PARSE: PARTITION-BASED SHAPLEY EXPLANATION OF GRAPH NEURAL NETWORKS WITH PAIRWISE INTERACTIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Graph Neural Networks (GNNs) achieve strong predictive performance but remain difficult to interpret. Most explainers score single nodes or edges, which shows *where* a model looks but not *how* substructures act together. We propose PARSE (PARtition-based Shapley Explanation), which explains a GNN by partitioning the graph into connected substructures of interacting nodes and measuring the pairwise interactions (synergy or redundancy) between them. PARSE is grounded in cooperative game theory: from one set of random node orderings, it estimates, separately, each node's own contribution (its Shapley value) and the Shapley interaction index of every pair, which guide the partition and a search for the node sets of highest fidelity. On controlled tasks whose known interaction is synergy, redundancy or exclusivity, PARSE's pairwise estimates between the two planted substructures have the sign of the planted interaction on every test graph that contains both, and its group ranking ranks first a pair of groups that holds both substructures on 37 to 41 of the 46 such graphs per task. On the test graphs of thirteen public datasets, against eleven explainers, PARSE is the most faithful method in 58 of 65 (dataset, metric) comparisons.

1 INTRODUCTION

A graph neural network (GNN) classifier often bases a prediction on how parts of a graph act together. Two substructures may be needed jointly (synergy), either may suffice (redundancy), or the class may flip when both are present (exclusivity). In a molecule, a functional group may affect a property only in the presence of another; in a sentence, a negation reverses the sentiment of the words it modifies. Node-importance scores show *where* a model looks, not *how* the parts combine. The Shapley value (Shapley, 1953) credits each node with its average effect over all contexts. Under exclusivity, where either substructure alone changes the prediction and the two together restore it, their individual contributions can largely cancel, although together they decide the prediction. The Shapley interaction index (SII) of Grabisch & Roubens (1999), which measures what a pair adds beyond its two separate effects, is large in exactly this case. Keeping each node's own contribution apart from its pairwise interactions therefore reveals such interacting parts. GraphSHAP-IQ (Muschalik et al., 2025) computes such interactions for GNNs by exploiting the model's architecture, and MAGE (Bui et al., 2024) uses a structure-aware interaction index. Structure-aware explainers build connectivity into the explanation or its value: SubgraphX (Yuan et al., 2021) returns one connected subgraph, and GStarX (Zhang et al., 2022), GraphEXT (Wu et al., 2025) and MAGE value a disconnected node set through its connected pieces, so two distant substructures can act together only through the nodes that join them. What is still needed is a way to estimate pairwise interactions for any GNN from model queries alone, at a feasible cost, and to turn them into explanations that are faithful to the model. Evaluating them is a further challenge: real graphs rarely come with ground truth for interactions, and planted ground truth need not be what a model actually uses (Faber et al., 2021), so both faithfulness and interaction recovery must be measured.

We propose PARSE (Figure 1). It treats the nodes as players in a cooperative game in which the payoff of a node set is the model's probability for its predicted class when all other nodes are masked, and it explains a prediction with connected groups of interacting nodes, a *partition*, and with node sets of high fidelity (how much the prediction depends on the set). Because every node set, connected

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.

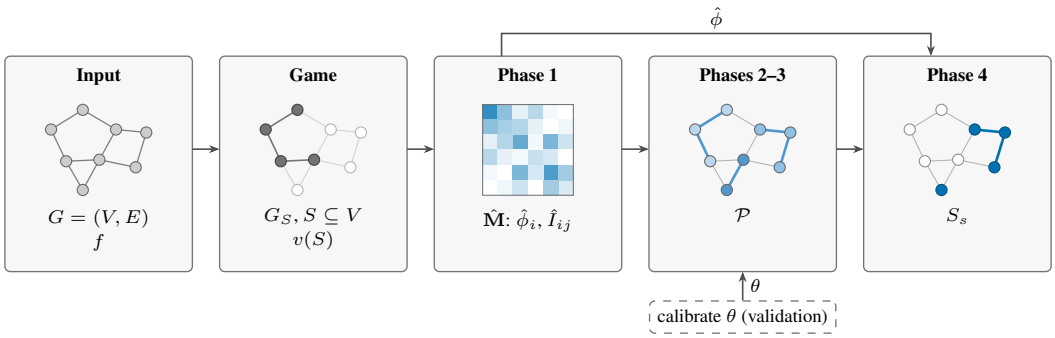


Figure 1: **Overview of PARSE.** White nodes are masked in G_S. Phase 1 estimates node values $\hat{\phi}_i$ and pair interactions $\hat{I}_{ij}$; Phases 2–3 group the nodes into a connected partition $\mathcal{P}$ (shades); Phase 4 returns one node set S_s per explanation size (blue).

or not, is scored as one masked graph of the explained model, PARSE needs only model queries, applies to any GNN, and relates substructures however far apart they lie. Our contributions are:

1. An interaction-aware explanation for any GNN from model queries alone: PARSE estimates the contribution of each node and, separately, the interaction of each pair of nodes, so it can find parts whose individual contributions cancel; the sign of the interaction between two of its groups shows whether the substructures act jointly (positive) or interchangeably (negative) for the predicted class (Section 4).
2. Node and pair values from the same orderings: the random node orderings of permutation sampling for the Shapley value (Castro et al., 2009) also give the SII of every pair, from the pairs that are consecutive in an ordering, at about twice the model queries of the node values alone (Section 4.2); the node values sum exactly to the change in the predicted-class probability from the fully masked graph to the full graph.
3. Connected interaction groups: from these values alone, PARSE splits the graph into connected groups of interacting nodes (Section 4.3); a group ranking, which estimates the SII of every pair of groups, shows which parts of the graph act together, even far apart.
4. Fidelity-driven explanations: for each explanation size, a search returns the node set of highest H-Fidelity that it finds (Section 4.4).
5. Evaluation (Sections 5–6): a public benchmark of thirteen datasets measures how faithful PARSE's explanations are against eleven explainers, and controlled tasks with two planted motifs check whether PARSE detects their synergy, redundancy or exclusivity, even when the motifs lie on opposite sides of the graph.

2 Related work

GNN explainers are grouped by how they derive importance from the model (Yuan et al., 2023). Gradient and decomposition methods trace the prediction back to the input, as in Grad×Input (Shrikumar et al., 2016), class activation maps (Pope et al., 2019) and relevance propagation over walks (Schnake et al., 2022). Perturbation methods learn a mask that preserves the prediction: GNNExplainer (Ying et al., 2019) optimizes one per instance to maximize the mutual information between the prediction and the masked subgraph, with size and entropy regularizers, and PGExplainer (Luo et al., 2020) trains one shared network that predicts edge masks from node embeddings. Surrogate methods fit an interpretable model to the predictions on samples around the input, such as a probabilistic graphical model (Vu & Thai, 2020), a nonlinear feature-selection model on a node's neighbourhood (Huang et al., 2023) or, in GraphSVX (Duval & Malliaros, 2021), a weighted linear model whose coefficients estimate Shapley values. Other explainers generate explanatory subgraphs (Lin et al., 2021), synthesize graph patterns that explain the model as a whole (Yuan et al., 2020), or seek counterfactual explanations, small parts whose removal changes the prediction (Lucic et al., 2022), possibly combined with factual ones that alone preserve it (Tan et al., 2022).

Shapley-based explanation treats input parts as players in a cooperative game. The Shapley value (Shapley, 1953) is the unique division of the total payoff that satisfies efficiency, symmetry, the null-player property and additivity; SHAP (Lundberg & Lee, 2017) made it a standard feature attribution, and its model-agnostic Kernel SHAP estimates it by weighted linear regression, while permutation sampling averages marginal contributions over random orderings (Castro et al., 2009; Štrumbelj & Kononenko, 2014). On graphs, several methods build graph structure into the game. SubgraphX (Yuan et al., 2021) searches connected subgraphs by Monte Carlo tree search, scoring each as one player in a Shapley game over its L-hop neighbourhood for an L-layer GNN, and SAME (Ye et al., 2023) scores combinations of several connected pieces found by expansion-based search. GStarX (Zhang et al., 2022) scores nodes with the Hamiache–Navarro value, which shares the surplus of cooperation between neighbours, and GraphEXT (Wu et al., 2025) values each node by its marginal contribution as it moves between coalitions, treating structure as an externality.

