relative to G. By construction of S_0, $\bar{G} \models \varphi_1 \wedge \varphi_2 \wedge \varphi_3$. Let β be the Y-assignment induced by $\bar{G}$. Since $\psi(\alpha, \beta)$ is true, one clause is satisfied. That is, the three contained literals are true under assignment α, β. Their corresponding vertices in $\bar{G}$ satisfy SatLit. T, F attributes on non-existential vertices and B attributes on non-universal nodes do not influence the truth of φ. Additional true values of T and F can only make more literal nodes satisfy SatLit. Thus $\bar{G} \models \varphi_4$. Since $\bar{G}$ was an arbitrary completion, this proves that S is a sufficient reason of size k.

We have shown that Ψ is true iff G_Ψ has a sufficient reason of size k. The construction is polynomial and $\mathcal{M}_\varphi$ is fixed, establishing Σ_2^{P}-hardness in data complexity. □

Theorem 4. *FindMinimalSufficientReason is $\mathsf{FP}^{\mathsf{NP}}$-complete in data complexity.*

Proof. For membership, fix a GNN $\mathcal{M}$. Given a graph $G = (V, E, \lambda)$, we initialise $S = (E, \lambda)$. Then, we iteratively loop over the edges and attribute values and use a $\mathrm{CSR}_\mathcal{M}$ oracle to check whether they can be removed/replaced with $\perp$, respectively. Since $\mathrm{CSR}_\mathcal{M}$ is in coNP, we can do this by querying the complement oracle and then inverting the answer. Finally, we return the obtained partial instance S^*. The algorithm preserves sufficiency at every iteration, so S^* is sufficient. Moreover, once removing an entry breaks sufficiency, removing further entries cannot restore sufficiency. Thus, S^* is inclusion-minimal. The algorithm runs in polynomial time. Thus, $\mathrm{FMSR}_\mathcal{M} \in \mathsf{FP}^{\mathsf{NP}}$.

For hardness, we show that we can simulate any NP oracle using the answer of an FMSR oracle. More specifically, we reduce from Non-4-Colourability similar to Theorem 2. The added difficulty is that the reduction must work for an arbitrary minimal sufficient reason returned by the FMSR oracle.

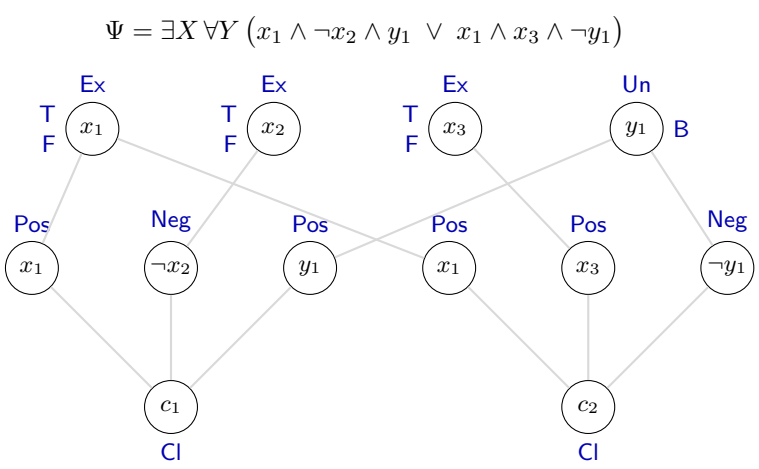

Figure 2: Illustration of the graph encoding used in the Σ_2^{P}-hardness proof for minimum sufficient reasons in Theorem 3. Vertices represent variables, literal occurrences, and clauses. Each literal vertex is connected to its corresponding variable and clause vertices. Blue labels indicate node attributes set to 1 in the constructed graph. The auxiliary clause $z \wedge z \wedge z$ is omitted for readability.

Let $H = (V_H, E_H)$ be a graph for which we want to decide Non-4-Colourability. Without loss of generality, we assume that E_H is not empty. We define the graph $G = (V, E, \lambda_G)$ with node set $V = V_H \cup V_E$, where $V_E = \{v_e : e \in E_H\}$ contains one vertex for each edge in E_H. Each edge-vertex $v_e \in V_E$ is connected to the two endpoints of the edge e it represents, i.e., $E = \{\{u, v_e\}, \{v_e, w\} : e = uw \in E_H\}$. We use four attributes $(\mathrm{Node}, \mathrm{Edge}, P_1, P_2)$ to encode the instance structure and node colours. On G, we set $\lambda_G(v) = (1, 0, 0, 0)$ for all $v \in V_H$ and $\lambda_G(v) = (0, 1, 0, 0)$ for all $v \in V_E$. Thus, the node attributes of G mark the node- and edge vertices, and assign the same colour to all vertices. See Figure 3 for an illustration.

