SUFFICIENT REASONS FOR EXPLAINING GNNS

Anonymous authors
Paper under double-blind review

ABSTRACT

Explanations of graph neural network (GNN) predictions are almost universally subgraphs of the input (or soft masks over it), judged by fidelity, asking whether keeping or removing the subgraph preserves or changes the prediction. Yet the real reason behind a GNN prediction routinely involves more than what is present—a missing edge, a negative condition on every neighbor, the absence of a motif. No subgraph can express such non-monotone reasons, though one may still pass every fidelity test. We propose *sufficient reasons* for GNNs: inclusion-minimal sets of (positive and negative) facts about the input such that every graph of the same size satisfying them receives the same prediction. The explanation is thus a *certificate* rather than a score. Analyzing GNN decision classes in the counting logic C_2, we show that they can be stratified by graph size, which fixes the context in which explanations are valid. Since checking every graph consistent with a candidate explanation is generally infeasible, we give a randomized algorithm that, with PAC guarantees, returns an ε-sufficient reason from which no fact can be removed. In controlled synthetic experiments with GNN-expressible target properties, our method finds an ε-sufficient explanation for every instance, while existing subgraph explainers do so only for a minority of instances, seldom for non-monotone properties, and with larger explanations whenever they succeed.

1 INTRODUCTION

Graph Neural Networks (GNNs) are increasingly used to make predictions on structured data, and a large body of work (post-hoc) explains the prediction of a trained GNN f on a given input graph $\mathcal{G}$ by: gradient (Pope et al., 2019) or relevance attributions (Schnake et al., 2022), learned masks (Ying et al., 2019; Luo et al., 2020; Schlichtkrull et al., 2021), subgraph search (Yuan et al., 2021), or counterfactuals (Lucic et al., 2022); see Yuan et al. (2023) for a survey. Despite their different mechanisms, these methods seek a subgraph $\mathcal{H} \subseteq \mathcal{G}$ (or a soft mask over nodes and/or edges of $\mathcal{G}$) that is considered important for the prediction $f(\mathcal{G})$, and they are evaluated by the behavior of the model under perturbations: *Fidelity*$-$ measures whether $\mathcal{H}$ alone preserves the prediction, while *Fidelity*$+$ measures whether removing $\mathcal{H}$ changes it (Pope et al., 2019; Yuan et al., 2023). These are useful measures of dependence, but they do not establish that $\mathcal{H}$ is the true reason for the decision.

The problem becomes apparent for *non-monotone* properties. Consider a GNN classifier f recognizing whether there is a vertex with "*no green neighbors and at least one blue neighbor*", and the graph $\mathcal{G}$ in Fig. 1(left): the green vertex a is a witness here, thus $f(\mathcal{G}) = 1$. Now consider the subgraph explanation $\mathcal{H}$ induced by the two bottom vertices. Keeping only $\mathcal{H}$ preserves the prediction, since c now has a blue neighbor and no green one, while removing $\mathcal{H}$ eliminates the only blue vertex and flips the prediction to 0. Thus, both Fidelity$-$ and Fidelity$+$ indicate that $\mathcal{H}$ is important, yet $\mathcal{H}$ does not even contain the witness a—its positive prediction arises only because restricting the graph removed exactly the information the property depends on. Note that testing against *all* subgraphs of $\mathcal{G}$ containing the explanation, as proposed in some recent works (Azzolin et al., 2025b; 2026), does not help either: the subgraph $\mathcal{H}'$ in Fig. 1(right) passes the test, since a stays a witness whatever part of $\mathcal{G}$ is added back, yet a graph in which c becomes (additionally) green and adjacent to a satisfies $\mathcal{H}'$ but is classified as 0. Both $\mathcal{H}$ and $\mathcal{H}'$ fail for the same reason: the prediction depends on what $\mathcal{G}$ does *not* contain, which no subgraph can express.

Such non-monotone properties are not a corner case: the one above is an ordinary C_2 sentence —the logic that characterizes the expressiveness of message-passing GNNs (Barceló et al., 2020; Grohe, 2021)— hence exactly the kind of property a standard GNN represents. Conditions on all neighbors,

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.



Figure 1: (left) The subgraph $\mathcal{H}$ of $\mathcal{G}$ passes both fidelity tests, although it does not even contain the witness a. (right) The subgraph $\mathcal{H}'$ passes even the test against all subgraphs of $\mathcal{G}$ containing it, yet its completions (dashed) include graphs classified 0. Thus, neither determines the prediction.

upper bounds on counts, or even just negative predictions of monotone properties all depend on absent structures and arise in practice, e.g., in chemical reactions (Weinbauer et al., 2025).

The example calls for a stronger notion of explanation. Rather than asking whether some perturbation preserves or changes the prediction, we ask whether the identified information is sufficient to *determine* it: informally, a partial description of a graph is a *sufficient reason* if every graph consistent with that description receives the same prediction. The explanation is thus a *certificate* for the prediction rather than a score, valid under arbitrary changes to whatever it does not specify.

We note that for unrestricted graph extensions, this notion is too strong: even the full graph $\mathcal{G}$ need not determine the prediction if new vertices can be added (e.g., a new green neighbor of a flips it). This is no accident: a GNN may indeed treat graphs of different sizes (completely) differently, as we prove in Section 4, so in general an explanation is a statement about graphs of a given size. We thus define sufficient reasons over all graphs with the same number of vertices as the input $\mathcal{G}$.

Concretely, we represent such a partial description by a set of ground facts, fixing some vertex labels and pairwise relations while leaving the rest open. A sufficient reason is such a set whose completions all receive the same prediction, and it is minimal if no fact can be dropped. Our sufficient reasons can be seen as the graph counterpart of abductive explanations from formal explainable AI (Shih et al., 2018; Ignatiev et al., 2019; Darwiche, 2023), so far studied for fixed-dimensional inputs, or for graphs only in the restricted form illustrated by $\mathcal{H}'$ (Azzolin et al., 2025b, Fig. 1, right).

The known correspondence between GNNs and C_2 (Xu et al., 2019; Morris et al., 2019; Grohe, 2021) then structures the explanations themselves: since a GNN cannot distinguish C_2-equivalent graphs, each prediction class is a union of C_2-equivalence classes, and over graphs of a fixed size we give it an explicit normal form (Section 4). Conversely, a sufficient reason may fix instance-specific facts that C_2 cannot see, so we further associate with each reason a canonical C_2 abstraction—the part of the explanation visible to a GNN model.

To compute a sufficient reason, we start from the complete description of the input graph and remove facts while testing whether the resulting partial specification remains sufficient. Since exhaustive verification over all completions is generally infeasible, we develop a randomized procedure that samples the completions and returns a description from which no fact can be removed, with an (ε, δ) PAC guarantee on its sufficiency (Section 5).

Our contributions are as follows: (1) We introduce *sufficient reasons for GNNs*—a formal notion of GNN explanation based on invariance of the prediction over all same-size graph completions, avoiding the limitations of existing methods in the presence of non-monotone properties (Section 3). (2) We connect this notion to the logical characterization of message-passing GNNs by C_2 logic, showing their decision classes are stratified by graph size, admit a normal form over graphs of fixed size, and every sufficient reason has a canonical C_2 abstraction (Section 4). (3) We propose a randomized algorithm that returns a description from which no fact can be removed, with (ε,δ) PAC guarantees on its sufficiency (Section 5). (4) In controlled experiments over diverse C_2 target properties, our method finds an ε-sufficient reason for every instance, often at sizes matching the optimum, whereas the existing methods produce generally insufficient and larger explanations. (Section 6).

2 Preliminaries

Even though our results generalize to arbitrary binary relational structures, we focus on vertex-colored, simple, undirected graphs for clarity of presentation. We use first-order logic (FO) and its two-variable counting fragment C_2 to describe structural properties of graphs (Otto, 2017). Our

structures are thus finite, simple, undirected graphs with a finite collection of unary predicates. Equality is available in the language. Let $\mathcal{L} = \{E, P_1, \ldots, P_r, =\}$ be such a vocabulary, where E is a binary relation representing the edge relation and each P_i is a unary relation. A graph structure is a finite structure $\mathcal{G} = (V, E, P_1, \ldots, P_r)$, where $E \subseteq V \times V$ is symmetric and irreflexive. The unary predicates can be seen as nodes' $\{0,1\}$-valued feature vectors of dimension r. The size of a graph is the number of vertices $n = |V|$. We denote by Graphs the set of all vertex-colored graphs and Graphs_n those having size n. The set of all neighbors of a node $a \in V$ is denoted $\mathcal{N}(a)$.

FO formulas are built from atomic formulas using Boolean connectives and existential and universal quantification. The atomic formulas of our language are $E(x,y)$, $P_i(x)$, $x = y$, where x, y are variables. A formula with no free variables is called a *sentence*. The semantics of FO formulas is defined as usual. For example, a graph $\mathcal{G}$ satisfies the sentence $\exists x \forall y\,(E(x,y) \rightarrow P_1(y))$ iff there exists a vertex a such that for all its neighbors b, we have $b \in P_1$. We write $\mathcal{G} \models \varphi$ when a sentence φ is true in $\mathcal{G}$. A *literal* is an atomic formula or its negation. Examples of literals are $\neg E(x,y)$, $P_i(x)$, $x \neq y$. When the variables are replaced by vertices of a particular graph, we obtain *ground literals*. A positive ground atom such as $E(a,b)$ or $P_i(a)$ is a *ground fact* about the graph. Negative ground literals such as $\neg E(a,b)$ or $\neg P_i(a)$ specify facts that do not hold. For a fixed vertex set V, every graph structure determines a complete set of ground literals: $\mathsf{Lit}(\mathcal{G}) = \{\ell : \ell \text{ is a ground literal and } \mathcal{G} \models \ell\}$. For every ground atom α, exactly one of α and $\neg\alpha$ belongs to $\mathsf{Lit}(\mathcal{G})$. Hence $\mathsf{Lit}(\mathcal{G})$ is a maximal consistent set of ground literals, uniquely determining the structure on V. Thus, for a fixed vertex set, we may view a graph $\mathcal{G}$ either as a structure or as the set of literals $\mathsf{Lit}(\mathcal{G})$. Given a set of literals $\mathbb{S}$ over a vertex set V, we call a graph $\mathcal{G}$ over V a *completion* of $\mathbb{S}$ if $\mathbb{S} \subseteq \mathsf{Lit}(\mathcal{G})$. The set of all completions of $\mathbb{S}$ is denoted $\mathsf{Comp}(\mathbb{S})$.

The logic C_2 extends FO with counting quantifiers while restricting formulas to at most two variables. The basic counting quantifier $\exists^{\geq k} x\, \varphi(x)$ states that there are at least k distinct vertices x for which $\varphi(x)$ holds. Thus, for example, $\exists^{\geq 3} y\, E(x,y)$ states that x has at least three neighbors. We also use the abbreviation $\exists^{=k} x\, \varphi(x)$ for $\exists^{\geq k} x\, \varphi(x) \wedge \neg\exists^{\geq k+1} x\, \varphi(x)$, which states that exactly k vertices satisfy φ. Note that C_2 can express the size of the graph by the sentence $\mathsf{Size}_n = \exists^{=n} x\,(x = x)$, which states that there are exactly n vertices. We write $\mathcal{G} \equiv_{\mathrm{C}_2} \mathcal{H}$ when $\mathcal{G}$ and $\mathcal{H}$ satisfy exactly the same C_2 sentences. The equivalence class w.r.t. $\equiv_{\mathrm{C}_2}$ is called a *C_2-equivalence class*. A C_2 sentence is called *primitive* if it is built only from counting quantifiers, conjunctions and literals (thus the negations are allowed only next to the atomic formulas).

2.1 Graph Neural Networks

We focus on widely used message-passing GNNs applied as graph classifiers, i.e. functions f that map a graph $\mathcal{G}$ to a prediction $f(\mathcal{G})$ by iterated (multi-layer) neighborhood aggregations and updates, followed by a readout (Wu et al., 2021). Let $\boldsymbol{v}_a^{(l)}$ denote the feature vector of node a at layer l. The l-th layer of a message-passing GNN is defined by $\boldsymbol{v}_a^{(l+1)} = \mathsf{comb}(\boldsymbol{v}_a^{(l)}, \mathsf{agg}(\{\!\{\boldsymbol{v}_b^{(l)} \mid b \in \mathcal{N}(a)\}\!\}))$, where comb is a combination MLP and agg is an aggregation function (e.g., sum, mean, or max) that is applied to the multiset of neighbor features. The final graph representation is obtained by applying an MLP ro to the aggregated feature vectors of all nodes from the last layer L: $f(\mathcal{G}) = \mathsf{ro}(\mathsf{agg}(\{\!\{\boldsymbol{v}_a^{(L)} \mid a \in V\}\!\}))$. In the following theorem, the equivalence between GNNs and the 1-WL algorithm was established by Xu et al. (2019); Morris et al. (2019). The equivalence between 1-WL and C_2 logic was established by Cai et al. (1992); for an overview, see Grohe (2021).