Interaction indices extend the Shapley value to sets of players. The SII (Grabisch & Roubens, 1999) averages the joint effect of a set over all contexts without dividing the total payoff, whereas the Shapley–Taylor index (Sundararajan et al., 2020), Faith-Shap (Tsai et al., 2023) and the k-SII (Bordt & von Luxburg, 2023) divide it among interactions up to a chosen order, and SHAP-IQ (Fumagalli et al., 2023) estimates interactions of any order from sampled coalitions. For GNNs, GraphSHAP-IQ (Muschalik et al., 2025) computes exact Shapley interactions for models with a linear readout, whose interactions stay within receptive fields; MAGE (Bui et al., 2024) uses the second-order Myerson–Taylor index, where nodes interact only through coalitions connecting them, to extract the motifs that most raise or lower the prediction; and GECo (Amato et al., 2026) splits the graph into communities by greedy modularity maximization, feeds each to the model and keeps those above a threshold such as the mean. Beyond explanation, graph partitions mainly serve computation, such as initializing network embeddings or computing them in parallel (Lin et al., 2020; Lin, 2021).

Several of these explainers return groups of nodes rather than single nodes: SubgraphX and SAME return connected subgraphs, MAGE returns motifs whose nodes interact through connecting coalitions, and GECo returns communities of the graph. In each case the groups follow the graph's connectivity, and the explanation reports how much each group matters, not how two groups act together.

Because most real graphs lack ground-truth explanations, explainers are compared mainly by fidelity (Yuan et al., 2023), the change in prediction when the explanation is removed or kept alone, and benchmarks standardize them (Amara et al., 2022; Agarwal et al., 2023). Planted motifs serve as ground truth only when the model must use them (Faber et al., 2021), and robust fidelity (Zheng et al., 2024) aims to withstand the distribution shift of masked graphs. As with removal-based explanations (Covert et al., 2021), fidelity is defined by how nodes are removed.

3 Background

A graph $G = (V, E)$ has $n = |V|$ nodes with features $\mathbf{x}_i$; a trained GNN f gives class logits $f_c(G)$.

Definition 1 (Cooperative game and second difference)**.** A cooperative game on the players V assigns a payoff $v(S) \in \mathbb{R}$ to every coalition $S \subseteq V$. For distinct $i, j \in V$ and $T \subseteq V \setminus \{i, j\}$, the second difference of v at T is $\Delta_{ij} v(T) = v(T \cup \{i, j\}) - v(T \cup \{i\}) - v(T \cup \{j\}) + v(T)$.

The second difference $\Delta_{ij} v(T)$ is positive when i and j add more together than apart (synergy), and negative when each makes the other less useful (redundancy).

Definition 2 (Shapley value and Shapley interaction index)**.** The Shapley value of player i (Shapley, 1953) and the Shapley interaction index (SII) of the pair $\{i, j\}$ (Grabisch & Roubens, 1999) are

$$\phi_i = \sum_{T \subseteq V \setminus \{i\}} \frac{|T|!\,(n - |T| - 1)!}{n!} \big[v(T \cup \{i\}) - v(T) \big], \tag{1}$$

$$I_{ij} = \sum_{T \subseteq V \setminus \{i,j\}} \frac{|T|!\,(n - |T| - 2)!}{(n-1)!}\, \Delta_{ij} v(T). \tag{2}$$

The Shapley value ϕ_i is the expected gain from adding i to the players before it in a uniformly random ordering of V, and the values sum to the total change in payoff, $\sum_{i \in V} \phi_i = v(V) - v(\emptyset)$. The SII has no such identity: I_{ij} measures how two nodes interact, not a share of the prediction.

4 METHODOLOGY

PARSE runs in four phases (Figure 1 and Algorithm 1). Phase 1 estimates each node's own contribution and, separately, the interaction of every pair; Phases 2–3 group interacting nodes into a connected partition; Phase 4 searches, for each explanation size, for the node set of highest H-Fidelity.

4.1 THE GAME

The players are the nodes. The explained class is the model's prediction on the full graph, $c^\star = \arg\max_c f_c(G)$. For a set $S \subseteq V$ of kept nodes, the masked graph G_S gives every node outside S the training-mean features $\bar{\mathbf{x}}$ and removes its edges, leaving it in the graph as an isolated node. The payoff is $v(S) = \mathrm{softmax}(f(G_S))_{c^\star}$, and $v(V) - v(\emptyset)$ is the change in this probability from the fully masked graph to the full graph. Every set, connected or not, is scored as one masked graph.

4.2 PHASE 1: SHAPLEY VALUES AND PAIRWISE INTERACTIONS

PARSE draws m independent, uniformly random orderings $\pi_1, \ldots, \pi_m$ of V ($m = 50$), as in permutation sampling of the Shapley value (Castro et al., 2009), and reads the pair interactions from the same orderings (Tsai et al., 2023). In ordering π_r, let $\mathrm{Pre}_r(i)$ be the nodes before i, let $A_r(i,j) = 1$ if i and j are consecutive and 0 otherwise, and let $T_r(i,j)$ be the nodes before that consecutive pair. The estimates are

$$\hat{\phi}_i = \frac{1}{m} \sum_{r=1}^{m} \Big[v\big(\mathrm{Pre}_r(i) \cup \{i\}\big) - v\big(\mathrm{Pre}_r(i)\big) \Big], \tag{3}$$

$$\hat{I}_{ij} = \frac{n}{2m} \sum_{r=1}^{m} A_r(i,j)\, \Delta_{ij} v\big(T_r(i,j)\big), \tag{4}$$

with $\Delta_{ij} v$ the second difference of Definition 1. They form the symmetric *interaction matrix* $\hat{\mathbf{M}}$ with $\hat{M}_{ii} = \hat{\phi}_i$ and $\hat{M}_{ij} = \hat{I}_{ij}$ ($i \neq j$). The diagonal holds each node's own contribution and the off-diagonal its pairwise interactions, and PARSE keeps the two apart. A Shapley value averages a node's effect over all contexts, so nodes that matter only in combination can have individual contributions that largely cancel while their interactions are large; the partition (Section 4.3) groups such nodes by their interactions, and the extended node ranking (Section 4.4) credits their pair and group effects back to them. A pair is consecutive in one ordering with probability $2/n$, so the factor $n/(2m)$ divides the pair's sum by its expected count $2m/n$. Along each ordering, the node gains add up to $v(V) - v(\emptyset)$, so for every draw $\sum_{i \in V} \hat{\phi}_i = v(V) - v(\emptyset)$. The node values need the $n + 1$ prefixes of each ordering and the pair values $n - 1$ further masked graphs (each prefix with its last node replaced), so Phase 1 evaluates at most $2 + 2m(n-1)$ masked graphs, about twice the $2 + m(n-1)$ of the node values alone.

4.3 PHASES 2–3: A CONNECTED INTERACTION PARTITION

Phases 2–3 turn $\hat{\mathbf{M}}$ into connected groups of interacting nodes. A partition $\mathcal{P} = \{C_1, \ldots, C_K\}$ of V is *connected* if every group induces a connected subgraph, and two sums of $\hat{\mathbf{M}}$ score it: the *group evidence* of a group and the *cross evidence* of any two groups, adjacent or not,

$$g(C) = \sum_{i \in C} \hat{M}_{ii} + \sum_{i,i' \in C,\ i<i'} \hat{M}_{ii'}, \qquad g(C, C') = \sum_{i \in C} \sum_{i' \in C'} \hat{M}_{ii'}. \tag{5}$$

Every group with $g(C) > 0$ and every pair of groups with $g(C, C') > 0$ is an *evidence unit*, with value a (its evidence) and size b ($|C|$ or $|C| + |C'|$). The Λ units are sorted so that $a_1 \geq \cdots \geq a_\Lambda > 0$,

Algorithm 1 PARSE on one graph. $\mathrm{Pre}_r(i)$: the nodes before i in ordering π_r.