We construct the fixed GMLC sentence $\varphi := \varphi_1 \wedge \varphi_2 \wedge \varphi_3$ with

$$\varphi_1 = \forall x \left((\mathrm{Node}(x) \vee \mathrm{Edge}(x)) \wedge \neg(\mathrm{Node}(x) \wedge \mathrm{Edge}(x))\right)$$
$$\varphi_2 = \forall x \left(\mathrm{Edge}(x) \to \exists^{\geq 2} y\, E(x, y)\right)$$
$$\varphi_3 = \exists x \left(\mathrm{Edge}(x) \wedge \bigvee_{i=1}^{4} \exists^{\geq 2} y \left(E(x, y) \wedge c_i(y)\right)\right),$$

where c_i is defined as in Equation (1). Since we assumed $E \neq \emptyset$, we have $\mathcal{M}_\varphi(G) = 1$.

Let $S_0 = (E, \lambda_0)$ with $\lambda_0(v) = (1, 0, \bot, \bot)$ for all $v \in V_H$ and $\lambda_0(v) = (0, 1, \bot, \bot)$ for all $v \in V_E$. Thus, S_0 fully specifies the edges and Node/Edge attributes while leaving only the colour attributes unspecified. This implies that the completions of S_0 correspond precisely to all assignments of four colours to V. We have $\bar{G} \models \varphi_1 \wedge \varphi_2$ for all completions $\bar{G}$ as all completions match the graph structure and Node/Edge attributes of G. The colours of edge-vertices do not influence the evaluation of φ_3. Hence, $\bar{G} \models \varphi_3$ for all completions $\bar{G}$ iff each assignment of four colours creates nodes $(u, v_e, w) \in V_H \times V_E \times V_H$ with $e = uw \in E_H$ and $c_i(u) \leftrightarrow c_i(w)$ for all i. This shows that S_0 is sufficient if and only if H is not 4-colourable.

Each sufficient reason for G must completely specify the Node and Edge attributes, since otherwise, there would be a completion that contradicts φ_1. Moreover, each sufficient reason must contain all edges, because otherwise, there would be a completion with an edge-vertex of degree < 2, contradicting φ_2. This shows that S_0 is the *unique* minimal sufficient reason iff H is not 4-colourable. Hence, we can use an $\mathrm{FMSR}_{\mathcal{M}_\varphi}$ oracle to decide Non-4-Colourability as follows: For an arbitrary valid oracle answer S^*, return that H is not 4-colourable iff $S^* = S_0$. By flipping the answer, we can also decide 4-Colourability. Hence, the FMSR oracle can simulate an NP oracle and therefore every computation in $\mathsf{FP}^{\mathsf{NP}}$, establishing hardness. □

Theorem 5. *CountCompletions is #P-complete in data complexity.*

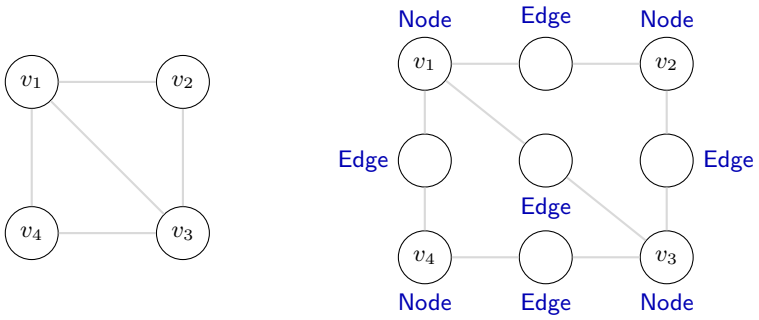

Figure 3: Illustration of the structure for the proof of Theorem 4. Left: an input graph H. Right: the incidence-style subdivision of H used in the reduction. Original vertices become nodes labelled by the attribute Node, original edges become nodes labelled by the attribute Edge, and adjacency in the right-hand graph encodes incidence in H.