Theorem 2.1. *Let f be a GNN and $\mathcal{G}, \mathcal{H}$ two graphs. If $\mathcal{G} \equiv_{C_2} \mathcal{H}$, then $f(\mathcal{G}) = f(\mathcal{H})$. Conversely, for $n \in \mathbb{N}$, there is a GNN f_n such that $f_n(\mathcal{G}) = f_n(\mathcal{H})$ implies $\mathcal{G} \equiv_{C_2} \mathcal{H}$ for any $\mathcal{G}, \mathcal{H} \in \mathsf{Graphs}_n$.*

3 Sufficient Reasons

Let f be a GNN and $\mathcal{G}$ an input graph. As indicated in the introduction, we want an explanation for $f(\mathcal{G})$ that forms a *sufficient reason* for the same prediction: a partial specification of $\mathcal{G}$, built from literals already present in $\mathcal{G}$, whose every completion preserves the prediction. Note that $\mathsf{Lit}(\mathcal{G})$ itself is always such a specification, since its only completion is $\mathcal{G}$ itself; the question is how much of this information can be dropped while still fixing the prediction. Answering it requires fixing a context C—a class of graphs over which the completions are quantified, and hence in which the resulting certificate is valid. This context cannot be the class of all graphs, since $\mathcal{G}$ cannot know

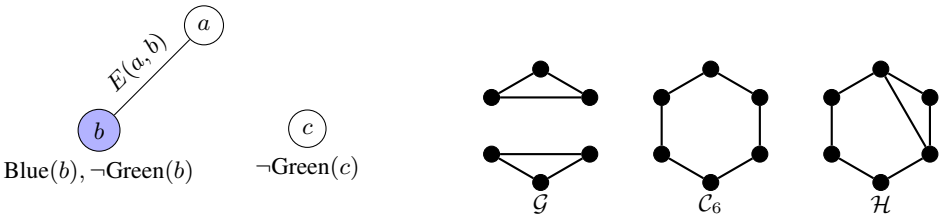


Figure 2: (left) A minimal sufficient reason for $f(\mathcal{G})$. (right) Graphs $\mathcal{G} = \mathcal{C}_3 \sqcup \mathcal{C}_3$, $\mathcal{C}_6$, and $\mathcal{H}$: extended $\mathcal{C}_6$ with a chord.

how f behaves on its extensions. For instance, if f recognizes whether all nodes are red, no size-n graph with all red nodes can guarantee that a larger graph with new nodes remains all red. Thus, we restrict $\mathsf{C} = \mathsf{Graphs}_n$, the class of graphs of the same size n as $\mathcal{G}$. One could instead take $\mathsf{C} = \bigcup_{i=1}^{n} \mathsf{Graphs}_i$, but this would considerably enlarge the space of graphs to search over for an explanation. A sufficient reason is thus a partial specification of $\mathcal{G}$ such that every graph in C consistent with it lies in the same decision region as $\mathcal{G}$. Unlike a subgraph explanation, $\mathbb{S}$ may contain both positive and negative literals—a feature essential for capturing non-monotone predictions.

Definition 3.1. Let $\mathcal{G}$ be a graph of size n and f a GNN. A subset of literals $\mathbb{S} \subseteq \mathsf{Lit}(\mathcal{G})$ is said to be a *sufficient reason* for $f(\mathcal{G})$ if $f(\mathcal{H}) = f(\mathcal{G})$ for any completion $\mathcal{H} \in \mathsf{Comp}(\mathbb{S})$. We call a sufficient reason $\mathbb{S}$ *inclusion-minimal* (shortly minimal) if no proper subset of $\mathbb{S}$ is sufficient. Thus, minimality is with respect to set inclusion, rather than cardinality.

This definition transfers existing sufficient reasons (Darwiche & Hirth, 2023; Darwiche, 2023), also known as prime-implicants (Shih et al., 2018), from assignments over a fixed set of boolean features to graphs, where completions range over the vertex set of $\mathcal{G}$. [1]

Example 3.2. Consider the graph $\mathcal{G}$ in Fig. 1(left) which satisfies the C_2 sentence $\varphi = \exists x\big(\forall y(E(x,y) \rightarrow \neg\mathrm{Green}(y)) \wedge \exists y(E(x,y) \wedge \mathrm{Blue}(y))\big)$. Suppose that f is a GNN such that $f(\mathcal{H}) = 1$ iff $\mathcal{H} \models \varphi$ and $f(\mathcal{H}) = 0$ otherwise. Note that

$$\begin{aligned}\mathsf{Lit}(\mathcal{G}) = \{&E(a,b), E(b,c), E(a,c), \mathrm{Green}(a), \neg\mathrm{Blue}(a), \neg\mathrm{Red}(a),\\ &\neg\mathrm{Green}(b), \mathrm{Blue}(b), \neg\mathrm{Red}(b), \neg\mathrm{Green}(c), \neg\mathrm{Blue}(c), \mathrm{Red}(c)\}.\end{aligned}$$

A sufficient reason for $f(\mathcal{G})$ is a subset of $\mathsf{Lit}(\mathcal{G})$, not necessarily a connected subgraph. A minimal sufficient reason for $f(\mathcal{G})$ is $\mathbb{S} = \{E(a,b), \mathrm{Blue}(b), \neg\mathrm{Green}(b), \neg\mathrm{Green}(c)\}$; see Fig. 2(left). The sufficient reason $\mathbb{S}$ states that a has a blue neighbor b and it cannot have a green neighbor as b, c are not green. Note that the example property is not monotone and, accordingly, $\mathbb{S}$ contains negative literals, while any subgraph explanation, such as $\mathcal{H}'$ from Fig. 1(right), would be insufficient. The sufficient reason is then minimal, as removing any literal from $\mathbb{S}$ would allow for a completion that changes the prediction of f.

4 What Can a Message-Passing GNN Distinguish?

This section situates sufficient reasons within the broader question of GNN expressiveness. Since message-passing GNNs identify graphs only up to C_2-equivalence, we characterize the resulting decision space; the proof of the following theorem is a standard model-theoretic argument and appears in the Appendix.

Theorem 4.1. *Let f be a GNN classifier and c a value in the range of f. There is a set T_c of C_2 sentences such that $f(\mathcal{G}) = c$ iff $\mathcal{G} \models \varphi$ for all $\varphi \in T_c$.*

The collection of the sets T_c, for c ranging over the outputs of f, gives complete information about the GNN f and can thus be seen as a model-level explanation of f. However, T_c can be infinite: for a classifier that tests whether a graph has more red vertices than green ones, no finite T_c exists, as we show in the Appendix (Lemma A.2). Nevertheless, restricting to graphs of a fixed size recovers finiteness: there is a single C_2 sentence that fully describes the behavior of f on Graphs_n, as the following theorem shows (proof in the Appendix).

[1] not merely over subgraphs, as in Azzolin et al. (2025b)

Theorem 4.2. *Let f be a GNN classifier and c a value in the range of f. For each $n \geq 1$, there is a C_2 sentence φ_n such that $f(\mathcal{H}) = c$ iff $\mathcal{H} \models \varphi_n$ whenever $\mathcal{H} \in \mathsf{Graphs}_n$. Moreover, $f^{-1}(c) \cap \mathsf{Graphs}_n$ is a finite union of C_2-equivalence classes.*

Theorem 4.2 further justifies our choice of context for sufficient reasons: since the GNN's decision space is stratified by the size of the input graph, with φ_n potentially differing from one size to the next, it is natural to require explanations to be sufficient only within the same-size context—a sufficient reason valid at one size need not remain sufficient at another.

The sentence φ_n from Theorem 4.2 completely determines the behavior of f on Graphs_n, and it can moreover be expressed as a disjunction of primitive C_2 sentences—an analog of disjunctive normal form for propositional logic, where every formula is a disjunction of conjunctions of literals. Proofs of the following theorem and its corollary appear in the Appendix.

Theorem 4.3. *Over graphs of fixed size n, every C_2 sentence φ is logically equivalent to a disjunction of primitive C_2 sentences.*

Corollary 4.4. *Let $\mathcal{G}$ be a graph of size n. Its C_2-equivalence class is definable by a C_2 sentence $\mathsf{Size}_n \wedge \varphi_{\mathcal{G}}$ where $\varphi_{\mathcal{G}}$ is a primitive C_2 sentence, i.e., $\mathcal{H} \equiv_{C_2} \mathcal{G}$ iff $\mathcal{H} \models \mathsf{Size}_n \wedge \varphi_{\mathcal{G}}$.*

Over graphs of size n, the decision space thus consists of finitely many C_2-equivalence classes, and by Theorem 2.1 any function on these classes is realizable by some GNN. Consequently, the full model over Graphs_n can always be explained by a disjunction of primitive C_2 sentences—though such an explanation may be very long, and computing it is likely intractable.

Since message-passing GNNs are C_2-invariant, one might ask whether explanations should themselves be expressed as C_2 formulas. In light of Corollary 4.4, a natural candidate would define a sufficient reason for $f(\mathcal{G})$ as the weakest primitive C_2 sentence φ such that $\mathcal{G} \models \varphi$ and $f(\mathcal{H}) = f(\mathcal{G})$ for every graph $\mathcal{H}$ of the same size satisfying φ. Such a definition, however, necessarily forgets distinctions between C_2-equivalent realizations of $\mathcal{G}$—distinctions that Definition 3.1 is free to exploit. The following example shows how the two notions can diverge.

Consider $\mathcal{G} = \mathcal{C}_3 \sqcup \mathcal{C}_3$. Its C_2-equivalence class also contains $\mathcal{C}_6$: C_2 only sees that all six vertices have degree two, so it cannot distinguish $\mathcal{G}$ from $\mathcal{C}_6$ (Fig. 2, right). A sufficient reason, by contrast, can pin down the concrete triangle edges; information invisible at the C_2 level. Suppose $f(\mathcal{G}) = 1$, $f(\mathcal{G}') = 1$ for every edge-extension $\mathcal{G}'$ of $\mathcal{G}$, and $f(\mathcal{H}) = 0$ (Fig. 2, right); such GNN f exists by Theorem 2.1, since $\mathcal{H} \not\equiv_{\mathrm{C}_2} \mathcal{G}$ and $\mathcal{H} \not\equiv_{\mathrm{C}_2} \mathcal{G}'$ for any edge-extension $\mathcal{G}'$. Removing all negative literals from $\mathsf{Lit}(\mathcal{G})$ leaves a sufficient reason $\mathbb{S}$ consisting of just the six triangle edges: every completion of $\mathbb{S}$ is an edge-extension of $\mathcal{G}$, hence predicted 1. Thus, $\mathbb{S}$ is a certificate for 2^9 graphs ($\mathcal{G}$ and all its edge-extensions).

A C_2 sentence cannot mimic this economy. The natural analog (keeping the triangle edges, forbidding nothing) is minimum degree ≥ 2, $\psi_0 = \forall x \exists^{\geq 2} y\, E(x, y)$, but $\mathcal{H} \models \psi_0$ too, so ψ_0 is not sufficient: unlike $\mathbb{S}$, ψ_0 is anonymous and cannot say which six edges are present. Repairing it requires an upper degree bound as well, forcing full 2-regularity:

$$\varphi_{\mathcal{G}} = \mathsf{Size}_6 \wedge \forall x \exists^{=2} y\, E(x, y) \equiv \mathsf{Size}_6 \wedge \exists^{\geq 6} x \big(\exists^{\geq 2} y E(x, y) \wedge \exists^{\geq 4} y \neg E(x, y) \big),$$

whose only models up to isomorphism are $\{\mathcal{G}, \mathcal{C}_6\}$—both predicted 1, since $\mathcal{G} \equiv_{\mathrm{C}_2} \mathcal{C}_6$. So the weakest sufficient ψ collapses to $\varphi_{\mathcal{G}}$: it can drop none of the regularity information, where $\mathbb{S}$ dropped all of it. The weakest-sentence route thus generates a certificate only for two graphs $\{\mathcal{G}, \mathcal{C}_6\}$. Nevertheless, every sufficient reason $\mathbb{S}$ can still be associated with a primitive C_2 sentence it entails.

Definition 4.5. Let $\mathcal{G}$ be a graph of size n and $\mathbb{S} \subseteq \mathsf{Lit}(\mathcal{G})$. The *$C_2$-abstraction* of $\mathbb{S}$ is the strongest primitive C_2 sentence $\psi_{\mathbb{S}}$ over Graphs_n that holds in every completion of $\mathbb{S}$, i.e., if a primitive C_2 sentence φ holds in every completion of $\mathbb{S}$, then $\mathsf{Size}_n \wedge \psi_{\mathbb{S}} \models \varphi$.

The following theorem shows that every sufficient reason admits a C_2-abstraction; its proof appears in Appendix.

Theorem 4.6. *Let $\mathcal{G}$ be a graph of size n and $\mathbb{S} \subseteq \mathsf{Lit}(\mathcal{G})$. There is a C_2-abstraction of $\mathbb{S}$.*

Thus, every sufficient reason $\mathbb{S}$ can be paired with a C_2-abstraction $\psi_{\mathbb{S}}$ capturing the C_2-property responsible for the prediction $f(\mathcal{G})$: $\mathbb{S}$ is the concrete explanation, while $\psi_{\mathbb{S}}$ is the part of that explanation visible to C_2—and hence to the message-passing GNN itself.

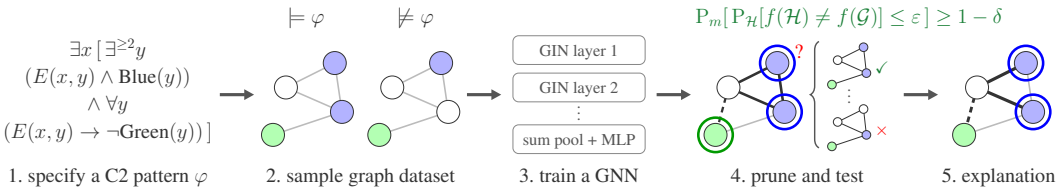


Figure 3: The workflow. A target property is specified in C_2, a dataset is sampled uniformly from its models and non-models, a GNN is trained, and an (ε, δ)-sufficient reason is extracted (Alg. 1).