Require: GNN f; graph $G = (V, E)$ with $n = |V|$; calibrated θ; fixed m, N_{SA}, τ_0, κ, $\mathcal{S}$, N_{ex}
Ensure: node sets S_s ($s \in \mathcal{S}$); node values $\hat{\phi}$; connected partition $\mathcal{P}$
$c^\star \leftarrow \arg\max_c f_c(G)$; $v(S) = \mathrm{softmax}(f(G_S))_{c^\star}$ ▷ Section 4.1
for $r = 1, \ldots, m$ **do** ▷ Phase 1
  draw a uniformly random ordering π_r of V
  $\hat{\phi}_i \leftarrow \hat{\phi}_i + v(\mathrm{Pre}_r(i) \cup \{i\}) - v(\mathrm{Pre}_r(i))$ for every node i ▷ Castro et al. (2009)
  $\hat{I}_{ij} \leftarrow \hat{I}_{ij} + \Delta_{ij} v(T)$ for each consecutive pair i, j; T: the nodes before them ▷ Tsai et al. (2023)
end for
$\hat{\phi} \leftarrow \hat{\phi}/m$; $\hat{I} \leftarrow \frac{n}{2m}\hat{I}$; $\hat{M}_{ii} \leftarrow \hat{\phi}_i$, $\hat{M}_{ij} \leftarrow \hat{I}_{ij}$ ▷ Eqs. 3–4
$\mathcal{P} \leftarrow$ single nodes; merge edge-joined groups by largest aff down to k_0 if possible ▷ Phases 2–3
$\mathcal{P} \leftarrow$ best partition visited by N_{SA} annealing steps (move, merge, detach) on Score ▷ Eq. 7
for $s \in \mathcal{S}$ **do** ▷ Phase 4
  if $\binom{n}{k_s} \le N_{\mathrm{ex}}$: $S_s \leftarrow$ the k_s-node subset of V with the highest HFid ▷ Eq. 8
  else: $S_s \leftarrow$ the highest-HFid set reached by swaps from four starts (the top k_s by $|\hat{\phi}_i|$, a set grown along edges, the partition start, a greedy set)
end for
return $\{S_s\}_{s \in \mathcal{S}}$, $\hat{\phi}$, $\mathcal{P}$

and the preferred group size is $\mu = \mu_c\, n^{\mu_\beta}$. A good partition concentrates its positive evidence in a few units of about the preferred size; the first $L \le \Lambda$ units score

$$R_L = \underbrace{\sum_{l \le L} \frac{a_l}{l^{\,p_{\mathrm{cov}}}}}_{\mathrm{Cov}_L} - \gamma_H \underbrace{\Big(-\sum_{l \le L} q_l \log_2 q_l\Big)}_{\mathcal{H}_L} - \gamma_S \underbrace{\frac{1}{L}\sum_{l \le L} \big(\ln b_l - \ln \mu\big)^2}_{\mathrm{Pen}_L}, \qquad q_l = \frac{a_l}{\sum_{l' \le L} a_{l'}}, \tag{6}$$

and the partition scores

$$\mathrm{Score}(\mathcal{P}) = \max\Big(0,\ \max_{L \le \Lambda} R_L\Big). \tag{7}$$

The coverage Cov_L rewards strong units, the entropy $\mathcal{H}_L$ penalizes value spread over many units, and the size cost Pen_L penalizes units far from the preferred size. The settings $\theta = (\gamma_H, \gamma_S, p_{\mathrm{cov}}, \mu_c, \mu_\beta)$ are calibrated (Section 4.5).

The partition search uses $\hat{\mathbf{M}}$ alone and queries no model. Phase 2 starts from single nodes and repeatedly merges the two groups C, C' joined by an edge that have the largest affinity $\mathrm{aff}(C, C') = \sum_{i \in C} \sum_{i' \in C'} |\hat{M}_{ii'}|$, until $k_0 = \max(1, \lceil n/\mu \rceil)$ groups remain or no two groups are joined by an edge. Phase 3 refines this partition by simulated annealing (Kirkpatrick et al., 1983), which makes N_{SA} proposals: move a node that has a neighbour in another group into that group, merge two groups joined by an edge, or detach a node as a new single-node group; groups that become disconnected are split. At step $t = 0, \ldots, N_{\mathrm{SA}} - 1$, it accepts a proposal that changes Score by Δ with probability $\min(1, e^{\Delta/\tau_t})$, where $\tau_t = \tau_0 \kappa^t$, with initial temperature τ_0 and cooling rate κ, and keeps the best partition visited.

4.4 Phase 4: explanation sets and outputs

The explanation is the output of one search that maximizes H-Fidelity; the partition supplies one of its starting sets. For a sparsity s (the fraction of nodes left out) from a set $\mathcal{S}$ of sparsities, PARSE returns one set of exactly $k_s = \min\big(n, \max(1, \lfloor (1-s)\, n + \frac{1}{2} \rfloor)\big)$ nodes, its *budget*. Following the standard fidelity metrics (Yuan et al., 2023; Amara et al., 2022), the *necessity* of a set S is the probability drop when the nodes of S are masked (higher is better), and its signed *sufficiency error* is the drop when all other nodes are masked (closer to zero is better): $\mathrm{Fid}^+(S) = v(V) - v(V \setminus S)$ and $\mathrm{Fid}^-(S) = v(V) - v(S)$. PARSE's set search maximizes their combination, the harmonic fidelity (H-Fidelity) of Zhang et al. (2022),

$$\mathrm{HFid}(S) = \frac{(1 + h_+)(1 - h_-)}{2 + h_+ - h_-}, \tag{8}$$

with the size-weighted terms $h_+ = \mathrm{Fid}^+(S)\,(1 - |S|/n)$ and $h_- = \mathrm{Fid}^-(S)\,|S|/n$. If the number of candidate sets $\binom{n}{k_s}$ is at most a threshold N_{ex}, the search tries every subset. Otherwise it is a

multi-start local search: from each of four starting sets it repeatedly swaps one node in and one node out to raise HFid, and it returns the set of highest HFid that it reaches. The starting sets are the k_s nodes of largest $|\hat{\phi}_i|$, a set grown along edges by $|\hat{\phi}_i|$, the partition start, and a set built greedily on HFid. The *partition start* fills the budget with whole groups of $\mathcal{P}$, first the groups of the first $L^\star$ units, where $L^\star$ is the shortest L that maximizes R_L (Eq. 6), then the other groups by decreasing $|g(C)|$; from the first group that does not fit, it takes only the nodes of largest $|\hat{M}_{ii}|$. Budgets are searched separately; sets need not be connected or nested.

PARSE thus returns three outputs: (a) the sets S_s, its explanation; (b) the node ranking $\hat{\phi}$; and (c) the partition $\mathcal{P}$, which shows the connected groups of interacting nodes and supplies one start of the set search. Two further outputs build on them to read the interactions at the level of groups and nodes. The *group ranking* re-runs the partition search on the same $\hat{\mathbf{M}}$ with its own settings, splits every group above a size cap into connected pieces, and treats the resulting groups as the players of a game. It estimates the SII $\hat{I}(C, C')$ of every pair of groups, with standard error $\widehat{\mathrm{se}}(C, C')$, and ranks the pairs by $r(C, C') = \max\big(0, |\hat{I}(C, C')| - \widehat{\mathrm{se}}(C, C')\big)$. The *extended node ranking* is a weighted sum of $|\hat{\phi}|$ and of node, pair and group effects estimated again on random subsets of the other nodes, with each pair or group effect credited to its nodes, so it can rank first nodes whose own Shapley values cancel. Section 5.2 evaluates both rankings.

4.5 Per-dataset calibration on validation graphs

For the explanation sets, θ is calibrated per dataset and all other settings are shared across datasets. The calibration objective averages fidelity metrics over sparsities and a set $\mathcal{D}_{\mathrm{cal}}$ of validation graphs,

$$J(\theta) = \frac{1}{|\mathcal{D}_{\mathrm{cal}}|} \sum_{G \in \mathcal{D}_{\mathrm{cal}}} \frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} \tfrac{1}{4}\big(\mathrm{HFid} + \mathrm{Fid}^{+} + \mathrm{Char} - |\mathrm{Fid}^{-}|\big)(G, S_s), \tag{9}$$

where the characterization (Amara et al., 2022) is the harmonic mean $\mathrm{Char}(S) = 2\rho_+\rho_-/(\rho_+ + \rho_-)$ of $\rho_+ = \max(0, \mathrm{Fid}^{+}(S))$ and $\rho_- = 1 - \max(0, \mathrm{Fid}^{-}(S))$. PARSE maximizes J by Bayesian optimization with Gaussian-process expected improvement (Jones et al., 1998) and adopts the best setting found. Appendix A lists the fixed settings, the search ranges and each dataset's adopted setting.

5 Experimental setup

We evaluate PARSE in two settings: a public benchmark that measures how faithfully each method's node sets explain a trained model, and controlled tasks whose interactions are known.