Proof. For membership, fix a GNN $\mathcal{M}$. Given a graph G and a partial instance $S \preceq G$, nondeterministically guess a completion $\bar{G}$ of S relative to G. If $\mathcal{M}(\bar{G}) = \mathcal{M}(G)$, accept it. The number of accepting computations equals the number of label-preserving completions.

For hardness, we reduce from #IndependentSet similar to Kula et al. (2026). Let $H = (V, E)$ be the graph for which we want to count the number of independent sets. Define the graph $G = (V, E, \lambda)$ with $\lambda(v) = 0$ for all $v \in V$, and construct the GMLC sentence

$$\varphi = \neg\exists x \left(P(x) \wedge \exists y \left(E(x, y) \wedge P(y)\right)\right).$$

Let $S = (E, \lambda_S)$ be the partial instance that specifies all edges but no attribute value. The completions $\bar{G} = (V, E, \lambda_{\bar{G}})$ of S with $\mathcal{M}_\varphi(\bar{G}) = 1$ correspond precisely to independent sets $X = \{v \in V : \lambda_{\bar{G}}(v) = 1\}$. Hence, the result of $\mathrm{CC}_{\mathcal{M}_\varphi}(G, S)$ equals the number of independent sets in H.

The construction is polynomial and $\mathcal{M}_\varphi$ is fixed, establishing #P-hardness in data complexity. □

Theorem 6. *MinimumChangeRequired is* NP*-complete in data complexity.*

Proof. For membership, fix a GNN $\mathcal{M}$. If it is possible to flip the decision of $\mathcal{M}$ on graph G with at most k changes, this is witnessed by a graph G'. Since we can verify both the distance between G and G' and $\mathcal{M}(G') \neq \mathcal{M}(G)$ in polynomial time, G' serves as a witness. Thus, $\mathrm{MCR}_{\mathcal{M}} \in$ NP.

For hardness, we reduce from VertexCover. Let $H = (V, E)$ be a graph and $k \in \mathbb{N}$. We may assume without loss of generality that E is not empty since otherwise, there is a trivial empty vertex cover. Let $G = (V, E, \lambda_G)$ with $\lambda_G(v) = 0$ for all $v \in V$. We construct the fixed GMLC sentence

$$\varphi := \neg\exists x \left(\neg P(x) \wedge \exists y\big(E(x, y) \wedge \neg P(y)\big)\right).$$

We know that $\mathcal{M}_\varphi(G) = 0$ because of the assumption that E is not empty. For each graph $G' = (V, E', \lambda')$ it holds $G' \models \varphi$ iff the set $X(G') = \{v \in V : \lambda'(v) = 1\}$ is a vertex cover in G'.

Suppose that there exists a graph $G' = (V, E', \lambda')$ with $E' \subseteq E$ and $\mathcal{M}_\varphi(G') = 1$ and $d(G, G') \leq k$. Each removed edge can be covered by flipping the node attribute of one adjacent node to 1. Thus, we obtain a vertex cover X' of H with $|X'| \leq k$ by selecting each node v with $\lambda'(v) = 1$ and one arbitrary endpoint of each $e \in E \setminus E'$.

Conversely, suppose that H has a vertex cover $X \subseteq V$ of size at most k. The graph $G' = (V, E, \lambda')$ with $\lambda'(v) = 1$ iff $v \in X$ satisfies $G' \models \varphi$. Since $\mathcal{M}_\varphi(G') = 1$ and $d(G, G') \leq k$, this shows that we can flip the decision of $\mathcal{M}_\varphi$ on G with at most k changes.

The construction is polynomial and $\mathcal{M}_\varphi$ is fixed, establishing NP-hardness in data complexity. □

C FPT Results for Bounded Treewidth

In this section, we formulate all explainability queries in monadic second-order logic (MSO). This allows us to apply different versions of Courcelle's theorem (Courcelle, 1990), proving that they are FPT parameterised by $\mathrm{tw}(G)$.

Theorem 7. *For every fixed bounded AC-GNN with final bounded readout, all five explainability queries are FPT parameterised by* $\mathrm{tw}(G)$.

Proof. Fix a GMLC sentence φ with predicate symbols $E, P_1, \ldots, P_m$. Let $G = (V, E, (P_1)_G, \ldots, (P_m)_G)$ be the input graph with $n = |V|$ vertices. We use standard MSO_2 with vertex-set variables ranging over V and edge-set variables ranging over E.