Algorithm 1 Probabilistic algorithm computing an ε-sufficient reason

Require: GNN f, graph $\mathcal{G}$, $\varepsilon, \delta \geq 0$, distribution $\mathcal{D}$
Ensure: An ε-sufficient, $(\varepsilon/4)$-non-redundant reason for $f(\mathcal{G})$

```
𝕊 ← complete specification of 𝒢
m ← 12(ln(n(n+1)/δ))/ε where n = |Lit(𝒢)|
repeat
    removed ← False
    for each literal ℓ ∈ 𝕊 do
        sample m i.i.d. ℋ_i ~ 𝒟(· | 𝕊 \ {ℓ})
        μ̂ ← (1/m) Σ_{i=1}^{m} 1_{f(ℋ_i)≠f(𝒢)}
        if μ̂ ≤ ε/2 then                ▷ If the fraction of counter-examples is small, remove the literal.
            𝕊 ← 𝕊 \ {ℓ}
            removed ← True
        end if
    end for
until not removed or 𝕊 = ∅
return 𝕊
```

5 Finding Sufficient Reasons

Given a graph $\mathcal{G}$ and a GNN f, one can look for a minimal sufficient reason by iteratively removing literals from $\text{Lit}(\mathcal{G})$ and checking whether the value of f does not depend on the removed literals. This process is computationally demanding, as after removing k literals, we must test 2^k completions to determine whether all of them map to $f(\mathcal{G})$. To reduce the number of evaluations, we propose two variants of a probabilistic algorithm that samples completions of the partial specification and checks whether some of them change the prediction. The first variant is *strict*, i.e., if it finds a single completion $\mathcal{H}$ such that $f(\mathcal{H}) \neq f(\mathcal{G})$, it keeps the literal (App. Alg. 2). The second *soft* variant keeps a literal, even if there is a small portion of completions that change the prediction (Alg. 1). Neither algorithm needs to return a minimal sufficient reason, but we can at least obtain PAC guarantees that the returned set of literals is an ε-sufficient reason with probability $1-\delta$.

Definition 5.1. Let $\mathcal{D}$ be a distribution over graphs of size n. For a partial specification $\mathbb{S}$, let $\mathcal{D}(\,\cdot \mid \mathbb{S})$ denote the corresponding distribution conditioned on satisfying $\mathbb{S}$. We say that $\mathbb{S}$ is an *ε-sufficient reason w.r.t.* $\mathcal{D}$ for $f(\mathcal{G})$ if $P_{\mathcal{H}\sim\mathcal{D}(\cdot|\mathbb{S})}\big(f(\mathcal{H}) \neq f(\mathcal{G})\big) \leq \varepsilon$. A set of literals $\mathbb{S}$ is *ε-non-redundant* w.r.t. $\mathcal{D}$ if $\mathbb{S} \setminus \{\ell\}$ is not an ε-sufficient reason w.r.t. $\mathcal{D}$ for all $\ell \in \mathbb{S}$.

The distribution $\mathcal{D}$ can be uniform over all completions of $\mathbb{S}$, or induced by the training data, with the latter being more realistic, but requiring a generative model for the sampling, hence we use the former in the current implementation. Further, note that a sufficient reason from Definition 3.1 is a 0-sufficient reason w.r.t. $\mathcal{D}$ provided $\mathcal{D}$ assigns non-zero probability to every graph of size n. The following theorems provide PAC guarantees for both variants of the deletion algorithm; the proofs are in the Appendix.

Theorem 5.2. *Let f be a GNN classifier and $\mathcal{G}$ an input graph. Algorithm 1 returns a set of literals $\mathbb{S}$ that is ε-sufficient and $(\varepsilon/4)$-non-redundant w.r.t. $\mathcal{D}$ with probability at least $1-\delta$. Moreover, it needs $O(n^2 \ln n^2)$ evaluations of the GNN f where $n = |\text{Lit}(\mathcal{G})|$.*

Theorem 5.3. *Let f be a GNN classifier and $\mathcal{G}$ an input graph. Algorithm 2 returns an ε-sufficient reason w.r.t. $\mathcal{D}$ with probability at least $1-\delta$, which is deterministically 0-non-redundant w.r.t. $\mathcal{D}$. Moreover, it needs $O(n \ln n)$ evaluations of the GNN f where $n = |\text{Lit}(\mathcal{G})|$.*

Both variants have their pros and cons. The strict variant (Alg. 2) needs much fewer samples than Alg. 1 to guarantee ε-sufficiency with probability $1 - \delta$. On the other hand, it is more sensitive to noise caused by GNN imperfections. So to obtain a small ε-sufficient reason, one needs to restart Alg. 2 several times. In comparison, Alg. 1 is more stable, generating small ε-sufficient reasons. However, it needs to iterate until a fixed point because testing whether the fraction of counter-examples is less than ε is not monotone w.r.t. literal deletion.

6 EXPERIMENTS

We train GNNs on eight target properties specified as C_2 sentences (Sec. B.1) over graphs with $n = 10$ vertices (Graphs_{10}) and two binary node features (Tab. 1). Four are positive existential (monotone), i.e. witnessed by a set of present facts—an existential pattern ($\exists^{\geq 3} P$), its two-hop variant ($\exists^{\geq 2}\exists^{\geq 3} P$), and two degree conditions ($\exists^{\geq 7} E$, $\exists^{\geq 3}\exists^{\geq 6} E$), and the other four are not—the example property ($\exists\forall\neg P$, Fig. 1), a universal condition ($\forall\exists P$), and two patterns whose witness involves absent features ($\exists^{\geq 3}\neg P$) or edges ($\exists^{\geq 4}\neg E$). Since every target is a C_2 sentence, one can sample training and test graphs exactly uniformly from its models and its non-models by the FO model sampling (Wang et al., 2024), so the class balance and the "share" (Tab. 1) of each class in the Graphs_{10} are known (Kuželka, 2021). Each model is a 4-layer GIN with sum pooling (Xu et al., 2019), 128 hidden units, trained to saturation (Tab. 3), and we use 5 seeds over a set of 20–30 test graphs per pattern (Sec. B). Sufficient reasons are computed by Alg. 1 with uniform completion distribution, $\varepsilon = \delta = 0.01$, $m = 15564$, 10 restarts, and passes iterated to a fixpoint, reporting the smallest certified reason, with the strict test (Alg. 2 in Sec.A.2, $\varepsilon = \delta = 0.01$, $m = 878$) serving as a reference. See overview of the process in Fig. 3.

As baselines, we use common GNNExplainer (Ying et al., 2019) and GraphMask (Schlichtkrull et al., 2021), read as sets of facts via their top-k edges and node features (App. B.3). The "subgraph search" is then our implementation of the prime implicants (Azzolin et al., 2025b), implemented via the same greedy deletion (App. Alg. 2) with restarts but limited to present (positive) facts, accepting a removal through their "extension sufficiency test" (EST Azzolin et al., 2026). All methods are then judged by the same ε-sufficiency test on newly sampled completions. For the mask baselines, we report the share of instances whose top-k—aligned to our explanation size—passes the sufficiency test, and the smallest k that passes it (calibrated size‡ in Tab. 1). Sizes are also reported relative to each method's own explanation space (rel.), since a mask can only select present edges and (positive or negative) features, whereas our reasons range over all 65 literals (see App. Sec. B.5 for details). For comparability with the literature, we additionally report Fidelity$\pm$ (Yuan et al., 2023) and the EST (Azzolin et al., 2026) in Tab. 2.

Our method certifies an (ε, δ)-sufficient reason for every instance, positive or negative while, at the same top-k size, the explanations of GNNExplainer and GraphMask are almost never sufficient (Tab. 1). When calibrated to be sufficient, they need over twice as many facts, and for 5 of the target classes, no sufficient k exists at all. Interestingly, the mask-based methods do capture some non-monotone and negative predictions—though at considerably larger explanation sizes—since the node-feature mask can fix absent features (but not absent edges, e.g., the $\exists\forall\neg P$ from Fig. 1). As expected, the subgraph search then succeeds almost perfectly for positive instances of the monotone targets ('$\star$' in Tab. 1), while collapsing elsewhere. Indeed, within the space of the input subgraphs, even the empty explanation passes for a negative prediction of an existential property (which our sufficiency test rejects). The exceptions are also instructive – for $\forall\exists P^{+}$ and $\exists\forall\neg P^{-}$ a (corner-case) positive certificate actually exists ("for *each* of the 10 nodes there is an edge to a P_2 vertex", see Sec. B.1), though much larger than our reason capable of negations.

Table 2 then reports the metrics from the literature. Fidelity^{+} (i.e., "does removing the explanation change the prediction?") is twice as high for our explanation as for the baselines. Fidelity^{-} evaluates a single, very sparse, out-of-distribution completion (Zheng et al., 2024), thus favoring large explanations and penalizing the ε tolerance. Indeed, with the strict test (Alg. 2) our Fidelity^{-} drops from 0.26 to 0.13, at the price of app. 50% larger explanations; while the baselines' fidelities move by at most 0.08, even though their calibrated explanations also grow (by 5–20%) without gaining sufficiency (Tab. 6 in Sec. B.7). Similar logic then applies to the EST metric.[2] The relative success

[2]For a 1%-sufficient reason ($\varepsilon = 0.01$) the probability of one of 50 samples failing is up to $1 - (1 - \varepsilon)^{50} \approx 40\%$ by construction, hence the reported numbers are the expected signature of our certificate, not a weakness.

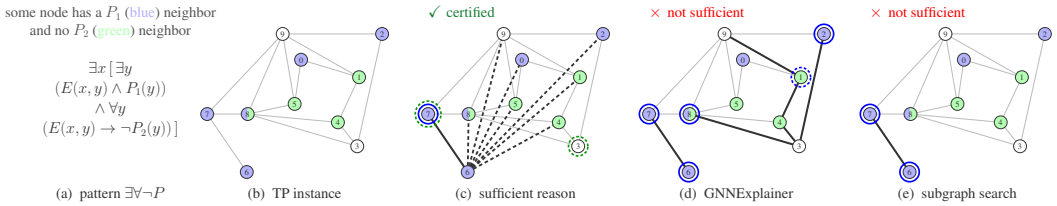


Figure 4: A positive instance (b) of the property (a) from Fig. 1. Our reason (c) marks the witness ('6') with its $P_1 \wedge \neg P_2$ neighbor ('7') and, for every other node, its non-adjacency to '6' (dashed edges, necessary for P_2 nodes) or $\neg P_2$ (dashed circle '3'), with 11 facts altogether, which is the optimum. The baselines (d,e), while marking some facts around the witness, remain insufficient.

Table 1: Per-pattern, per-class results: class share (%) of the Graphs_{10} space, then per method the ε-sufficiency (%), explanation size (mean±std over instances and five seeds, sufficient explanations only), and relative size (rel., %, Sec. B.7). †: sufficient by construction; ‡: sufficiency-calibrated top-k; '–' = no sufficient k found; *: positive class of a positive existential (monotone) target.

		Suff. reasons			GNNExplainer			GraphMask			Subgraph search		
pattern	share %	suff.† %	size	rel. %	suff. %	size‡	rel. %	suff. %	size‡	rel. %	suff. %	size	rel. %
$\exists^{\geq 3} P^{+\star}$	77.3	**100**	$\mathbf{7.3 \pm 0.8}$	**11**	0	24.4 ± 10.5	57	0	34.2 ± 6.9	80	97	13.6 ± 7.8	42
$\exists^{\geq 3} P^{-}$	22.7	**100**	$\mathbf{20.5 \pm 4.8}$	**32**	0	–	–	0	–	–	0	–	–
$\exists^{\geq 2}\exists^{\geq 3} P^{+\star}$	75.9	**100**	$\mathbf{8.0 \pm 0.7}$	**12**	0	27.2 ± 10.6	65	0	32.8 ± 7.8	79	**100**	18.3 ± 5.2	61
$\exists^{\geq 2}\exists^{\geq 3} P^{-}$	24.1	**100**	$\mathbf{20.5 \pm 6.3}$	**32**	0	23.0 ± 0.0	61	0	18.0 ± 0.0	47	0	–	–
$\exists^{\geq 3}\neg P^{+}$	77.3	**100**	$\mathbf{8.9 \pm 1.3}$	**14**	0	30.1 ± 7.5	70	0	34.8 ± 7.3	80	0	–	–
$\exists^{\geq 3}\neg P^{-}$	22.7	**100**	$\mathbf{14.2 \pm 4.5}$	**22**	0	28.7 ± 5.6	66	0	31.5 ± 9.7	75	0	–	–
$\exists^{\geq 4}\neg E^{+}$	59.7	**100**	$\mathbf{11.3 \pm 5.4}$	**17**	0	43.0 ± 0.8	96	0	41.3 ± 1.9	93	1	27.0	69
$\exists^{\geq 4}\neg E^{-}$	40.3	**100**	$\mathbf{17.4 \pm 12.2}$	**27**	0	36.3 ± 7.2	82	0	35.2 ± 9.9	79	0	–	–
$\forall\exists P^{+}$	73.6	**100**	$\mathbf{11.1 \pm 0.6}$	**17**	0	37.0 ± 4.7	84	0	36.7 ± 6.5	83	11	39.6 ± 3.8	100
$\forall\exists P^{-}$	26.4	**100**	$\mathbf{11.0 \pm 2.8}$	**17**	0	–	–	0	–	–	0	–	–
$\exists\forall\neg P^{+}$	34.4	**100**	$\mathbf{10.8 \pm 0.5}$	**17**	0	–	–	0	–	–	0	–	–
$\exists\forall\neg P^{-}$	65.6	**100**	$\mathbf{11.8 \pm 1.6}$	**18**	0	29.4 ± 9.0	62	0	37.8 ± 8.8	80	20	37.8 ± 9.0	91
$\exists^{\geq 7} E^{+\star}$	55.7	**100**	$\mathbf{7.0 \pm 0.0}$	**11**	0	37.0 ± 1.5	80	0	30.0 ± 8.5	65	**100**	27.4 ± 2.3	74
$\exists^{\geq 7} E^{-}$	44.3	**100**	$\mathbf{15.3 \pm 0.5}$	**24**	0	–	–	0	–	–	0	–	–
$\exists^{\geq 3}\exists^{\geq 6} E^{+\star}$	44.7	**100**	$\mathbf{14.5 \pm 0.5}$	**22**	0	37.4 ± 0.5	83	2	31.2 ± 8.4	69	**100**	30.1 ± 2.7	86
$\exists^{\geq 3}\exists^{\geq 6} E^{-}$	55.3	**100**	$\mathbf{14.0 \pm 0.0}$	**22**	0	–	–	0	–	–	0	–	–

of the subgraph search on these metrics is then an artifact of explanation size (none of its 278 ε-sufficient ones are small; 28 are the untouched input), while on 23% of instances (mostly negative predictions) it collapses (App. Tab. 7).