5.1 Public benchmark

We use thirteen graph-classification datasets, each split at random into 80% training, 10% validation and 10% test graphs. They are four molecule sets, BACE and BBBP (Wu et al., 2018), Benzene (Sanchez-Lengeling et al., 2020; Agarwal et al., 2023) and Mutagenicity (Kazius et al., 2005; Morris et al., 2020); two protein sets, ENZYMES and PROTEINS (Morris et al., 2020); three sentiment-graph sets, Graph-SST2, Graph-SST5 and Graph-Twitter (Socher et al., 2013; Yuan et al., 2023; Liu et al., 2021); and four synthetic sets of Barabási–Albert graphs with planted motifs: BA-2Motifs (Luo et al., 2020), with a house or a five-node cycle, and, following Bui et al. (2024), BA-House-Grid, BA-House-And-Grid and BA-House-Or-Grid, with house and grid motifs. Each model f is a three-layer GCN (Kipf & Welling, 2017) or GIN (Xu et al., 2019) with mean pooling, chosen per dataset on validation data; Appendix B describes each dataset, its split and how its model is trained and selected. We compare PARSE with eleven explainers from Section 2: Grad×Input (Shrikumar et al., 2016), GNNExplainer (Ying et al., 2019), PGExplainer (Luo et al., 2020), GraphSVX (Duval & Malliaros, 2021), SAME (Ye et al., 2023), GraphEXT (Wu et al., 2025), GStarX (Zhang et al., 2022), GraphSHAP-IQ (Muschalik et al., 2025), MAGE (Bui et al., 2024), SubgraphX (Yuan et al., 2021) and GECo (Amato et al., 2026). Each baseline runs in its authors' released code or a public implementation with its authors' settings, and Grad×Input is implemented directly; Appendix C details each implementation and its settings. Each output becomes one node set per budget: its top-scored nodes, or its returned subgraph completed to the budget.

All methods are scored in the game of Section 4.1, with the same explained class and masker (*training-mean masking*), on one set of k_s nodes (Section 4.4) for each of the five sparsities $s \in \{0.5, 0.6, 0.7, 0.8, 0.9\}$. We report five metrics. Four are defined in Section 4: H-Fidelity (Eq. 8), Fid^+ and the signed Fid^- (Section 4.4), and characterization (Section 4.5). The fifth, Fid_α, is a robust fidelity in the spirit of Zheng et al. (2024), adapted to node masking and computed on the set at $s = 0.5$: its necessity side masks each explanation node with probability $\alpha = 0.7$ and each other node with probability $1 - \alpha$, its sufficiency side swaps the two probabilities, and Fid_α is the necessity-side probability drop minus the sufficiency-side drop, each averaged over random masks (higher is better). Each metric is averaged over budgets within a graph and then over graphs; for Fid^-, the absolute value of this mean is compared (closer to zero is better). For each dataset and metric, we compare PARSE with each baseline on the same test graphs, a standard paired design (Demšar, 2006); a difference within one standard error of the paired per-graph differences counts as a tie. PARSE *wins* if it beats every baseline and *ties for best* if no baseline beats it but at least one ties. Every method is run on a random draw of each dataset's test graphs. Appendix D details the test draw, the time and memory limits shared by all methods and the statistical tests.

5.2 Controlled tasks

To test whether PARSE finds the right interaction, we built three synthetic tasks whose label is a known rule over two planted motifs. Each graph is a small tree (the *scaffold*) with a 7-node attachment on each of two leaves that are as far apart as possible; each attachment contains a triangle (motif A), a square (motif B) or a hexagon (a decoy), and all nodes have the same feature. The motifs are far apart, so their interaction links two distant parts of the graph: any connected node set that holds both covers more than half of the graph. Let $a, b \in \{0, 1\}$ indicate whether motifs A and B are present. The three labels are $a \wedge b$, where the motifs count only together (synergy; we call this task AND); $a \vee b$, where either motif suffices (redundancy; OR); and $a \oplus b$, where exactly one motif must be present (exclusivity; XOR). On a graph with both motifs, let $c^\star$ be the class the predictor assigns, and let $y(a, b) = 1$ if the rule gives class $c^\star$ for that presence state and 0 otherwise. The interaction between A and B should then have the sign

$$\sigma^\star = \mathrm{sign}\big(y(1,1) - y(1,0) - y(0,1) + y(0,0)\big), \tag{10}$$

which is $+$ for AND and $-$ for OR (both explained as class 1), and $+$ for XOR (explained as class 0): there, the two motifs together restore class 0, so exclusivity acts as synergy toward that class. The graphs built on one tree form a *scaffold family*; whole families are assigned to training, validation or test, which leaves 46 test graphs with both motifs in each task. Each predictor is a GIN-type network (Xu et al., 2019) with random-walk structural encoding (Dwivedi et al., 2022), chosen on validation graphs, and it classifies every test graph correctly.

PARSE runs as in Section 4, except that a masked node gets feature 0 and loses its edges, since all nodes share one feature. We score its group ranking and its extended node ranking (Section 4.4), whose settings are calibrated on the validation families with objectives based on the known motifs, and evaluate them on the test families. A pair of groups is *correct* if its rank score (Section 4.4) is positive and its estimated SII has sign $\sigma^\star$. Its *overlap* is the smaller of two intersection-over-union (IoU) values: one group with A and the other with B. A correct pair is a *soft* match if its overlap is at least 0.5 and a *strict* match if, in addition, one group contains all of A and the other all of B; top-1 asks whether the highest-ranked pair is a match. The *motif sign* tests the Phase-1 pair estimates alone: does $\sum_{i \in A,\, j \in B} \hat{M}_{ij}$, the summed pairwise SII between the nodes of the two motifs, have sign $\sigma^\star$? Node AUC is the area under the ROC curve of the extended node ranking against the motif nodes, and the motif nodes are *ranked first* on a graph when all of them precede every other node. Appendix F details how the tasks are built and scored.

6 Results

6.1 Public benchmark

PARSE is the most faithful method in 58 of the 65 (dataset, metric) comparisons, ties for best in 3 and is beaten in 4 (Table 1). It is best on H-Fidelity, Fid^+ and characterization on every dataset, and 51 of its 58 wins, including all 39 on these metrics, are significant against the closest baseline

Table 1: Outcome per dataset and metric. ✓: PARSE beats every baseline by more than one paired standard error; tie: no baseline beats PARSE by that much, and the named one is within it; italics: the baseline that beats PARSE. H-Fid.: H-Fidelity; Char.: characterization.

	Dataset	H-Fid.	Fid^+	Fid^-	Char.	Fid_α
Real-world	BACE	✓	✓	*GraphEXT*	✓	✓
	BBBP	✓	✓	*GECo*	✓	✓
	Benzene	✓	✓	✓	✓	✓
	ENZYMES	✓	✓	✓	✓	tie: GraphEXT
	Graph-SST2	✓	✓	tie: SAME	✓	✓
	Graph-SST5	✓	✓	*SAME*	✓	✓
	Graph-Twitter	✓	✓	✓	✓	✓
	Mutagenicity	✓	✓	✓	✓	✓
	PROTEINS	✓	✓	✓	✓	✓
Synthetic	BA-2Motifs	✓	✓	✓	✓	✓
	BA-House-And-Grid	✓	✓	tie: GraphSHAP-IQ	✓	✓
	BA-House-Grid	✓	✓	✓	✓	✓
	BA-House-Or-Grid	✓	✓	✓	✓	*GraphSHAP-IQ*
	PARSE best	13/13	13/13	8/13 (+2 ties)	13/13	11/13 (+1 tie)

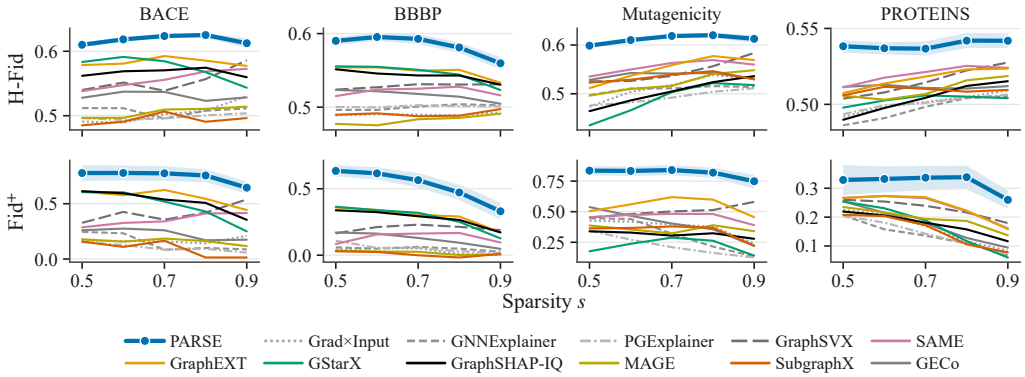


Figure 2: Mean H-Fidelity (top) and Fid^+ (bottom) against sparsity on the four datasets where PARSE leads most clearly; Appendix E.2 shows all thirteen. Shaded: PARSE's 95% bootstrap interval.