A tuple $X = (X_E, X_1, \ldots, X_m)$ with edge-set variable X_E and vertex-set variables X_i represents the graph $G_X := (V, X_E, X_1, \ldots, X_m)$. Conversely, every graph H with node set $V(H) = V$ and edge set $E(H) \subseteq E$ is represented by a variable assignment to X. Thus, we have a bijection between such graphs and variable assignments to X. Similarly, a tuple $Y = (Y_E, Y_1, \ldots, Y_m)$ with edge-set variable Y_E and vertex-set variables Y_i represents a partial instance $S_Y = (Y_E, \lambda_Y)$ specifying the edges in Y_E and fixing the value of $P_i(v)$ to its value in G whenever $v \in Y_i$. More formally,

$$\lambda_i^Y(v) = \begin{cases} \bot & \text{if } v \notin Y_i \\ 0 & \text{if } v \in Y_i, G \not\models P_i(v) \\ 1 & \text{if } v \in Y_i, G \models P_i(v) \end{cases}.$$

Conversely, we denote by $Y_S = (Y_E, Y_1, \ldots, Y_m)$ the variable assignment corresponding to the partial instance S. More formally, $Y_E = E(S)$ and $Y_i = \{v \in V : \lambda_i^S(v) \neq \bot\}$. As for graphs, we have a bijection between partial instances $S \preceq G$ and variable assignments to Y. We define the size of Y as $w(Y) := |Y_E| + \sum_{i=1}^m |Y_i|$.

By representing graphs and partial instances in MSO_2, we are now able to quantify over them. In order to evaluate whether G_X models φ, we need an MSO_2 formula $\hat{\varphi}$ such that for all variable assignments to X and all nodes v, $(G, X, v) \models \hat{\varphi}$ iff $(G_X, v) \models \varphi$. We define the translation $TR(\varphi)$ that replaces $P_i(v)$ with $v \in X_i$ and $E(u, v)$ with $\{u, v\} \in X_E$. All variables from φ range over V. We show by induction on the formula that this translation is correct. For atomic formulas, we have

$$(G, X, v) \models TR(P_i(v)) \Leftrightarrow (G, X, v) \models (v \in X_i) \Leftrightarrow (G_X, v) \models P_i(v).$$

Assume for the induction that $(G, X, v) \models TR(\varphi)$ iff $(G_X, v) \models \varphi$ for all assignments to X and all nodes v. For logical connectives $\wedge$ and $\neg$, equivalence follows immediately. For guarded quantification, we obtain

$$\begin{aligned}(G, X, v) \models TR(\exists^{\geq p} y\,(E(v, y) \wedge \varphi(y))) &\Leftrightarrow (G, X, v) \models \exists^{\geq p} y\,(\{v, y\} \in X_E \wedge TR(\varphi(y))) \\ &\Leftrightarrow (G_X, v) \models \exists^{\geq p} y\,(E(v, y) \wedge \varphi(y)).\end{aligned}$$

For global counting quantification, the induction hypothesis ensures that the same vertices satisfy the translated subformula in G and the original subformula in G_X. Hence, the equivalence is preserved. We define $\hat{\varphi}(X) := TR(\varphi)$. Since φ has no free variables, we get $(G, X) \models \hat{\varphi}(X)$ iff $G_X \models \varphi$.

Define

$$\mathrm{Comp}(X, Y) := Y_E \subseteq X_E \wedge \bigwedge_{i=1}^m \big(X_i \cap Y_i = P_i \cap Y_i\big).$$

Recall that for all set variables X_i it holds that

$$X_i = \{v \in V : G_X \models P_i(v)\} = \{v \in V : \lambda_i^{G_X}(v) = 1\}$$

and, since P_i is a predicate symbol,

$$P_i = \{v \in V : G \models P_i(v)\} = \{v \in V : \lambda_i^G(v) = 1\}.$$

Thus, $X_i \cap Y_i = P_i \cap Y_i$ if and only if $\lambda_i^{G_X}(v) = \lambda_i^G(v)$ for all v with $\lambda_i^{S_Y}(v) \neq \bot$. Further, $G \models Y_E \subseteq X_E$ iff $E(S_Y) \subseteq E(G_X) \subseteq E$. Thus $G \models \mathrm{Comp}(X, Y)$ if and only if G_X is a completion of S_Y relative to G.