Notably, for properties where the optimal reason is known (App. Sec. B.1), our sizes often match it, such as the 7 for the seven edges of $\exists^{\geq 7} E^{+}$, 7.3 for $\exists^{\geq 3} P^{+}$ whose witness needs 7 (witness + properties of 3 neighbors) facts, or 10.8 (optimal $n + 1 = 11$ with the strict test, see Tab. 5) for the example property (Fig. 1) shown in Fig. 4. Negative predictions generally cost more (11–32 facts), since refuting an existential target means forbidding the structure everywhere, while universal targets need about n facts per class.

Computationally, one sufficiency test draws m samples (878 for strict, 15,564 for soft version); certifying a whole instance chains these tests across the literals and restarts, which our implemented WL-hash (Weisfeiler & Lehman, 1968) caching and batching reduce to ≈5–9 min (strict) vs. ≈70–115 min (soft) median wall-clock per instance on a single GPU (App. Tab. 5). Note that the (faster) strict variant still generates very reasonable, although generally larger, explanations (see App. Tab. 5).

7 Related Work

The post-hoc explainers (Sec. 1) differ in mechanism, but all return a subset of the facts *present* in the input. Counterfactual methods (Lucic et al., 2022; Bajaj et al., 2021) report which present edges to delete, but likewise cannot state that an absent edge matters. Global methods instead fit

Table 2: Pooled results over all patterns (mean±std across seeds). Suff. reasons under the soft (ε=δ=0.01) and strict test; mask baselines evaluated at our (soft) explanation's size, subgraph search at its own. EST: share (%) flipped on ≥1 of 50 sampled extensions (lower is better); $\text{rel}_\cap$ and rel_{imp}: relative sizes over the commonly sufficient and over all instances (Sec. B.5). Symbols as in Tab. 1.

	ε-suff. (%)	EST (%)	Fid^+	Fid^-	$\text{rel}_\cap$ (%)	rel_{imp} (%)
Suff. reasons (soft)	$\mathbf{100}^\dagger$	31 ± 2	$\mathbf{0.66 \pm 0.02}$	0.26 ± 0.03	$\mathbf{13 \pm 4}$	$\mathbf{19 \pm 10}$
Suff. reasons (strict)	$\mathbf{100}^\dagger$	$\mathbf{8 \pm 1}$	$\mathbf{0.66 \pm 0.01}$	0.13 ± 0.02	20 ± 8	31 ± 18
GNNExplainer (Ying et al., 2019)	0 ± 0	31 ± 2	0.36 ± 0.02	0.55 ± 0.03	70 ± 21	$43 \pm 28^\ddagger$
GraphMask (Schlichtkrull et al., 2021)	0 ± 0	50 ± 1	0.31 ± 0.02	0.52 ± 0.01	74 ± 18	$44 \pm 30^\ddagger$
Subgraph search (≈Azzolin et al., 2025b)	27 ± 1	12 ± 2	0.50 ± 0.01	$\mathbf{0.07 \pm 0.02}$	65 ± 23	33 ± 24

a rule that approximates f across a dataset —a boolean formula over learned concepts (Azzolin et al., 2023), computation trees (Armgaan et al., 2024), grounded predicates (Geng et al., 2026), or a distilled decision tree over counting formulas Pluska et al. (2024); Köhler & Heindorf (2024). Nevertheless, we show why their exactness is not guaranteed: a C_2 description of a GNN (Th. 4.1) is in general stratified by size (Th. 4.2), and its normal form may be exponentially long (Th. 4.3). The C_2-abstraction of Def. 4.5 is then the instance-level counterpart of such sought-after rules, with an exact consequence of a certified reason, rather than just a fit.

Besides Fidelity± (Pope et al., 2019; Yuan et al., 2023), explanations are scored by their combinations (Amara et al., 2022), the KL divergence of the models' outputs (Agarwal et al., 2023), or perturbations of the complement (Zheng et al., 2024; Fang et al., 2023; Azzolin et al., 2025a). The perturbed graphs are, however, known to be out-of-distribution (Zheng et al., 2024; Amara et al., 2023), as are the induced subgraphs (Azzolin et al., 2025b; 2026) (Sec. 6), which is why our completions keep the vertex set context fixed. Our criticism here is more fundamental. All existing tests evaluate positive subgraphs. As shown throughout this paper, this can only be sufficient for monotone properties and positive predictions (Sec. 6), even if the explanation subgraph is completely in-distribution. Notably, Weinbauer et al. (2025) name admitting all extensions from the domain (beyond subgraphs) as the general formulation and mention sampling-based search as an open problem. Our Algorithms 1, 2 address exactly that. Finally, comparing explanations to human-annotated motifs, presuming that the GNN sees it, is known to be unreliable (Faber et al., 2021),[3] which is why our experiments use well-defined C_2 targets (Grohe, 2021).

Sufficient reasons, prime implicants and abductive explanations (Shih et al., 2018; Ignatiev et al., 2019; Darwiche & Hirth, 2023) are one related notion, studied for fixed-dimensional input. Since finding the small ones is hard (Barceló et al., 2020), relaxations that hold for "most completions" have been explored (Wäldchen et al., 2021; Wang et al., 2021; Blanc et al., 2021). Our approach (Def. 5.1) can then be seen as their graph counterpart, addressing the implied issues of context bounding (Th. 4.2) and C_2 invariance (Grohe, 2021).

8 Conclusion

We proposed sufficient reasons for GNNs – explanation certificates rather than scores. Generally, every explanation is a statement about some set of alternative inputs, but with the existing mask-based methods, that set is wrong, since subgraphs of the input cannot say what must be absent, and they ignore that a GNN may decide graphs of different sizes by different rules. Our notion allows for negative facts and builds on the known C_2 characterization of the GNN decision space, which fixes the size context and provides a canonical abstraction of each reason. The reasons are found by a randomized algorithm with PAC guarantees, yielding a certified explanation for all tested C_2 properties, including non-monotone targets where the existing methods generally fail. The guarantees cost a considerable number of GNN evaluations, which can nevertheless be traded against the explanation quality through looser acceptance thresholds or fewer restarts. Scaling to larger graphs, e.g. by restricting the set of completions to a given input data neighborhood or distribution, is left for future work. Another future research direction is to investigate methods for succinctly representing sufficient reasons (i.e., sets of literals) as sentences of FO logic.

[3] For instance, when recognizing a "house-shaped" motif from the popular BA datasets (Ying et al., 2019).

AI USE STATEMENT

In this work, we used generative AI tools to refine hypotheses, provide feedback on research methodology and experiments, and to implement experimental methods. We have not used generative AI tools to generate synthetic data sets (we used FO model sampling), help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, assist with translation, clean and reformat datasets, support qualitative and thematic data analysis, or interpret results. Additionally, we used generative AI tools to create and improve TikZ figures, brainstorm GNN pattern selection, analyze the existing literature, and edit a research paper to improve readability. We have reviewed all AI-assisted work. We verified and tested the code's correctness. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

Our work introduces a general-purpose explanation method for graph neural networks tested on a synthetic dataset. Therefore, it does not raise major ethical concerns beyond the established risks connected to the deployment of machine learning methods.

REPRODUCIBILITY STATEMENT

The code supporting the experimental results can be found at `https://anonymous.4open.science/r/sufrex-D6F4/`, along with instructions for re-running the presented experiments and test configuration of the full pipeline. All datasets are regenerated exactly from the target formulas and fixed seeds (Sec. B), so the dataset needs not to be distributed separately. Seeds for sampling, training, and explanation are fixed throughout, and every reported number derives from per-instance artifacts stored by the pipeline. Expected runtimes are reported in Tab. 5. The experimental setting is described in Sec. 6 and App. B; proofs supporting the theoretical claims are in App. A.1.

REFERENCES

Chirag Agarwal, Owen Queen, Himabindu Lakkaraju, and Marinka Zitnik. Evaluating explainability for graph neural networks. *Scientific Data*, 10(144), 2023. doi: 10.1038/s41597-023-01974-x.

Kenza Amara, Zhitao Ying, Zitao Zhang, Zhichao Han, Yang Zhao, Yinan Shan, Ulrik Brandes, Sebastian Schemm, and Ce Zhang. Graphframex: Towards systematic evaluation of explainability methods for graph neural networks. In Bastian Rieck and Razvan Pascanu (eds.), *Learning on Graphs Conference, LoG 2022, 9-12 December 2022, Virtual Event*, volume 198 of *Proceedings of Machine Learning Research*, pp. 44. PMLR, 2022.

Kenza Amara, Mennatallah El-Assady, and Rex Ying. GInX-Eval: Towards in-distribution evaluation of graph neural network explanations. In *NeurIPS 2023 Workshop on XAI in Action: Past, Present, and Future Applications*, 2023.

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

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

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*, 2025a.

Steve Azzolin, Sagar Malhotra, Andrea Passerini, and Stefano Teso. Beyond topological self-explainable GNNs: A formal explainability perspective. In *Proceedings of the 42nd International Conference on Machine Learning, ICML 2025*, volume 267 of *Proceedings of Machine Learning Research*. PMLR, 2025b. arXiv:2502.02719.

Steve Azzolin, Stefano Teso, Bruno Lepri, Andrea Passerini, and Sagar Malhotra. GNN explanations that do not explain and how to find them. In *The Fourteenth International Conference on Learning Representations, ICLR 2026*, 2026. 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 *Advances in Neural Information Processing Systems 34, NeurIPS 2021*, pp. 5644–5655, 2021.

Pablo Barceló, Egor V. Kostylev, Mikaël Monet, Jorge Pérez, Juan L. Reutter, and Juan Pablo Silva. The logical expressiveness of graph neural networks. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

Pablo Barceló, Mikaël Monet, Jorge Pérez, and Bernardo Subercaseaux. Model interpretability through the lens of computational complexity. In *Advances in Neural Information Processing Systems 33, NeurIPS 2020*, 2020.

Guy Blanc, Jane Lange, and Li-Yang Tan. Provably efficient, succinct, and precise explanations. In *Advances in Neural Information Processing Systems 34, NeurIPS 2021*, 2021.

Jin-yi Cai, Martin Fürer, and Neil Immerman. An optimal lower bound on the number of variables for graph identification. *Combinatorica*, 12(4):389–410, 1992.

Adnan Darwiche. Logic for explainable AI. In *38th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2023, Boston, MA, USA, June 26-29, 2023*, pp. 1–11. IEEE, 2023. doi: 10.1109/LICS56636.2023.10175757.

Adnan Darwiche and Auguste Hirth. On the (complete) reasons behind decisions. *Journal of Logic, Language and Information*, 32(1):63–88, 2023.

Lukas Faber, Amin K. Moghaddam, and Roger Wattenhofer. When comparing to ground truth is wrong: On evaluating GNN explanation methods. In Feida Zhu, Beng Chin Ooi, and Chunyan Miao (eds.), *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021*, pp. 332–341. ACM, 2021.

Junfeng Fang, Wei Liu, Yuan Gao, Zemin Liu, An Zhang, Xiang Wang, and Xiangnan He. Evaluating post-hoc explanations for graph neural networks via robustness analysis. In *Advances in Neural Information Processing Systems 36, NeurIPS 2023*, 2023.

Matthias Fey and Jan Eric Lenssen. Fast graph representation learning with pytorch geometric. *arXiv preprint arXiv:1903.02428*, 2019.

Chuqin Geng, Ziyu Zhao, Zhaoyue Wang, Haolin Ye, Yuhe Jiang, and Xujie Si. LogicXGNN: Grounded logical rules for explaining graph neural networks. In *The Fourteenth International Conference on Learning Representations, ICLR 2026*, 2026.

Bernardo Cuenca Grau, Eva Feng, and Przemyslaw Andrzej Walega. The correspondence between bounded graph neural networks and fragments of first-order logic. In Sven Koenig, Chad Jenkins, and Matthew E. Taylor (eds.), *Fortieth AAAI Conference on Artificial Intelligence, Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence, Sixteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2026, Singapore, January 20-27, 2026*, pp. 19135–19142. AAAI Press, 2026. doi: 10.1609/AAAI.V40I23.38987. URL https://doi.org/10.1609/aaai.v40i23.38987.

Martin Grohe. The logic of graph neural networks. In *36th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2021, Rome, Italy, June 29 - July 2, 2021*, pp. 1–17. IEEE, 2021. doi: 10.1109/LICS52264.2021.9470677.

Alexey Ignatiev, Nina Narodytska, and João Marques-Silva. Abduction-based explanations for machine learning models. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pp. 1511–1519. AAAI Press, 2019. doi: 10.1609/AAAI.V33I01.33011511.

Dominik Köhler and Stefan Heindorf. Utilizing description logics for global explanations of heterogeneous graph neural networks. arXiv:2405.12654, 2024.

Ondrej Kuželka. Weighted first-order model counting in the two-variable fragment with counting quantifiers. *Journal of Artificial Intelligence Research*, 70:1281–1307, 2021.

Ana Lucic, Maartje A. ter Hoeve, Gabriele Tolomei, Maarten de Rijke, and Fabrizio Silvestri. CF-GNNExplainer: Counterfactual explanations for graph neural networks. In *Proceedings of the 25th International Conference on Artificial Intelligence and Statistics, AISTATS 2022*, 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.

Michael Mitzenmacher and Eli Upfal. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge University Press, 2005. ISBN 978-0-521-83540-4. doi: 10.1017/CBO9780511813603.

Christopher Morris, Martin Ritzert, Matthias Fey, William L. Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. Weisfeiler and leman go neural: Higher-order graph neural networks. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pp. 4602–4609. AAAI Press, 2019. doi: 10.1609/AAAI.V33I01.33014602.

Martin Otto. *Bounded Variable Logics and Counting: A Study in Finite Models*, volume 9 of *Lecture Notes in Logic*. Cambridge University Press, 2017. ISBN 9781316716878. doi: 10.1017/9781316716878.