under a Wilcoxon signed-rank test (Wilcoxon, 1945) with Holm's correction (Holm, 1979). Its closest competitors are game-theoretic explainers: in 64 of the 65 comparisons the closest baseline is one of them, most often GraphEXT or GraphSHAP-IQ. The margins are largest on Mutagenicity, BBBP and BACE, where PARSE's Fid^+ exceeds the closest baseline's by 0.27, 0.23 and 0.19 and its characterization by 0.30, 0.26 and 0.18. On Fid_α, which enters neither PARSE's calibration objective nor its search, PARSE is best on 11 of 13 datasets and ties for best on one more.

PARSE's lead holds across explanation sizes: its mean H-Fidelity is the highest of all methods at every sparsity on all thirteen datasets, ahead of the best baseline by 0.012 to 0.020 on average over the datasets and by up to 0.063 (Mutagenicity, $s = 0.5$), and its mean Fid^+ is the highest on eleven (Figure 2 shows four of them). It also holds on small and large graphs alike: its H-Fidelity margin over the closest baseline has a 95% bootstrap interval above zero on the smallest, middle and largest third of every dataset's test graphs. Re-scored under three other maskers (zero features, or masked nodes that keep their edges), its node sets keep the best mean Fid^+ and $|\mathrm{Fid}^-|$ with zero features and stay in the top two under all three. The lead is visible on single graphs: on the Mutagenicity test graph with the median H-Fidelity margin, removing PARSE's eight explanation nodes lowers the predicted-class probability by 0.958, against at most 0.865 for any baseline. Once calibrated, a one-off step of about 1.4 hours per dataset, PARSE explains a graph in about 2.5 s, less than the 9

Table 2: Controlled results on the 46 test graphs with both motifs per task. Motif sign: Phase-1 pair estimates; strict and soft top-1: group ranking; node AUC and motif nodes first: extended node ranking.

Task	Motif sign	Strict top-1	Soft top-1	Node AUC	Motif nodes first
AND (synergy)	46/46	37/46	37/46	1.000	46/46
OR (redundancy)	46/46	37/46	40/46	1.000	46/46
XOR (exclusivity)	46/46	41/46	43/46	0.999	44/46

to 57 s per graph of SubgraphX, GraphEXT, GStarX, SAME and MAGE. Appendix E gives every number behind these results, including the curves of all thirteen datasets and example explanations.

6.2 Controlled tasks

PARSE's Phase-1 pair estimates recover the sign of the planted interaction on every test graph with both motifs in all three tasks (motif sign in Table 2), including XOR, where either motif alone flips the explained class and the two together restore it; there, the summed pairwise SII between the two motifs averages $+4.20$. At node level, the extended node ranking, which adds pair and group effects to $|\hat{\phi}_i|$, pinpoints the motifs (node AUC near 1) and places exactly the motif nodes first on 46, 46 and 44 of the 46 graphs of AND, OR and XOR. At group level, the group ranking locates the interacting regions: on 37, 37 and 41 of the 46 graphs of AND, OR and XOR (strict top-1 in Table 2), a pair of groups holds both whole motifs, and that pair is ranked first among a median of 36 to 45 candidate pairs. When only one motif is present, it alone decides the class under OR and XOR, and PARSE ranks its nodes first on every such graph.

PARSE detects interacting parts of a graph, here two motifs on opposite sides of the graph, 6 to 10 hops apart, because it scores any node set as one masked graph and its SII, partition and group ranking cover every pair of nodes or groups, adjacent or not: whenever its top-ranked pair of groups holds both whole motifs, the two groups lie 2 to 10 hops apart.

7 Conclusion

We presented PARSE, an interaction-aware explainer for graph neural networks. From one set of random node orderings, PARSE estimates, separately, the Shapley value of every node and the Shapley interaction index of every pair, groups the graph into connected parts that interact, and searches for the node sets of highest H-Fidelity. On controlled tasks with known synergy, redundancy and exclusivity, PARSE's pair estimates between the two motifs had the sign of the planted interaction on every test graph with both motifs, a distinction that Shapley values alone cannot express, and its group ranking ranked first a pair of groups holding both motifs on 37 to 41 of the 46 such graphs per task.

Because PARSE partitions the graph into groups and scores the interaction of every pair of groups, it can find two disjoint groups that act together and report them as the two interacting substructures, rather than one large group that contains both. Every node set, connected or not, is scored as one masked graph, so both the pairwise interactions of Phase 1 and the group interactions of the group ranking are measured between any two parts of the graph, adjacent or far apart. The joint effect of two distant motifs therefore appears as an interaction between two groups, each holding one motif, and its sign tells whether the two act jointly or interchangeably. On the controlled tasks, whenever its top-ranked pair of groups held both motifs, the pair had the sign of the planted interaction and its two groups lay 2 to 10 hops apart, whereas any connected node set holding both motifs would cover more than half of the graph.

On the test graphs of a public benchmark of thirteen datasets, against eleven explainers, PARSE was the most faithful method in 58 of 65 comparisons. Separating each node's own contribution from its pairwise interactions thus yields explanations that are both faithful to the model and informative about how its substructures act together.

AI USE STATEMENT

In this work, we used generative AI tools (large language models) to implement methods: they assisted in writing the code of PARSE, of the experiments and of the baseline setup, which follows each baseline's paper and released code. We also used them for feedback on the design of the experiments and to help analyze and interpret the results, and, among the uses with recommended disclosure, to draft and edit parts of this paper and to format references. We did not use them for translation or for qualitative data analysis. We have reviewed all AI-assisted work: the code was checked by running it and comparing its outputs with the recorded results, and every number in the text was checked against these results. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work raises no ethical concerns. It uses only publicly available benchmark datasets and synthetic graphs, involves no human subjects and collects no new data.

REPRODUCIBILITY STATEMENT

The code of PARSE, the trained models and all results, including every method's node sets and metric values on each test graph, are available at `https://anonymous.4open.science/r/PARSE` and are included in the supplementary material. Section 4 and Algorithm 1 specify PARSE; Appendix A gives its fixed settings and its calibrated setting for each dataset, Appendix B the datasets, splits and trained models, Appendix C the implementation and settings of every baseline, Appendix D the evaluation protocol, Appendix E every benchmark result and Appendix F the controlled tasks.

REFERENCES

C. Agarwal, O. Queen, H. Lakkaraju, and M. Zitnik, "Evaluating explainability for graph neural networks," *Scientific Data*, vol. 10, art. 144, 2023.

K. Amara, Z. Ying, Z. Zhang, Z. Han, Y. Zhao, Y. Shan, U. Brandes, S. Schemm, and C. Zhang, "GraphFramEx: Towards systematic evaluation of explainability methods for graph neural networks," in *Proc. 1st Learning on Graphs Conf. (LoG)*, PMLR, vol. 198, pp. 44:1–44:23, 2022.

D. Amato, S. Calderaro, G. Lo Bosco, R. Rizzo, and F. Vella, "GECo: A community-based graph neural network explainer," *Computing*, vol. 108, no. 4, art. 55, 2026. doi: 10.1007/s00607-026-01642-z.

S. Bordt and U. von Luxburg, "From Shapley values to generalized additive models and back," in *Proc. 26th Int. Conf. Artif. Intell. Statist. (AISTATS)*, PMLR, vol. 206, 2023, pp. 709–745.

N. Bui, H. T. Nguyen, V. A. Nguyen, and R. Ying, "Explaining graph neural networks via structure-aware interaction index," in *Proc. 41st Int. Conf. Mach. Learn. (ICML)*, PMLR, vol. 235, 2024.

J. Castro, D. Gómez, and J. Tejada, "Polynomial calculation of the Shapley value based on sampling," *Computers & Operations Research*, vol. 36, no. 5, pp. 1726–1730, 2009.