We encode sufficiency with the following formula:

$$\mathrm{SR}_\varphi(Y) := \forall X \left(\mathrm{Comp}(X, Y) \to (\varphi \leftrightarrow \hat{\varphi}(X))\right),$$

where $\forall X$ quantifies all components of X. By the bijection between variable assignments and graphs, this analogously quantifies over all attributed graphs H with node set $V(H) = V$ and edge set $E(H) \subseteq E$. Hence, $G \models \mathrm{SR}_\varphi(Y)$ iff S_Y is a sufficient reason for φ.

Fix a GNN $\mathcal{M}$ and an equivalent GMLC sentence φ. Let G be the input graph. Then, SR_φ and $\hat{\varphi}$ are fixed MSO_2 formulas whose size depends only on $|\varphi|$. It directly follows that we can *check sufficiency* of a given partial instance $S \preceq G$ by checking $G \models \mathrm{SR}_\varphi(Y_S)$. By Cygan et al. (2015, Theorem 7.11), we can evaluate $G \models \mathrm{SR}_\varphi(Y_S)$ in time $f(|\varphi|, \mathrm{tw}(G))n$ for some computable function f. Hence, $\mathrm{CSR}_\mathcal{M}$ is FPT parameterised by $\mathrm{tw}(G)$. To determine the size of a *minimum sufficient reason*, we can equivalently minimise $w(Y)$ subject to $G \models \mathrm{SR}_\varphi(Y)$. Since the weight function is linear, by the optimisation version of Courcelle's theorem (Arnborg et al., 1991, see Cygan et al., 2015, Theorem 7.12), we can find this minimum in time $f(|\varphi|, \mathrm{tw}(G))n$ for some computable function f. Hence, $\mathrm{MSR}_\mathcal{M}$ is FPT parameterised by $\mathrm{tw}(G)$. To *find minimal sufficient reasons*, we use the greedy deletion algorithm from Theorem 4. Since the $\mathrm{CSR}_\mathcal{M}$ oracle runs in FPT time and the algorithm needs at most $|E| + m|V|$ iterations, this shows that $\mathrm{FMSR}_\mathcal{M}$ is FPT parameterised by $\mathrm{tw}(G)$.

To *count prediction-preserving completions* of S, we expand G with unary predicates F_E, $F_1, \ldots, F_m$. Here, F_E marks the edges in E_S, and F_i marks the vertices whose i-th attribute is specified by S. We denote the expanded structure by G^S, and write $F = (F_E, F_1, \ldots, F_m)$. We define the fixed MSO_2 formula

$$\Phi(X) := \mathrm{Comp}(X, F) \wedge \left(\varphi \leftrightarrow \hat{\varphi}(X)\right),$$

where $\mathrm{Comp}(X, F)$ is obtained from $\mathrm{Comp}(X, Y)$ by replacing the variables in Y with the corresponding predicates in F. By the bijection $X \leftrightarrow G_X$, the assignments to X satisfying $\Phi(X)$ in G^S correspond precisely to the prediction-preserving completions of S relative to G. The predicates in F are part of the input structure, so only assignments to X are counted. Adding unary predicates does not change treewidth. Hence, the MSO_2 counting theorem of Courcelle et al. (2001, Theorem 32) computes the desired count in time $f(|\varphi|, \mathrm{tw}(G)) \cdot n^{O(1)}$ for some computable function f. Since φ is fixed, $\mathrm{CC}_\mathcal{M}$ is FPT parameterised by $\mathrm{tw}(G)$.

To determine the *minimum change required* to flip the GNN prediction, we use the analogy between non-sufficient partial instances and counterfactuals. Let $N := |E| + m|V|$ be the number of elements in G that can be specified. By definition, each non-sufficient partial instance of size k has a counterfactual at distance at most $N - k$ as a completion. Conversely, every counterfactual at distance k induces a non-sufficient partial instance of size $N - k$ that fixes everything except the k changes. Thus, we can minimise the change required to flip the GNN prediction by maximising the size of a non-sufficient partial instance. If no assignment to Y satisfies $\neg\, \mathrm{SR}_\varphi(Y)$, no counterfactual exists. Otherwise, compute

$$K := \max\{w(Y) : G \models \neg\, \mathrm{SR}_\varphi(Y)\}.$$

As argued before, we can compute K in time $f(|\varphi|, \mathrm{tw}(G))n$ for some computable function f. Then, the minimum number of changes required to flip the prediction is $N - K$. Hence, $\mathrm{MCR}_\mathcal{M}$ is FPT parameterised by $\mathrm{tw}(G)$. □