Alexander Pluska, Pascal Welke, Thomas Gärtner, and Sagar Malhotra. Logical distillation of graph neural networks. In *ICML 2024 Workshop on Mechanistic Interpretability*, 2024.

Phillip E. Pope, Soheil Kolouri, Mohammad Rostami, Charles E. Martin, and Heiko Hoffmann. Explainability methods for graph convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pp. 10772–10781. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPR.2019.01103.

Michael Sejr Schlichtkrull, Nicola De Cao, and Ivan Titov. Interpreting graph neural networks for NLP with differentiable edge masking. In *9th International Conference on Learning Representations, ICLR 2021*, 2021.

Thomas Schnake, Oliver Eberle, Jonas Lederer, Shinichi Nakajima, Kristof T. Schütt, Klaus-Robert Müller, and Grégoire Montavon. Higher-order explanations of graph neural networks via relevant walks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(11):7581–7596, 2022. doi: 10.1109/TPAMI.2021.3115452.

Andy Shih, Arthur Choi, and Adnan Darwiche. A symbolic approach to explaining Bayesian network classifiers. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence, IJCAI 2018*, pp. 5103–5111, 2018.

Stephan Wäldchen, Jan MacDonald, Sascha Hauch, and Gitta Kutyniok. The computational complexity of understanding binary classifier decisions. *Journal of Artificial Intelligence Research*, 70:351–387, 2021. doi: 10.1613/jair.1.12359.

Eric Wang, Pasha Khosravi, and Guy Van den Broeck. Probabilistic sufficient explanations. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence, IJCAI 2021*, pp. 3082–3088, 2021.

Yuanhong Wang, Juhua Pu, Yuyi Wang, and Ondrej Kuželka. Lifted algorithms for symmetric weighted first-order model sampling. *Artif. Intell.*, 331:104114, 2024.

Klaus Weinbauer, Tieu-Long Phan, Peter F. Stadler, Thomas Gärtner, and Sagar Malhotra. Prime implicant explanations for reaction feasibility prediction. In *ECML-PKDD 2025 Workshop on Advances in Interpretable Machine Learning and Artificial Intelligence (AIMLAI)*, 2025. arXiv:2510.09226.

Boris Weisfeiler and AA Lehman. A reduction of a graph to a canonical form and an algebra arising during this reduction. *Nauchno-Technicheskaya Informatsia*, 2(9):12–16, 1968.

Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. A Comprehensive Survey on Graph Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1):4–24, January 2021. ISSN 2162-2388. doi: 10.1109/TNNLS.2020.2978386. IEEE Transactions on Neural Networks and Learning Systems.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

Zhitao Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. Gnnexplainer: Generating explanations for graph neural networks. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, 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 *Proceedings of the 38th International Conference on Machine Learning, ICML 2021*, volume 139 of *Proceedings of Machine Learning Research*, pp. 12241–12252. PMLR, 2021.

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. doi: 10.1109/TPAMI.2022.3204236.

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*, 2024.

A Theoretical Results and Proofs

A.1 Proofs for Section 4

We first recall terminology and notation used in the proofs of the following statements. Given a sentence φ, we call a graph $\mathcal{G}$ a *model* of φ. The collection of all models of φ is denoted $\mathsf{Mod}(\varphi)$. Similarly, given a set of sentences T, a model of T is a graph $\mathcal{G}$ satisfying all the sentences from T. All the models of T are denoted $\mathsf{Mod}(T)$. Recall that $\mathsf{Size}_n = \exists^{=n}(x = x)$.

Theorem A.1 (Theorem 4.1 restated)**.** *Let f be a GNN classifier and c a value in the range of f. There is a set T_c of C_2 sentences such that $f(\mathcal{G}) = c$ iff $\mathcal{G} \in \mathsf{Mod}(T_c)$ (i.e., $\mathcal{G} \models \varphi$ for all $\varphi \in T_c$).*

Proof. Recall that $f(\mathcal{G}) = f(\mathcal{H})$ for $\mathcal{G} \equiv_{\mathrm{C}_2} \mathcal{H}$ by Theorem 2.1. Let $C = f^{-1}(c)$ be the class of graphs classified by c. We define T_c as the set of C_2 sentences valid in all members of C. We show that $\mathsf{Mod}(T_c) = C$. Clearly, $C \subseteq \mathsf{Mod}(T_c)$. Conversely, suppose we have a graph $\mathcal{G}$ of size n not belonging to C. If C contains no member of size n, then $\neg\mathsf{Size}_n$ does not hold in $\mathcal{G}$ and belongs to T_c. Next, assume that there is at least one graph in C of size n. There are only finitely many non-isomorphic graphs $\mathcal{H}_i \in C$ of size n. For each $\mathcal{H}_i$ there is a C_2 sentence φ_i such that $\mathcal{H}_i \models \varphi_i$ and $\mathcal{G} \not\models \varphi_i$ (otherwise $\mathcal{G} \equiv_{\mathrm{C}_2} \mathcal{H}_i$). Consequently, $\mathsf{Size}_n \to \bigvee_i \varphi_i$ belongs to T_c and is not valid in $\mathcal{G}$. In both cases, we have $\mathcal{G} \notin \mathsf{Mod}(T_c)$. □

Lemma A.2. *There is a GNN $f\colon \mathsf{Graphs} \to \{0, 1\}$ such that both sets of C_2 sentences T_0, T_1 from Theorem 4.1 cannot be finitely presented.*

Proof. Let f output 1 iff the graph has more red than green vertices; f is realized by a message-passing GNN with sum readout. Suppose a single C_2 sentence φ axiomatized $f^{-1}(1)$ and let t be the largest threshold in its counting quantifiers. The two edgeless graphs on $2t+3$ vertices with $t+1$ red and $t+2$ green, resp. $t+2$ red and $t+1$ green vertices satisfy the same C_2 sentences with thresholds at most t as can be proved via the t-graded modification of 2-pebble games (Grau et al., 2026); yet f assigns them different values. Hence, no such φ exists, and neither T_1 nor (symmetrically) T_0 has a finite equivalent. □

Theorem A.3 (Restated Theorem 4.2)**.** *Let f be a GNN classifier and c a value in the range of f. For each $n \geq 1$ and each $\mathcal{H} \in \mathsf{Graphs}_n$, there is a C_2 sentence φ_n such that $\mathcal{H} \in f^{-1}(c)$ iff $\mathcal{H} \models \varphi_n$. Moreover, $f^{-1}(c) \cap \mathsf{Graphs}_n$ is a finite union of C_2-equivalence classes.*

Proof. Let $n \geq 1$. As there are only finitely many graphs of size n, there are only finitely many C_2-equivalence classes of graphs of size n. Thus $f^{-1}(c) \cap \mathsf{Graphs}_n = \bigcup_i C_i$ is a finite union of C_2-equivalence classes C_i. As any C_2-equivalence class C_i is definable by a C_2 sentence ψ_i (see (Otto, 2017, Lemma 1.33)), we can define $\varphi_n = \bigvee_i \psi_i$'s. Thus, for all $\mathcal{H} \in \mathsf{Graphs}_n$, we have $\mathcal{H} \in f^{-1}(c)$ iff $\mathcal{H} \models \varphi_n$. □

Theorem A.4 (Theorem 4.3 restated)**.** *Over graphs of fixed size n, every C_2 sentence φ is logically equivalent to a disjunction of primitive C_2 sentences.*

Proof. The proof of Theorem 4.3 follows from the following lemmas.

Lemma A.5. *Let $n \geq 1$, $k \leq n$, and G a graph of size n. The following equivalence holds in C_2:*

$$G \models \neg\exists^{\geq k} x\ \varphi(x) \text{ iff } G \models \exists^{\geq n-k+1} x\ \neg\varphi(x).$$

Proof. If there are not k distinct nodes x satisfying $\varphi(x)$, there must be at least $n-k+1$ nodes such that $\varphi(x)$ does not hold. Conversely, if there are at least $n-k+1$ nodes such that $\neg\varphi(x)$, there can be at most $n-(n-k+1) = k-1$ nodes such that $\varphi(x)$. □

Lemma A.6. *Let G be a graph. In C_2, we have $G \models \exists^{\geq k} x\ (\varphi(x) \vee \psi(x))$ iff*

$$G \models \bigvee_{k_1+k_2+k_3=k} \big(\exists^{\geq k_1} x\ (\varphi(x) \wedge \psi(x)) \wedge \exists^{\geq k_2} x\ (\varphi(x) \wedge \neg\psi(x)) \wedge \exists^{\geq k_3} x\ (\neg\varphi(x) \wedge \psi(x))\big).$$

Proof. ($\Rightarrow$): Let m_1 (resp. m_2, m_3) be the number of witnesses x for $\phi(x) \wedge \psi(x)$ (resp. $\varphi(x) \wedge \neg\psi(x)$, $\neg\varphi(x) \wedge \psi(x)$). By the assumption, we have $m_1 + m_2 + m_3 \geq k$. Consequently, there are k_1, k_2, k_3 such that $k_i \leq m_i$ and $k_1 + k_2 + k_3 = k$. Thus $G \models \exists^{\geq k_1} x\ (\varphi(x) \wedge \psi(x)) \wedge \exists^{\geq k_2} x\ (\varphi(x) \wedge \neg\psi(x)) \wedge \exists^{\geq k_3} x\ (\neg\varphi(x) \wedge \psi(x))$.

($\Leftarrow$): Let k_1, k_2, k_3 be such that $G \models \exists^{\geq k_1} x\ (\varphi(x) \wedge \psi(x)) \wedge \exists^{\geq k_2} x\ (\varphi(x) \wedge \neg\psi(x)) \wedge \exists^{\geq k_3} x\ (\neg\varphi(x) \wedge \psi(x))$. Note that k_i witnesses for the i-th conjunct cannot be among the witnesses for the remaining conjuncts. Thus, there are at least $k = k_1 + k_2 + k_3$ nodes x satisfying $\varphi(x) \vee \psi(x)$. □

The Lemmata A.5 and A.6 allow us to define a normal form for C_2-sentences over graphs of a fixed size n. By Lemma A.5, we first distribute negations up to the level of atomic formulas. Next, we distribute counting quantifiers and conjunctions over disjunctions. Thus, we end up with a DNF-like normal form being a disjunction of primitive C_2 sentences. However, the normal form can be exponentially long due to Lemma A.6. □

Corollary A.7 (Restated Corollary 4.4)**.** *Let $\mathcal{G}$ be a graph of size n. Its C_2-equivalence class is definable by a C_2 sentence $\mathsf{Size}_n \wedge \varphi_{\mathcal{G}}$ where $\varphi_{\mathcal{G}}$ is a primitive C_2 sentence, i.e., $\mathcal{H} \equiv_{C_2} \mathcal{G}$ iff $\mathcal{H} \models \mathsf{Size}_n \wedge \varphi_{\mathcal{G}}$.*

Proof. Every C_2-equivalence class of $\mathcal{G}$ is definable by a C_2 sentence $\varphi_{\mathcal{G}}$ (Otto, 2017, Lemma 1.33). By Theorem 4.3, $\varphi_{\mathcal{G}}$ is equivalent to a disjunction of primitive C_2 sentences. As $\mathsf{Mod}(\varphi \vee \psi) = \mathsf{Mod}(\varphi) \cup \mathsf{Mod}(\psi)$, $\varphi_{\mathcal{G}}$ must be equivalent to a primitive C_2 sentence; otherwise the C_2-equivalence class of $\mathcal{G}$ would contain two non-equivalent graphs. □

Algorithm 2 Probabilistic algorithm computing an inclusion-minimal ε-sufficient reason

Require: GNN f, graph $\mathcal{G}$, $\varepsilon, \delta \geq 0$, distribution $\mathcal{D}$
Ensure: An ε-sufficient 0-non-redundant reason for $f(\mathcal{G})$
  $\mathbb{S} \leftarrow$ complete specification of $\mathcal{G}$
  $m \leftarrow (\ln n/\delta)/\varepsilon$ where $n = |\mathsf{Lit}(\mathcal{G})|$
  for each literal $\ell \in \mathbb{S}$ **do**
    sample m i.i.d. $\mathcal{H} \sim \mathcal{D}(\cdot \mid \mathbb{S} \setminus \{\ell\})$
    if all samples $\mathcal{H}$ preserve $f(\mathcal{G})$ **then**
      $\mathbb{S} \leftarrow \mathbb{S} \setminus \{\ell\}$
    end if
  end for
  return $\mathbb{S}$

Theorem A.8 (Restated Theorem 4.6)**.** *Let $\mathcal{G}$ be a graph of size n and $\mathbb{S} \subseteq \mathsf{Lit}(\mathcal{G})$. There is a C_2-abstraction of $\mathbb{S}$.*

Proof. Let $\mathbb{S} \subseteq \mathsf{Lit}(\mathcal{G})$. Consider the set of all primitive C_2 sentences φ such that $\mathcal{H} \models \varphi$ for all $\mathcal{H} \in \mathsf{Comp}(\mathbb{S})$. There are only finitely many non-equivalent such sentences over Graphs_n. Let $\psi_{\mathbb{S}}$ be the conjunction of all such sentences. Then, $\psi_{\mathbb{S}}$ is a primitive C_2 sentence that is satisfied by all completions of $\mathbb{S}$. Moreover, if there were a stronger primitive C_2 sentence ψ' satisfied by all completions of $\mathbb{S}$, then ψ' would be included in the conjunction defining $\psi_{\mathbb{S}}$, contradicting the assumption that $\psi_{\mathbb{S}}$ is the strongest. Therefore, $\psi_{\mathbb{S}}$ is the strongest primitive C_2 sentence satisfied by all completions of $\mathbb{S}$. □

A.2 Algorithms and PAC Guarantees

We start with two lemmas that follow from Chernoff bounds.