I. Covert, S. Lundberg, and S.-I. Lee, "Explaining by removing: A unified framework for model explanation," *J. Mach. Learn. Res.*, vol. 22, no. 209, pp. 1–90, 2021.

J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, 2006.

A. Duval and F. D. Malliaros, "GraphSVX: Shapley value explanations for graph neural networks," in *Proc. Eur. Conf. Mach. Learn. Knowl. Discovery Databases (ECML-PKDD)*, 2021, pp. 302–318.

V. P. Dwivedi, A. T. Luu, T. Laurent, Y. Bengio, and X. Bresson, "Graph neural networks with learnable structural and positional representations," in *Proc. 10th Int. Conf. Learn. Representations (ICLR)*, 2022.

L. Faber, A. K. Moghaddam, and R. Wattenhofer, "When comparing to ground truth is wrong: On evaluating GNN explanation methods," in *Proc. 27th ACM SIGKDD Conf. Knowl. Discovery Data Mining (KDD)*, 2021, pp. 332–341.

F. Fumagalli, M. Muschalik, P. Kolpaczki, E. Hüllermeier, and B. Hammer, "SHAP-IQ: Unified approximation of any-order Shapley interactions," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 36, 2023, pp. 11515–11551.

M. Grabisch and M. Roubens, "An axiomatic approach to the concept of interaction among players in cooperative games," *Int. J. Game Theory*, vol. 28, no. 4, pp. 547–565, 1999.
S. Holm, "A simple sequentially rejective multiple test procedure," *Scandinavian Journal of Statistics*, vol. 6, no. 2, pp. 65–70, 1979.
Q. Huang, M. Yamada, Y. Tian, D. Singh, and Y. Chang, "GraphLIME: Local interpretable model explanations for graph neural networks," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 7, pp. 6968–6972, 2023.
D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient global optimization of expensive black-box functions," *J. Global Optimization*, vol. 13, no. 4, pp. 455–492, 1998.
J. Kazius, R. McGuire, and R. Bursi, "Derivation and validation of toxicophores for mutagenicity prediction," *J. Medicinal Chemistry*, vol. 48, no. 1, pp. 312–320, 2005.
T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. 5th Int. Conf. Learn. Representations (ICLR)*, 2017.
S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.
W. Lin, "Large-scale network embedding in Apache Spark," in *Proc. 27th ACM SIGKDD Conf. Knowl. Discovery Data Mining (KDD)*, 2021, pp. 3271–3279.
W. Lin, F. He, F. Zhang, X. Cheng, and H. Cai, "Initialization for network embedding: A graph partition approach," in *Proc. 13th ACM Int. Conf. Web Search and Data Mining (WSDM)*, 2020, pp. 367–374.
W. Lin, H. Lan, and B. Li, "Generative causal explanations for graph neural networks," in *Proc. 38th Int. Conf. Mach. Learn. (ICML)*, PMLR, vol. 139, 2021, pp. 6666–6679.
M. Liu *et al.*, "DIG: A turnkey library for diving into graph deep learning research," *J. Mach. Learn. Res.*, vol. 22, no. 240, pp. 1–9, 2021.
A. Lucic, M. A. ter Hoeve, G. Tolomei, M. de Rijke, and F. Silvestri, "CF-GNNExplainer: Counterfactual explanations for graph neural networks," in *Proc. 25th Int. Conf. Artif. Intell. Statist. (AISTATS)*, PMLR, vol. 151, 2022, pp. 4499–4511.
S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 30, 2017.
D. Luo, W. Cheng, D. Xu, W. Yu, B. Zong, H. Chen, and X. Zhang, "Parameterized explainer for graph neural network," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 33, 2020.
C. Morris, N. M. Kriege, F. Bause, K. Kersting, P. Mutzel, and M. Neumann, "TUDataset: A collection of benchmark datasets for learning with graphs," in *ICML 2020 Workshop on Graph Representation Learning and Beyond (GRL+)*, 2020. arXiv:2007.08663.
M. Muschalik, F. Fumagalli, P. Frazzetto, J. Strotherm, L. Hermes, A. Sperduti, E. Hüllermeier, and B. Hammer, "Exact computation of any-order Shapley interactions for graph neural networks," in *Proc. 13th Int. Conf. Learn. Representations (ICLR)*, 2025.
P. E. Pope, S. Kolouri, M. Rostami, C. E. Martin, and H. Hoffmann, "Explainability methods for graph convolutional neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019, pp. 10764–10773.
B. Sanchez-Lengeling, J. Wei, B. Lee, E. Reif, P. Wang, W. W. Qian, K. McCloskey, L. Colwell, and A. Wiltschko, "Evaluating attribution for graph neural networks," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 33, 2020.
T. Schnake, O. Eberle, J. Lederer, S. Nakajima, K. T. Schütt, K.-R. Müller, and G. Montavon, "Higher-order explanations of graph neural networks via relevant walks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 11, pp. 7581–7596, 2022.
L. S. Shapley, "A value for n-person games," in *Contributions to the Theory of Games*, vol. II, Princeton Univ. Press, 1953, pp. 307–317.
A. Shrikumar, P. Greenside, A. Shcherbina, and A. Kundaje, "Not just a black box: Learning important features through propagating activation differences," *arXiv:1605.01713*, 2016.
R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proc. Conf. Empirical Methods Natural Language Processing (EMNLP)*, 2013, pp. 1631–1642.
E. Štrumbelj and I. Kononenko, "Explaining prediction models and individual predictions with feature contributions," *Knowl. Inf. Syst.*, vol. 41, no. 3, pp. 647–665, 2014.
M. Sundararajan, K. Dhamdhere, and A. Agarwal, "The Shapley Taylor interaction index," in *Proc. 37th Int. Conf. Mach. Learn. (ICML)*, PMLR, vol. 119, 2020, pp. 9259–9268.
J. Tan, S. Geng, Z. Fu, Y. Ge, S. Xu, Y. Li, and Y. Zhang, "Learning and evaluating graph neural network explanations based on counterfactual and factual reasoning," in *Proc. ACM Web Conf. (WWW)*, 2022, pp. 1018–1027.
C.-P. Tsai, C.-K. Yeh, and P. Ravikumar, "Faith-Shap: The faithful Shapley interaction index," *J. Mach. Learn. Res.*, vol. 24, no. 94, pp. 1–42, 2023.

M. N. Vu and M. T. Thai, "PGM-Explainer: Probabilistic graphical model explanations for graph neural networks," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 33, 2020, pp. 12225–12235.
F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bulletin*, vol. 1, no. 6, pp. 80–83, 1945.
Z. Wu, B. Ramsundar, E. N. Feinberg, J. Gomes, C. Geniesse, A. S. Pappu, K. Leswing, and V. Pande, "MoleculeNet: A benchmark for molecular machine learning," *Chemical Science*, vol. 9, no. 2, pp. 513–530, 2018.
L. Wu, D. Hao, and Z. Fan, "Explainable graph neural networks via structural externalities," in *Proc. 34th Int. Joint Conf. Artif. Intell. (IJCAI)*, 2025.
K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?," in *Proc. 7th Int. Conf. Learn. Representations (ICLR)*, 2019.
Z. Ye, R. Huang, Q. Wu, and Q. Liu, "SAME: Uncovering GNN black box with structure-aware Shapley-based multipiece explanations," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 36, 2023.
R. Ying, D. Bourgeois, J. You, M. Zitnik, and J. Leskovec, "GNNExplainer: Generating explanations for graph neural networks," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 32, 2019.
H. Yuan, J. Tang, X. Hu, and S. Ji, "XGNN: Towards model-level explanations of graph neural networks," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2020, pp. 430–438.
H. Yuan, H. Yu, J. Wang, K. Li, and S. Ji, "On explainability of graph neural networks via subgraph explorations," in *Proc. 38th Int. Conf. Mach. Learn. (ICML)*, PMLR, vol. 139, 2021, pp. 12241–12252.
H. Yuan, H. Yu, S. Gui, and S. Ji, "Explainability in graph neural networks: A taxonomic survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 5, pp. 5782–5799, 2023.
S. Zhang, Y. Liu, N. Shah, and Y. Sun, "GStarX: Explaining graph neural networks with structure-aware cooperative games," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 35, 2022.
X. Zheng, F. Shirani, T. Wang, W. Cheng, Z. Chen, H. Chen, H. Wei, and D. Luo, "Towards robust fidelity for evaluating explainability of graph neural networks," in *Proc. 12th Int. Conf. Learn. Representations (ICLR)*, 2024.