Lemma A.9. *Let $X_1, \ldots, X_m$ be i.i.d. $\{0,1\}$-valued random variables following Bernoulli distribution with $P(X_i = 1) = p$, $X = \frac{1}{m}\sum_{i=1}^{m} X_i$, and $0 \leq \varepsilon < p$. Then*

$$P(X \leq \varepsilon/2) \leq \exp(-m\varepsilon/8).$$

Proof. Using the Chernoff bound from (Mitzenmacher & Upfal, 2005, Theorem 4.4), we have $P(mX \leq mp/2) \leq \exp(-mp/8)$. Consequently,

$$P(X \leq \varepsilon/2) = P(mX \leq m\varepsilon/2) \leq P(mX \leq mp/2) \leq \exp(-mp/8) \leq \exp(-m\varepsilon/8).$$

□

Lemma A.10. *Let $X_1, \ldots, X_m$ be i.i.d. $\{0,1\}$-valued random variables following Bernoulli distribution with $P(X_i = 1) = p$, $X = \frac{1}{m}\sum_{i=1}^{m} X_i$, and $4p \leq \varepsilon \leq 1$. Then*

$$P(X \geq \varepsilon/2) \leq \exp(-m\varepsilon/12).$$

Proof. Let $Y_1, \ldots, Y_m$ be the i.i.d. $\{0,1\}$-valued random variables following Bernoulli distribution with $P(Y_i = 1) = \varepsilon/4$ and $Y = \frac{1}{m}\sum_{i=1}^{m} Y_i$. As $P(X_i = 1) \leq P(Y_i = 1)$ for all i's, it follows $P(X \geq \varepsilon/2) \leq P(Y \geq \varepsilon/2) = P(mY \leq m\varepsilon/2)$. Note that $\mathbb{E}[mY] = m\varepsilon/4$. Using the Chernoff bound from (Mitzenmacher & Upfal, 2005, Theorem 4.5), we have $P(mY \leq 2(m\varepsilon/4)) \leq \exp(-m\varepsilon/12)$. □

Theorem A.11 (Restated Theorem 5.2)**.** *Let f be a GNN classifier and $\mathcal{G}$ an input graph. Algorithm 1 returns a set of literals $\mathbb{S}$ that is ε-sufficient and $(\varepsilon/4)$-non-redundant w.r.t. $\mathcal{D}$ with probability at least $1 - \delta$. Moreover, it needs $O(n^2 \ln n^2)$ evaluations of the GNN f where $n = |\mathsf{Lit}(\mathcal{G})|$.*

Proof. Algorithm 1 is based on successive deletions of probably redundant literals. The inner loop iterates through all remaining literals $\ell \in \mathbb{S}$ and estimates the number of completions $\mathsf{Comp}(\mathbb{S} \setminus \{\ell\}$ that changes the prediction by sampling $m = 12(\ln(n(n+1)/\delta))/\varepsilon$ completions $\mathcal{H}_1, \ldots, \mathcal{H}_m \in \mathsf{Comp}(\mathbb{S})$. If the estimated mean $\hat{\mu} = \frac{1}{m}\sum_{i=1}^{m} \mathbf{1}_{f(\mathcal{H}_i) \neq f(\mathcal{G})}$ is less than or equal to $\varepsilon/2$, the literal ℓ is removed from $\mathbb{S}$ because it contains only a small portion of counter-examples among the samples.

The outer loop computes a fixed point. This is necessary as the portion of counter-examples is not, in general, monotone w.r.t. removing literals. In the worst case, each inner loop deletes only a single literal. Thus, the number of outer-loop iterations is at most n. Indeed, consider the last outer iteration. If it only tests the fixed point (without removing any literal), it must be at most the $(n - |\mathbb{S}| + 1)$-st iteration for $|\mathbb{S}| > 0$ and thus $n - |\mathbb{S}| + 1 \leq n$. If it ends with $\mathbb{S} = \emptyset$, then it must be at most the n-th iteration. Consequently, the number of all inner-loop iterations is at most $n+n-1+n-2+\cdots+1 = n(n+1)/2$. Let $N = n(n+1)/2$. So the number of GNN evaluations is $mN \in O(n^2 \ln n^2)$.

It remains to prove that the returned set of literals $\mathbb{S}$ is ε-sufficient and $(\varepsilon/4)$-non-redundant with probability at least $1 - \delta$. Given a set of literals $\mathbb{S} \subseteq \mathsf{Lit}(\mathcal{G})$, define the probability $p_{\mathbb{S}}$ that randomly chosen completion from $\mathsf{Comp}(\mathbb{S})$ changes the decision, i.e., $p_{\mathbb{S}} = P_{\mathcal{H}\sim\mathcal{D}(\cdot|\mathbb{S})}\big(f(\mathcal{H}) \neq f(\mathcal{G})\big)$. So $p_{\mathbb{S}}$ reflects the portion of counter-examples among $\mathsf{Comp}(\mathbb{S})$.

Consider a particular literal ℓ and $\mathbb{S}' = \mathbb{S} \setminus \{\ell\}$. Assume the algorithm falsely removes ℓ, although $\mathbb{S}'$ is not ε-sufficient (i.e., $p_{\mathbb{S}'} > \varepsilon$). This happens when $\hat{\mu} \leq \varepsilon/2$ with probability at most $\exp(-m\varepsilon/8)$ by Lemma A.9. Note that $\exp(-m\varepsilon/8) \leq \exp(-m\varepsilon/12)$. Dually, suppose the algorithm falsely keeps ℓ, although $\mathbb{S}'$ is $(\varepsilon/4)$-sufficient (i.e., $p_{\mathbb{S}'} \leq \varepsilon/4$). This happens when $\hat{\mu} > \varepsilon/2$ with probability at most $\exp(-m\varepsilon/12)$ by Lemma A.10. By the union bound, the probability that at least one of the above situations happens within N iterations is bounded by $2N\exp(-m\varepsilon/12)$. To get confidence δ, we need $2N\exp(-m\varepsilon/12) \leq \delta$ which is equivalent to $m \geq \frac{12\ln(n(n+1)/\delta)}{\varepsilon}$. Thus, the returned $\mathbb{S}$ is ε-sufficient and $(\varepsilon/4)$-non-redundant with probability $1 - \delta$. □

Theorem A.12 (Restated Theorem 5.3)**.** *Let f be a GNN classifier and $\mathcal{G}$ an input graph. Algorithm 2 returns a ε-sufficient reason w.r.t. $\mathcal{D}$ with probability at least $1 - \delta$, which is deterministically 0-non-redundant w.r.t. $\mathcal{D}$. Moreover, it needs $O(n \ln n)$ evaluations of the GNN f where $n = |\mathsf{Lit}(\mathcal{G})|$.*

Proof. Algorithm 2 makes a single loop through all literals $\mathsf{Lit}(\mathcal{G})$ and deletes a literal ℓ if it pass the strict test (i.e., all the samples do not change the prediction). Thus, it makes at most n iterations. As all the true redundant literals pass the strict test (they must be removed), the resulting set $\mathbb{S}$ is deterministically 0-non-redundant.

Assume that it falsely removes a literal $\ell \in \mathsf{Lit}(\mathcal{G})$ although $P(f(\mathcal{H}) = f(\mathcal{G}) \mid \mathbb{S}) \geq 1 - \varepsilon$. The probability that this happens is at least $(1 - \varepsilon)^m$. To get the confidence δ in n iterations, we need to have $n(1 - \varepsilon)^m \leq \delta$. Using $1 - \varepsilon \leq \exp(-\varepsilon)$, we have

$$n(1-\varepsilon)^m \leq n\exp(-m\varepsilon) \leq \delta \implies m \geq \frac{\ln n/\delta}{\varepsilon}$$

As there is at most n iterations, we need a total of $\frac{n\ln n/\delta}{\varepsilon} \in O(n \ln n)$ evaluations of f. □

B Experimental Details

B.1 Target Formulas

The following patterns were used in the experiments, including the C_2 sentence defining the positive (resp. negative) class. The sentences describing negative classes are the negations of the sentences describing positive ones. We leverage the fact that our graphs are of size 10 and apply Lemma A.5 to distribute the negation to the atomic formulas.

1. $\exists^{\geq 3}P$: there is a P_1-node having at least three P_2-neighbors.
   - the positive class: $\exists x\big(P_1(x) \wedge \exists^{\geq 3} y(E(x,y) \wedge P_2(y))\big)$,
   - the negative class: $\forall x\big(\neg P_1(x) \vee \exists^{\geq 8} y(\neg E(x,y) \vee \neg P_2(y))\big)$.
2. $\exists^{\geq 2}\exists^{\geq 3}P$: there is a P_1-node having at least two neighbors such that each of them has at least three P_2-neighbors.
   - the positive class: $\exists x\Big(P_1(x) \wedge \exists^{\geq 2} y\big(E(x,y) \wedge \exists^{\geq 3} x(E(y,x) \wedge P_2(y))\big)\Big)$,

- the negative class: $\forall x\Big(\neg P_1(x) \vee \exists^{\geq 9} y\big(\neg E(x,y) \vee \exists^{\geq 8} x(\neg E(y,x) \vee \neg P_2(y))\big)\Big)$.

3. $\exists^{\geq 3}\neg P$: there is a P_1-node having at least three $\neg P_2$-neighbors.
   - the positive class: $\exists x\big(P_1(x) \wedge \exists^{\geq 3} y(E(x,y) \wedge \neg P_2(y))\big)$,
   - the negative class: $\forall x\big(\neg P_1(x) \vee \exists^{\geq 8} y(\neg E(x,y) \vee P_2(y))\big)$.
4. $\exists^{\geq 4}\neg E$: there is a P_1-node having at least four P_2-non-neighbors.
   - the positive class: $\exists x\big(P_1(x) \wedge \exists^{\geq 4} y(\neg E(x,y) \wedge P_2(y))\big)$,
   - the negative class: $\forall x\big(\neg P_1(x) \vee \exists^{\geq 7} y(E(x,y) \vee \neg P_2(y))\big)$.
5. $\forall\exists P$: every P_1-node has a P_2-neighbor.
   - the positive class: $\forall x\big(P_1(x) \to \exists y(E(x,y) \wedge P_2(y))\big)$,
   - the negative class: $\exists x\big(P_1(x) \wedge \forall y(\neg E(x,y) \vee \neg P_2(y))\big)$.
6. $\exists\forall\neg P$: there is a node having a P_1-neighbor and none its neighbors has the property P_2.
   - the positive class: $\exists x\big(\exists y(E(x,y) \wedge P_1(y)) \wedge \forall y(E(x,y) \to \neg P_2(y))\big)$,
   - the negative class: $\forall x\big(\forall y(\neg E(x,y) \vee \neg P_1(y)) \vee \exists y(E(x,y) \wedge P_2(y))\big)$.
7. $\exists^{\geq 7} E$: there is a node of degree at least seven.
   - the positive class: $\exists x \exists^{\geq 7} y E(x,y)$,
   - the negative class: $\forall x \exists^{\geq 4} y \neg E(x,y)$.
8. $\exists^{\geq 3}\exists^{\geq 6} E$: there is a node having at least three neighbors of degree at least six.
   - the positive class: $\exists x \exists^{\geq 3} y\big(E(x,y) \wedge \exists^{\geq 6} x E(y,x)\big)$,
   - the negative class: $\forall x \exists^{\geq 8} y\big(\neg E(x,y) \vee \exists^{\geq 5} x\, \neg E(y,x)\big)$.

Interestingly, in some cases, we can precisely determine the minimum size of a sufficient reason. For a positive instance of a positive existential pattern, a minimal sufficient reason w.r.t. the pattern itself must fix a complete witness, so its size does not depend on n: 7 facts for ($\exists^{\geq 3} P$) and ($\exists^{\geq 3}\neg P$): $P_1(x)$, three edges and three P_2 resp. $\neg P_2$ facts, 9 for ($\exists^{\geq 4}\neg E$): $P_1(x)$, four non-edges and four P_2 facts, and 7 for ($\exists^{\geq 7} E$). For ($\exists\forall\neg P$) the reason must fix the P_1 neighbor ($E(x,y)$, $P_1(y)$, $\neg P_2(y)$) and, for each of the remaining $n-2$ vertices, either its non-adjacency to x or its $\neg P_2$ fact, i.e. $n+1=11$ facts. For other cases, the minimum size cannot be easily determined and is instance-dependent. Universal patterns and negative predictions of positive existential patterns need a fact about every vertex, hence at least n.

Sometimes non-monotone patterns can be explained by positive literals, explaining why the subgraph search found an ε-sufficient reason for some instances (Tab. 1). For instance, consider the positive class of ($\forall\exists P$). Its defining sentence can be equivalently expressed as $\forall x(\neg P_1(x) \vee \exists y(E(x,y) \wedge P_2))$. The disjunction gives us a choice in how to certify the validity of this sentence. The certificate found by the subgraph search for ($\forall\exists P^+$) is of this kind: for each of the 10 vertices, it fixes an edge to a P_2 vertex together with that vertex's P_2 fact, which is sufficient without negative literals but could be even twice as large as a sufficient reason exploiting the negative literal $\neg P_1(x)$.

B.2 Data and Training

For each target, we sample $5,000$ graphs from each class (10,000 per target, balanced) exactly uniformly from the models and non-models of the target sentence over the Graphs_{10} space via weighted first-order model sampling (Wang et al., 2024), or, where the lifted compilation is infeasible ($\exists^{\geq 2}\exists^{\geq 3} P$, $\exists\forall\neg P$), via equally uniform rejection sampling. The class shares in Tab. 1 are thus the corresponding exact counts normalized by the space size ($|\mathsf{Graphs}_{10}| = 2^{65}$), resp. Monte-Carlo estimates for the two rejection-sampled targets. Graphs are split $80/20$ into training and test sets. Each GNN is a 4-layer GIN with sum pooling (Xu et al., 2019), 128 hidden units, trained with Adam (batch size 64, using initial learning rate of 10^{-3} with cosine annealing; no dropout or weight decay) for at most 600 epochs with early stopping after 10 consecutive epochs at 100% training accuracy (test accuracies visible in Tab. 3). A model is then trained independently for each of the 5 seeds on the same data.

For each target pattern-class, we explain a fixed set of 20–30 test graphs shared across the five seeds: the ten lowest- and ten highest-confidence correct predictions (both classes covered), plus up to ten misclassified ones, where the model errs at all (1,025 explanations in total). The misclassified and low-confidence instances are thus over-represented, which (deliberately) tests the robustness of the approach and makes the reported explanations conservative.

B.3 Explanation Baselines

For our experiments, we integrate two GNN explanation baselines available in PyTorch Geometric (PyG) (Fey & Lenssen, 2019), GNNExplainer (Ying et al., 2019) and GraphMaskExplainer (Schlichtkrull et al., 2021).

Our graphs are undirected, which is in PyG represented as directed graph with symmetric edge pairs. While the original (dense) implementation of GNNExplainer enforces a symmetric mask in every optimization step ($M \leftarrow (M + M^\top)/2$), the PyG re-implementation dropped this step and optimizes each diretion independently. We restore the original behavior, resulting in one learnable logit per each undirected edge.

GraphMask uses layer-wise directed messages by design, with no reference implementation enforcing the symmetry. We leave the optimization untouched and only map the directed output to undirected facts at read-out, using mean of the two directed values (the actual gap in the direction values is small, approx. 0.091 when pooled over the soft experiments).

The outputs of both methods—scored present edges and node-feature mask entries (including negative feature facts)—are then interpreted as ranked sets of facts. The final explanation is selected using the top-k threshold on the ranked set. The sufficiency-calibrated top-k and the size-aligned evaluation of these rankings are defined in Sec. 6.

B.4 Completion Sampling

The uniform completions are sampled by setting each of the unspecified atoms (45 for undirected edges and 20 for node features, hence 65 in total) independently to true or false (with uniform $1/2$ probability). The sampled graphs are further accompanied by a Weisfeiler-Leman hash, so that isomorphic completions are only evaluated once, with their predictions cached. Experiments ran on a single GPU, with runtimes reported in Tab. 5.

B.5 Metrics

We report the relative size of an explanation as a percentage of the method's own explanation space at that instance: all $B = 65$ literals for sufficient reasons, the attribute atoms plus present edges for the mask explainers, and the positive atoms of the instance for the subgraph search from Sec. 6. In Tab. 2, $\text{rel}_\cap$ averages relative sizes over the $N_\cap = 255$ of $N = 1025$ instances for which every method found a sufficient explanation, and rel_{imp} over all instances, imputing an instance without a sufficient explanation by the relative size of our own soft explanation (to avoid over-penalizing the baselines). ε-sufficiency and Fidelity$^\pm$ are evaluated for every method on the full n-vertex graph, never on induced subgraphs: ε-sufficiency over sampled completions of the explanation, Fidelity$^-$ on the single completion with all unspecified features zero and unspecified edges absent, and Fidelity$^+$ on the instance with the explanation's atoms removed; EST and the acceptance test of the subgraph search are the only induced-subgraph evaluations, reported verbatim as defined by Azzolin et al. (2026). EST is computed only for explanations its induced subgraph formalism can represent; explanations with negative facts only are excluded. Fidelities are averaged per seed, and their standard deviation is reported across the five seeds.

B.6 Example Explanations

Figures 5, 6, 7, 8, and 9 show visualizations of explanations on additional testing instances, comparing our sufficient reasons with GNNExplainer and subgraph search. The instances have been chosen from the seed 100 of the explained set and cover true positive and true negative cases. The asymmetry between positive and negative classes is apparent even in our sufficient reasons: while

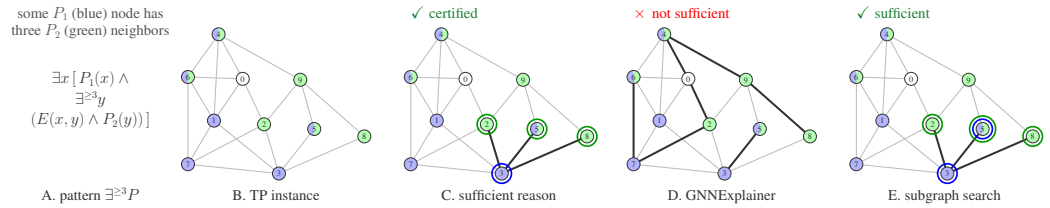


Figure 5: Conventions as in Fig. 4. The monotone base case where our sufficient reason explanation corresponds to the witness. Subgraph search also succeeds (thanks to the lack of negations), albeit with a slightly larger explanation. GNNexplainer at the size aligned to ours fails.

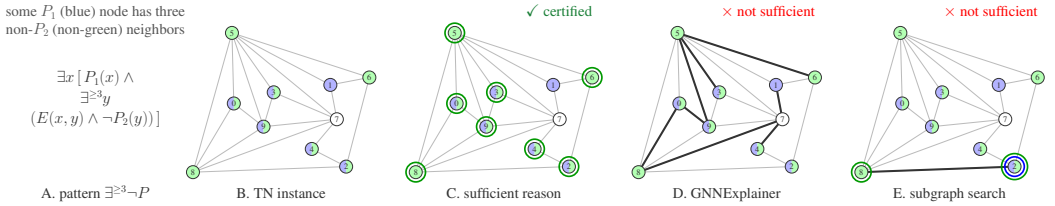


Figure 6: Conventions as in Fig. 4. The negative prediction of an absence-witnessed pattern. The sufficient reason forbids a witness. To see why, consider the description of the negative class: $\forall x\big(\neg P_1(x) \vee \exists^{\geq 8} y(\neg E(x,y) \vee P_2(y))\big)$. This sentence is entailed by $\exists^{\geq 8} y P_2(y)$. Both baselines fail.

the positive witness of a pattern can be compact, its absence often requires disallowing many more literals.

B.7 Detailed Results

The tables below describe additional experimental details and different types of result aggregations and comparisons. Tab. 3 shows training and testing set accuracies of our GNNs for each of the patterns. The differences in the test accuracies reflect the varying complexities of pattern detection w.r.t. inductive biases of the GNN. Tab. 4 provides a break-down of results from Tab. 2, isolating per-pattern results, but pooling the positive and negative instances in each pattern-based subset. Tab. 5 compares the strict and soft regime of our sufficient reasons, broken down by patterns and class membership. Tab. 6 provides the results of the strict version of sufficient reasons (as an alternative to the result of the soft version from Tab. 1). Tab. 7 shows the degenerate or trivial solutions that subgraph search often resorts to, but which do not provide a meaningful explanation.

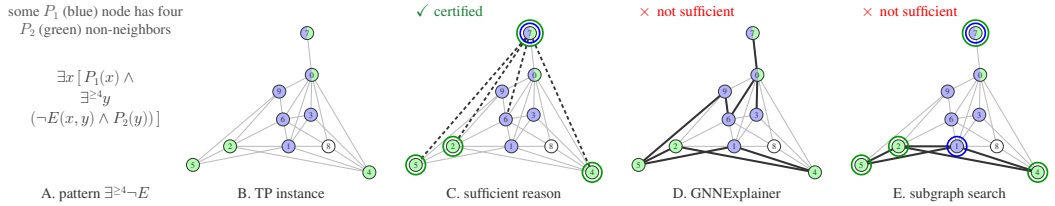


Figure 7: Conventions as in Fig. 4. The pattern witness contains absent edges, so it cannot be expressed via a strict subgraph. Thus, both baselines are insufficient. Our sufficient reason detects P_1 node '7' having four P_2 non-neighbors, including '7' itself (our graphs do not contain self-loops and '7' possesses both properties P_1, P_2). The non-edge between '6' and '7' would be redundant, provided GNN were a perfect classifier.

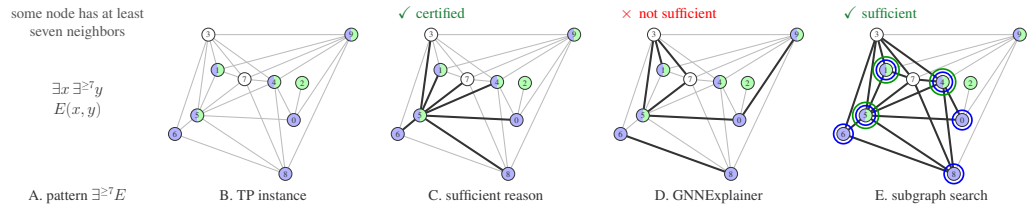


Figure 8: Conventions as in Fig. 4. Our reason cleanly corresponds to the witness. Subgraph search explanation is also sufficient but approx. $4\times$ larger. GNNExplainer fails at size 7.

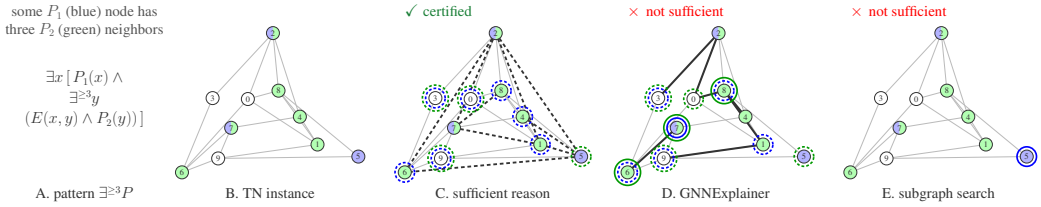


Figure 9: Conventions as in Fig. 4. Same pattern as Fig. 5, but a negative instance. To rule out the witness, the reason must state what is absent almost everywhere. The negative class is defined by the sentence $\varphi = \forall x \big(\neg P_1(x) \vee \exists^{\geq 8} y (\neg E(x, y) \vee \neg P_2(y))\big)$. Note that all the nodes except '2', '5', '7' are non-P_1, so they satisfy φ. For the remaining P_1 nodes, there are at least 8 nodes that are either their non-neighbors or non-P_2. For instance, '7' has four non-neighbors ('1', '2', '8', '7') and there are four non-P_2 nodes ('0', '3', '5', '9'). Neither baseline finds a sufficient explanation.

Table 3: Train and test accuracy of the explained GIN models per pattern (%; mean±std over the five seeds, and the worst test seed). The deviation is purely test-side generalization. Pattern names as in Tab. 1.

pattern	train acc. (%)	test acc. (%)	worst seed
$\exists^{\geq 3} P$	100.0 ± 0.0	99.5 ± 0.2	99.2
$\exists^{\geq 2} \exists^{\geq 3} P$	100.0 ± 0.0	98.8 ± 0.0	98.8
$\exists^{\geq 3} \neg P$	100.0 ± 0.0	92.3 ± 2.3	89.4
$\exists^{\geq 4} \neg E$	100.0 ± 0.0	94.3 ± 2.5	90.5
$\forall \exists P$	100.0 ± 0.0	99.9 ± 0.1	99.8
$\exists \forall \neg P$	100.0 ± 0.0	100.0 ± 0.0	100.0
$\exists^{\geq 7} E$	100.0 ± 0.0	100.0 ± 0.0	100.0
$\exists^{\geq 3} \exists^{\geq 6} E$	100.0 ± 0.0	100.0 ± 0.0	100.0

Table 4: Per-pattern breakdown of Table 2; symbols and conventions as there ($N_\cap$ per pattern block; '–' in $\text{rel}_\cap$ marks an empty intersection, where the column is undefined for every method).