A SETTINGS AND CALIBRATION

PARSE has two kinds of settings. The fixed settings of Table 3 are shared by all datasets; they set the cost of Phase 1, the partition search of Phases 2–3 and the set search of Phase 4. The five settings $\theta = (\gamma_H, \gamma_S, p_{\mathrm{cov}}, \mu_c, \mu_\beta)$ of the partition score are calibrated once per dataset on validation graphs, and Table 4 gives the adopted values and the one-off cost of this calibration.

A.1 FIXED SETTINGS

Phase 3 makes $N_{\mathrm{SA}} = 100$ annealing proposals, starting at temperature $\tau_0 = 1$ and cooling by the factor $\kappa = 0.99$ per step. Phase 4 searches a budget exhaustively when it has at most $N_{\mathrm{ex}} = 5{,}000$ candidate sets, and otherwise by swap search from four starting sets: the nodes of largest $|\hat{\phi}_i|$, a set grown along edges, the partition start and a set built greedily.

The components of θ shape the partition score R_L that Phases 2–3 maximize. The weight γ_H scales the entropy term, which penalizes evidence spread over many units; γ_S scales the size cost, which penalizes units far from the preferred size; p_{cov} sets how fast the coverage discounts later units; and μ_c and μ_β set the preferred size $\mu = \mu_c n^{\mu_\beta}$, which also fixes the number k_0 of groups at which Phase 2 stops.

Table 3: Fixed settings of PARSE, shared by all datasets, and the ranges within which θ is calibrated per dataset.

Step	Setting	Value
Phase 1	orderings m	50
Phases 2–3	annealing steps N_{SA}; τ_0; κ	100; 1; 0.99
Phase 4	sparsities $\mathcal{S}$	$\{0.5, 0.6, 0.7, 0.8, 0.9\}$
	exhaustive limit N_{ex}	5,000 subsets
Calibration	validation graphs; trials	50 per dataset (35 on ENZYMES); 40
	range of γ_H	$[0.1, 1]$
	range of γ_S	$[0.1, 3]$
	range of p_{cov}	$[0.5, 2]$
	range of μ_c	$[0.5, 2]$
	range of μ_β	$[0.2, 0.8]$

A.2 CALIBRATION

The calibration objective J averages H-Fidelity, Fid^+, characterization and $-|\mathrm{Fid}^-|$ over the five sparsities and the calibration graphs. The calibration graphs are 50 graphs drawn at random from each dataset's validation split (35 on ENZYMES). PARSE runs 40 trials per dataset: the first 12 points of a scrambled Sobol sequence over the ranges of Table 3, then 28 points proposed by Gaussian-process expected improvement (Jones et al., 1998) with a Matérn-5/2 kernel. PARSE adopts the trial with the highest J among those that finish within the time and memory limits under which every method is run.

Table 4 lists the adopted settings. Calibration is a one-off cost per dataset, about 1.4 hours on average. Appendix E.5 gives the per-graph cost of every method.

Table 4: PARSE's adopted setting θ per dataset, and the time and masked graphs of its calibration, summed over all trials.

Dataset	γ_H	γ_S	p_{cov}	μ_c	μ_β	Time (s)	Masked graphs
BACE	0.712	0.461	0.800	1.134	0.534	7,492	80,060,704
BBBP	0.431	0.140	1.856	0.557	0.549	6,384	59,710,686
Benzene	0.342	0.533	1.600	1.883	0.406	790	19,687,766
ENZYMES	0.714	2.463	0.960	1.891	0.407	7,276	41,505,079
Graph-SST2	0.797	0.227	0.780	0.780	0.536	1,117	4,567,684
Graph-SST5	0.197	2.914	1.894	0.554	0.566	2,927	15,999,531
Graph-Twitter	0.282	1.104	1.089	1.165	0.319	2,497	25,576,695
Mutagenicity	0.942	0.226	1.308	0.567	0.609	8,171	81,839,002
PROTEINS	0.416	1.260	1.017	1.984	0.797	8,385	84,208,624
BA-2Motifs	0.414	2.059	0.893	1.450	0.563	3,994	25,115,570
BA-House-And-Grid	0.878	1.346	1.378	1.384	0.616	11,056	41,999,568
BA-House-Grid	0.387	2.065	1.576	1.479	0.715	946	23,746,940
BA-House-Or-Grid	0.283	1.120	0.962	1.141	0.405	3,071	25,379,565
Total						64,107	529,397,414

B Datasets and models

The public benchmark uses thirteen graph-classification datasets from four domains: molecules, proteins, sentences represented as graphs, and synthetic graphs with planted motifs. For each dataset, one GNN classifier is trained on the training split and selected on the validation split; it is the model f that every method explains. Table 5 summarizes the datasets, their splits, the models and the explained test graphs.

B.1 The datasets

Molecules. Nodes are atoms and edges are bonds. BACE and BBBP (Wu et al., 2018) carry 9 atom features, and Benzene and Mutagenicity a one-hot encoding of 14 atom types. BACE labels 1,513 molecules by whether they inhibit human β-secretase 1, and BBBP labels molecules by whether they penetrate the blood–brain barrier; we use its first 2,000 molecules with at least three atoms. Benzene (Sanchez-Lengeling et al., 2020; Agarwal et al., 2023) labels 12,000 molecules by whether they contain a benzene ring, and Mutagenicity (Kazius et al., 2005; Morris et al., 2020) labels 4,337 molecules as mutagenic or not.

Proteins (Morris et al., 2020). Nodes are secondary-structure elements, joined when they are neighbours along the amino-acid sequence or close in space. ENZYMES assigns 600 enzymes to the six top-level enzyme classes, with 21 node features, and PROTEINS labels 1,113 proteins as enzymes or not, with 4 node features.

Sentiment graphs (Socher et al., 2013; Yuan et al., 2023; Liu et al., 2021). Each graph is a sentence: its nodes are words, with 768-dimensional pretrained word embeddings as features, and its edges follow the sentence's dependency parse. Graph-SST2 (two classes; its first 2,000 graphs) and Graph-SST5 (11,855 graphs, five classes) are built from movie-review sentences with sentiment labels, and Graph-Twitter (6,940 graphs, three classes) from tweets.

Synthetic graphs. BA-2Motifs (Luo et al., 2020) has 1,000 graphs of 25 nodes: a 20-node Barabási–Albert base graph with a house (class 1) or a five-node cycle (class 0) attached. Every node carries the same 10-dimensional feature vector, so only the structure is informative. BA-House-Grid, BA-House-And-Grid and BA-House-Or-Grid follow Bui et al. (2024). Each has 1,000 graphs, built from a Barabási–Albert base graph of 12 to 18 nodes to which a five-node house, a nine-node 3×3 grid, both or (in the And and Or variants) neither is attached, each motif by one edge to a random base node. BA-House-Grid separates graphs with a house (class 0) from graphs with a grid (class 1); in BA-House-And-Grid a graph is in class 1 exactly when both motifs are present, and in BA-House-Or-Grid when at least one is. Nodes carry their degree, clustering coefficient and $1/(\text{degree} + 1)$, standardized within each graph. The And and Or variants thus bring synergy and redundancy between two motifs into the public benchmark.

B.2 SPLITS

Every dataset is split at random into 80% training, 10% validation and 10% test graphs (Table 5). The models are trained on the training split, and model selection and PARSE's calibration use the validation split. On the three BA-House datasets, the test graphs come from a new sample of 100 graphs from the same generator. Appendix D.3 describes which test graphs are explained.

B.3 MODELS

Every model has three message-passing layers of width 64 with ReLU activations, mean pooling and a linear output layer. Four variants are trained per dataset: GCN (Kipf & Welling, 2017) or GIN (Xu et al., 2019) layers, each with or without batch normalization. Each variant is trained on the training split (on Benzene and Graph-SST5, on 6,000 graphs sampled from it) with Adam (learning rate 10^{-3}, weight decay 10^{-5}, batch size 64) for up to 150 epochs, keeping the weights with the best validation accuracy and stopping after 30 epochs without improvement.

The variant is chosen on validation data. It is eligible if its validation accuracy exceeds the majority-class rate and if masking random node sets changes its predicted-class probability, which fidelity needs in order to separate explainers; the most accurate eligible variant is chosen.