	ε-suff. (%)	EST (%)	Fid^+	Fid^-	$\text{rel}_\cap$ (%)	rel_{imp} (%)
$\exists^{\geq 3} P$ ($N_\cap$=61 of 150)						
Suff. reasons (soft)	$\mathbf{100}^\dagger$	29 ± 14	0.61 ± 0.05	0.07 ± 0.03	$\mathbf{11 \pm 1}$	$\mathbf{21 \pm 11}$
Suff. reasons (strict)	$\mathbf{100}^\dagger$	9 ± 5	$\mathbf{0.63 \pm 0.02}$	0.07 ± 0.05	20 ± 9	31 ± 15
GNNExplainer	0 ± 0	31 ± 5	0.27 ± 0.08	0.51 ± 0.10	57 ± 24	$40 \pm 22^\ddagger$
GraphMask	0 ± 0	42 ± 7	0.42 ± 0.09	0.51 ± 0.09	80 ± 14	$50 \pm 28^\ddagger$
Subgraph search	49 ± 9	$\mathbf{3 \pm 5}$	0.51 ± 0.11	$\mathbf{0.00 \pm 0.00}$	41 ± 23	36 ± 18
$\exists^{\geq 2}\exists^{\geq 3} P$ ($N_\cap$=81 of 150)						
Suff. reasons (soft)	$\mathbf{100}^\dagger$	75 ± 5	0.57 ± 0.04	0.57 ± 0.02	$\mathbf{12 \pm 1}$	$\mathbf{20 \pm 12}$
Suff. reasons (strict)	$\mathbf{100}^\dagger$	19 ± 3	$\mathbf{0.65 \pm 0.06}$	0.31 ± 0.02	25 ± 6	34 ± 16
GNNExplainer	0 ± 0	31 ± 6	0.51 ± 0.02	0.57 ± 0.02	65 ± 24	$51 \pm 25^\ddagger$
GraphMask	0 ± 0	49 ± 3	0.47 ± 0.02	0.59 ± 0.04	79 ± 16	$58 \pm 27^\ddagger$
Subgraph search	57 ± 2	$\mathbf{13 \pm 3}$	0.59 ± 0.05	$\mathbf{0.01 \pm 0.02}$	61 ± 18	49 ± 21
$\exists^{\geq 3}\neg P$ ($N_\cap$=0 of 150)						
Suff. reasons (soft)	$\mathbf{100}^\dagger$	49 ± 6	$\mathbf{0.85 \pm 0.04}$	0.39 ± 0.27	–	$\mathbf{18 \pm 7}$
Suff. reasons (strict)	$\mathbf{100}^\dagger$	$\mathbf{7 \pm 4}$	0.84 ± 0.06	$\mathbf{0.05 \pm 0.04}$	–	46 ± 17
GNNExplainer	0 ± 0	28 ± 7	0.43 ± 0.09	0.47 ± 0.05	–	$38 \pm 26^\ddagger$
GraphMask	0 ± 0	43 ± 3	0.26 ± 0.06	0.47 ± 0.10	–	$42 \pm 30^\ddagger$
Subgraph search	0 ± 0	25 ± 6	0.57 ± 0.07	0.09 ± 0.13	–	$\mathbf{18 \pm 7}$
$\exists^{\geq 4}\neg E$ ($N_\cap$=0 of 150)						
Suff. reasons (soft)	$\mathbf{100}^\dagger$	41 ± 3	$\mathbf{0.73 \pm 0.06}$	0.20 ± 0.14	–	$\mathbf{22 \pm 16}$
Suff. reasons (strict)	$\mathbf{100}^\dagger$	$\mathbf{11 \pm 4}$	0.60 ± 0.04	$\mathbf{0.03 \pm 0.04}$	–	45 ± 20
GNNExplainer	0 ± 0	24 ± 4	0.11 ± 0.09	0.59 ± 0.05	–	$44 \pm 31^\ddagger$
GraphMask	0 ± 0	57 ± 6	0.40 ± 0.02	0.55 ± 0.03	–	$43 \pm 31^\ddagger$
Subgraph search	1 ± 1	29 ± 10	0.43 ± 0.05	0.38 ± 0.06	–	23 ± 16
$\forall\exists P$ ($N_\cap$=3 of 125)						
Suff. reasons (soft)	$\mathbf{100}^\dagger$	9 ± 6	0.82 ± 0.02	$\mathbf{0.00 \pm 0.00}$	$\mathbf{17 \pm 0}$	$\mathbf{17 \pm 3}$
Suff. reasons (strict)	$\mathbf{100}^\dagger$	8 ± 3	$\mathbf{0.85 \pm 0.05}$	0.01 ± 0.02	50 ± 1	24 ± 11
GNNExplainer	0 ± 0	38 ± 2	0.35 ± 0.03	0.78 ± 0.06	90 ± 0	$37 \pm 31^\ddagger$
GraphMask	0 ± 0	60 ± 9	0.35 ± 0.03	0.53 ± 0.07	95 ± 3	$37 \pm 31^\ddagger$
Subgraph search	6 ± 3	$\mathbf{5 \pm 3}$	0.28 ± 0.06	$\mathbf{0.00 \pm 0.00}$	100 ± 0	22 ± 19
$\exists\forall\neg P$ ($N_\cap$=5 of 100)						
Suff. reasons (soft)	$\mathbf{100}^\dagger$	6 ± 2	0.60 ± 0.00	$\mathbf{0.02 \pm 0.04}$	$\mathbf{17 \pm 0}$	$\mathbf{17 \pm 2}$
Suff. reasons (strict)	$\mathbf{100}^\dagger$	$\mathbf{0 \pm 0}$	$\mathbf{0.61 \pm 0.02}$	0.04 ± 0.02	18 ± 2	18 ± 2
GNNExplainer	0 ± 0	35 ± 3	0.36 ± 0.04	0.36 ± 0.02	75 ± 9	$25 \pm 18^\ddagger$
GraphMask	0 ± 0	53 ± 4	0.06 ± 0.02	0.47 ± 0.12	82 ± 8	$28 \pm 25^\ddagger$
Subgraph search	5 ± 3	11 ± 6	0.54 ± 0.04	0.03 ± 0.02	91 ± 15	21 ± 16
$\exists^{\geq 7} E$ ($N_\cap$=65 of 100)						
Suff. reasons (soft)	$\mathbf{100}^\dagger$	$\mathbf{0 \pm 0}$	0.60 ± 0.00	0.39 ± 0.32	$\mathbf{11 \pm 0}$	$\mathbf{15 \pm 6}$
Suff. reasons (strict)	$\mathbf{100}^\dagger$	$\mathbf{0 \pm 0}$	0.60 ± 0.00	0.39 ± 0.32	$\mathbf{11 \pm 0}$	$\mathbf{15 \pm 6}$
GNNExplainer	0 ± 0	55 ± 0	0.50 ± 0.00	0.65 ± 0.00	80 ± 4	$60 \pm 27^\ddagger$
GraphMask	0 ± 0	64 ± 2	0.09 ± 0.06	0.65 ± 0.00	65 ± 18	$50 \pm 25^\ddagger$
Subgraph search	65 ± 0	$\mathbf{0 \pm 0}$	$\mathbf{0.65 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	74 ± 7	56 ± 25
$\exists^{\geq 3}\exists^{\geq 6} E$ ($N_\cap$=40 of 100)						
Suff. reasons (soft)	$\mathbf{100}^\dagger$	12 ± 0	$\mathbf{0.40 \pm 0.00}$	0.40 ± 0.00	$\mathbf{22 \pm 1}$	$\mathbf{22 \pm 1}$
Suff. reasons (strict)	$\mathbf{100}^\dagger$	$\mathbf{0 \pm 0}$	$\mathbf{0.40 \pm 0.00}$	0.15 ± 0.00	23 ± 1	25 ± 1
GNNExplainer	0 ± 0	10 ± 0	$\mathbf{0.40 \pm 0.00}$	0.40 ± 0.00	83 ± 1	$46 \pm 30^\ddagger$
GraphMask	1 ± 2	31 ± 6	0.22 ± 0.05	0.40 ± 0.00	69 ± 19	$41 \pm 26^\ddagger$
Subgraph search	40 ± 0	$\mathbf{0 \pm 0}$	$\mathbf{0.40 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	86 ± 9	47 ± 32

Table 5: Sufficient reasons under the soft (ε=δ=0.01, m=15,564, R=10) vs. strict test (m=878, R=20), per pattern and class: explanation size, relative size (% of the 65-literal space; mean), and median wall-clock runtime per instance in minutes (same host, comparable batch parallelism); sizes mean±std over instances × five seeds. Both variants are sufficient by construction. Pattern-name symbols as in Table 1.

pattern	share %	soft size	soft rel. %	soft time min	strict size	strict rel. %	strict time min
$\exists^{\geq 3} P^{+\star}$	77.3	$\mathbf{7.3 \pm 0.8}$	**11**	85	14.7 ± 7.0	23	**7**
$\exists^{\geq 3} P^{-}$	22.7	$\mathbf{20.5 \pm 4.8}$	**32**	80	25.1 ± 9.5	39	**6**
$\exists^{\geq 2}\exists^{\geq 3} P^{+\star}$	75.9	$\mathbf{8.0 \pm 0.7}$	**12**	104	16.7 ± 4.8	26	**7**
$\exists^{\geq 2}\exists^{\geq 3} P^{-}$	24.1	$\mathbf{20.5 \pm 6.3}$	**32**	102	29.2 ± 11.4	45	**5**
$\exists^{\geq 3}\neg P^{+}$	77.3	$\mathbf{8.9 \pm 1.3}$	**14**	107	29.4 ± 11.7	45	**5**
$\exists^{\geq 3}\neg P^{-}$	22.7	$\mathbf{14.2 \pm 4.5}$	**22**	98	30.6 ± 10.7	47	**5**
$\exists^{\geq 4}\neg E^{+}$	59.7	$\mathbf{11.3 \pm 5.4}$	**17**	106	28.2 ± 10.4	43	**5**
$\exists^{\geq 4}\neg E^{-}$	40.3	$\mathbf{17.4 \pm 12.2}$	**27**	103	30.3 ± 14.9	47	**5**
$\forall\exists P^{+}$	73.6	$\mathbf{11.1 \pm 0.6}$	**17**	84	15.4 ± 6.8	24	**7**
$\forall\exists P^{-}$	26.4	$\mathbf{11.0 \pm 2.8}$	**17**	71	15.4 ± 7.8	24	**7**
$\exists\forall\neg P^{+}$	34.4	$\mathbf{10.8 \pm 0.5}$	**17**	70	11.0 ± 0.0	**17**	**8**
$\exists\forall\neg P^{-}$	65.6	$\mathbf{11.8 \pm 1.6}$	**18**	86	12.6 ± 2.6	19	**8**
$\exists^{\geq 7} E^{+\star}$	55.7	$\mathbf{7.0 \pm 0.0}$	**11**	74	$\mathbf{7.0 \pm 0.0}$	**11**	**8**
$\exists^{\geq 7} E^{-}$	44.3	15.3 ± 0.5	24	68	$\mathbf{15.1 \pm 0.3}$	**23**	**7**
$\exists^{\geq 3}\exists^{\geq 6} E^{+\star}$	44.7	$\mathbf{14.5 \pm 0.5}$	**22**	98	15.2 ± 0.4	23	**8**
$\exists^{\geq 3}\exists^{\geq 6} E^{-}$	55.3	$\mathbf{14.0 \pm 0.0}$	**22**	94	16.4 ± 0.5	25	**7**

Table 6: Strict-test counterpart of Tab. 1: all methods evaluated under the strict test (m=878; Subgraph-search footprints re-judged accordingly). Symbols and column definitions as there.

pattern	share %	Suff. reasons suff.† %	Suff. reasons size	Suff. reasons rel. %	GNNExplainer suff. %	GNNExplainer size‡	GNNExplainer rel. %	GraphMask suff. %	GraphMask size‡	GraphMask rel. %	Subgraph search suff. %	Subgraph search size	Subgraph search rel. %
$\exists^{\geq 3} P^{+\star}$	77.3	**100**	$\mathbf{14.7 \pm 7.0}$	**23**	1	29.2 ± 10.7	68	0	35.9 ± 6.1	84	71	14.7 ± 8.4	44
$\exists^{\geq 3} P^{-}$	22.7	**100**	$\mathbf{25.1 \pm 9.5}$	**39**	0	–	–	0	–	–	0	–	–
$\exists^{\geq 2}\exists^{\geq 3} P^{+\star}$	75.9	**100**	$\mathbf{16.7 \pm 4.8}$	**26**	6	32.8 ± 8.6	80	0	35.2 ± 6.9	85	98	18.5 ± 5.2	62
$\exists^{\geq 2}\exists^{\geq 3} P^{-}$	24.1	**100**	$\mathbf{29.2 \pm 11.4}$	**45**	0	23.0 ± 0.0	61	0	18.0 ± 0.0	47	0	–	–
$\exists^{\geq 3}\neg P^{+}$	77.3	**100**	$\mathbf{29.4 \pm 11.7}$	**45**	3	32.7 ± 7.1	75	0	37.4 ± 6.5	86	0	–	–
$\exists^{\geq 3}\neg P^{-}$	22.7	**100**	$\mathbf{30.6 \pm 10.7}$	**47**	0	32.8 ± 6.5	75	0	32.9 ± 9.9	78	0	–	–
$\exists^{\geq 4}\neg E^{+}$	59.7	**100**	$\mathbf{28.2 \pm 10.4}$	**43**	0	45.0 ± 0.0	100	0	44.0 ± 0.0	98	0	–	–
$\exists^{\geq 4}\neg E^{-}$	40.3	**100**	$\mathbf{30.3 \pm 14.9}$	**47**	0	38.1 ± 7.2	85	1	36.5 ± 9.7	82	0	–	–
$\forall\exists P^{+}$	73.6	**100**	$\mathbf{15.4 \pm 6.8}$	**24**	0	36.9 ± 4.8	84	0	37.2 ± 6.5	85	9	40.5 ± 3.3	100
$\forall\exists P^{-}$	26.4	**100**	$\mathbf{15.4 \pm 7.8}$	**24**	0	–	–	0	–	–	0	–	–
$\exists\forall\neg P^{+}$	34.4	**100**	$\mathbf{11.0 \pm 0.0}$	**17**	0	–	–	0	–	–	0	–	–
$\exists\forall\neg P^{-}$	65.6	**100**	$\mathbf{12.6 \pm 2.6}$	**19**	0	34.9 ± 6.1	76	0	41.6 ± 6.5	84	12	43.0 ± 0.0	100
$\exists^{\geq 7} E^{+\star}$	55.7	**100**	$\mathbf{7.0 \pm 0.0}$	**11**	0	39.1 ± 2.2	84	0	33.3 ± 8.3	72	**100**	27.4 ± 2.3	74
$\exists^{\geq 7} E^{-}$	44.3	**100**	$\mathbf{15.1 \pm 0.3}$	**23**	0	–	–	0	–	–	0	–	–
$\exists^{\geq 3}\exists^{\geq 6} E^{+\star}$	44.7	**100**	$\mathbf{15.2 \pm 0.4}$	**23**	0	40.1 ± 1.3	89	0	35.7 ± 7.4	79	**100**	30.1 ± 2.7	86
$\exists^{\geq 3}\exists^{\geq 6} E^{-}$	55.3	**100**	$\mathbf{16.4 \pm 0.5}$	**25**	0	–	–	0	–	–	0	–	–

Table 7: Degenerate outcomes of the Subgraph search per pattern and class: share (%) of N instances where the greedy collapses to a single node (its minimum; never ε-sufficient) and where it stalls on the untouched input graph. Pattern names as in Tab. 1.

pattern	N	single-node (%)	full graph (%)
$\exists^{\geq 3} P^{+\star}$	76	0	11
$\exists^{\geq 3} P^{-}$	74	62	9
$\exists^{\geq 2}\exists^{\geq 3} P^{+\star}$	86	0	6
$\exists^{\geq 2}\exists^{\geq 3} P^{-}$	64	61	0
$\exists^{\geq 3}\neg P^{+}$	72	0	10
$\exists^{\geq 3}\neg P^{-}$	78	14	3
$\exists^{\geq 4}\neg E^{+}$	69	1	3
$\exists^{\geq 4}\neg E^{-}$	81	0	32
$\forall\exists P^{+}$	64	0	11
$\forall\exists P^{-}$	61	66	5
$\exists\forall\neg P^{+}$	75	1	0
$\exists\forall\neg P^{-}$	25	4	16
$\exists^{\geq 7} E^{+\star}$	65	0	0
$\exists^{\geq 7} E^{-}$	35	100	0
$\exists^{\geq 3}\exists^{\geq 6} E^{+\star}$	40	0	12
$\exists^{\geq 3}\exists^{\geq 6} E^{-}$	60	100	0
pooled	1025	23	7