Table 5: Datasets and models of the public benchmark: split sizes (BBBP and Graph-SST2: their first 2,000 graphs), layer type of the model, validation accuracy of the model and of the majority class, and the number and node counts (mean, min–max) of the explained test graphs.

Dataset	Domain	Classes	Graphs (train/val/test)	Model	Val. acc.	Majority	Test graphs	Nodes, test draw
BACE	molecules (MoleculeNet)	2	1,210 / 151 / 152	GCN	0.868	0.642	17	31.9 (18–48)
BBBP	molecules (MoleculeNet)	2	1,600 / 200 / 200	GCN	0.915	0.820	36	24.4 (11–57)
Benzene	molecules (GraphXAI)	2	9,600 / 1,200 / 1,200	GIN	0.925	0.527	50	19.7 (8–25)
ENZYMES	proteins (TU)	6	480 / 60 / 60	GIN	0.667	0.250	50	29.0 (5–59)
Graph-SST2	sentiment text graphs (DIG)	2	1,600 / 200 / 200	GIN	0.835	0.560	50	9.4 (1–43)
Graph-SST5	sentiment text graphs (DIG)	5	9,484 / 1,185 / 1,186	GIN (no BN)	0.478	0.272	50	19.7 (3–38)
Graph-Twitter	sentiment text graphs (DIG)	3	5,552 / 694 / 694	GCN	0.673	0.480	50	20.3 (5–36)
Mutagenicity	molecules (TU)	2	3,469 / 433 / 435	GIN	0.855	0.552	50	36.7 (7–138)
PROTEINS	proteins (TU)	2	890 / 111 / 112	GIN	0.730	0.541	50	43.8 (6–481)
BA-2Motifs	synthetic (DIG)	2	800 / 100 / 100	GIN	1.000	0.550	50	25.0 (25–25)
BA-House-And-Grid	synthetic (generated)	2	800 / 100 / 100[‡]	GCN	1.000	0.530	50	24.1 (12–32)
BA-House-Grid	synthetic (generated)	2	800 / 100 / 100[‡]	GCN	1.000	0.540	50	22.1 (17–27)
BA-House-Or-Grid	synthetic (generated)	2	800 / 100 / 100[‡]	GCN	1.000	0.540	50	21.0 (12–32)

[‡]Test graphs drawn from a new 100-graph sample of the dataset's generator.

C BASELINES

The eleven baselines span five families of explainers: gradients, learned masks, Shapley values, interaction indices and subgraph search. Each baseline follows its authors' published choices wherever they apply (Appendices C.1 and C.3 list the adaptations), and every output is scored in the same game as PARSE.

C.1 IMPLEMENTATIONS AND SETTINGS

GNNExplainer and PGExplainer run in PyTorch Geometric and SubgraphX in DIG (Liu et al., 2021); GStarX, GraphEXT, GraphSHAP-IQ, MAGE and GECo run in their authors' released code, GraphSVX in the port included in the GStarX release, and SAME in the version included in the MAGE release. Grad$\times$Input is implemented directly: the absolute value of the gradient of the predicted-class logit times the input, summed over each node's features.

Each hyperparameter is taken, by preference, from (i) the value the authors' paper states for that dataset; (ii) the authors' released configuration for that dataset; (iii) the value the paper states for all its experiments; (iv) the default of the released code. On the datasets its paper does not cover, SAME uses the values of Table 6. The benchmark fixes the budget and size settings, among them the five sparsities and the size parameters of SubgraphX, SAME and MAGE, as well as the time and memory limits shared by all methods. Table 6 lists every implementation and setting, including the values that differ across datasets.

Table 6: Baseline implementations and settings (the authors' values, except those fixed by the benchmark or listed in Appendices C.1 and C.3; values that differ across datasets in parentheses), and how each output becomes one node set per budget.

Method	Implementation	Settings	Node sets
Grad×Input (Shrikumar et al., 2016)	our implementation	gradient of the predicted-class logit times input, absolute value summed over features	signed-descending prefixes
GNNExplainer (Ying et al., 2019)	PyTorch Geometric	Adam; learning rate 0.001; 100 epochs; edge-size coefficient 0.005; edge-entropy coefficient 1; one node mask shared by all features; feature-size coefficient 1; feature-entropy coefficient 0.1; raw model outputs	|score|-descending prefixes
PGExplainer (Luo et al., 2020)	PyTorch Geometric	30 epochs; learning rate 0.003; edge-size coefficient 0.05; edge-entropy coefficient 1; temperature 5→2; sampling bias 0; hidden width 64; 200 training graphs	signed-descending prefixes
GraphSVX (Duval & Malliaros, 2021)	GraphSVX port in the GStarX authors' release	200 samples; $S = 4$; separated coalition sampling; weighted linear regression; masked nodes removed	signed-descending prefixes
SAME (Ye et al., 2023)	SAME code in the MAGE authors' release	20 rollouts; 100 samples; 7 candidates; exploration constant 10 (5 on BBBP); single explanation size 3 (2 on BACE; 5 on BBBP); local radius 4 (5 on BACE); expansion size 14 (20 on BACE, Graph-Twitter, BA-2Motifs; 12 on BBBP, Graph-SST2, Graph-SST5); high-to-low expansion (low-to-high on Graph-SST2, Graph-SST5, Graph-Twitter); local Monte Carlo Shapley reward; split masking; maximum explanation size 5	native per-budget MCTS coalition, completed to k nodes
GraphEXT (Wu et al., 2025)	authors' GraphEXT code	samples 100	signed-descending prefixes
GStarX (Zhang et al., 2022)	GStarX authors' release	$\tau = 0.01$; maximum sample size 10; normalized-probability payoff; no superadditive extension; random sampling; 4 hops; masked nodes removed; 200 payoff reference graphs	signed-descending prefixes
GraphSHAP-IQ (Muschalik et al., 2025)	authors' code (bounded approximation)	k-SII of order 2; interactions of at most 2 nodes; the graph's mean node features as baseline; at most 100,000 queries and 512 nodes; work budget 150,000,000	signed-descending prefixes
MAGE (Bui et al., 2024)	MAGE authors' release	200 samples; $\tau = 1$; $\omega = 0.5$; $\epsilon = 0.1$; order-2 Myerson–Taylor index; normalized-logit payoff (normalized probability on BACE, BBBP, ENZYMES, PROTEINS, BA-2Motifs, BA-House-Grid); split masking (zero filling on Graph-SST2, Graph-SST5, Graph-Twitter); 1 motif (4 on Graph-SST2, Graph-SST5, Graph-Twitter); connectivity constraint (none on Graph-SST2, Graph-SST5, Graph-Twitter); explanation size $0.5n$	|score|-descending prefixes
SubgraphX (Yuan et al., 2021)	DIG	20 rollouts; 100 samples; exploration constant 10 (5 on BBBP, Graph-SST2, Graph-Twitter); expansion size 12 (20 on Graph-Twitter); low-to-high expansion; local radius 4; local Monte Carlo Shapley reward; zero filling; minimum size 5	native per-budget MCTS coalition, completed to k nodes
GECo (Amato et al., 2026)	authors' released package, unchanged	none exposed	prefixes of the native ranking

C.2 From outputs to scored node sets

Several baselines use their own value function internally, for example SubgraphX's zero filling or the split masking of SAME and MAGE. All outputs are scored like PARSE's: for the same explained class $c^\star$, under training-mean masking, and with exactly k_s nodes at each sparsity s. Each output becomes one node set per budget by a fixed rule (last column of Table 6). Grad×Input, PGExplainer, GraphSVX, GraphEXT, GStarX and GraphSHAP-IQ contribute their k_s highest-scored nodes, and GNNExplainer and MAGE their k_s nodes of largest absolute score. GNNExplainer and PGExplainer score edges: nodes are ranked by the mean mask value of their incident edges, and a graph without edges gets no explanation from them. MAGE's node score is $T_{ii} + \frac{1}{2}\sum_{j \neq i} |T_{ij}|$ for its order-2 Myerson–Taylor interaction matrix $\mathbf{T}$, from which MAGE selects its motifs. SubgraphX and SAME return one coalition per budget from their tree search; a smaller coalition is completed with the nodes of highest reward over the coalitions the search explored, and a larger one keeps its k_s members of highest reward. GECo contributes the first k_s nodes of its own ranking. All methods run under the same limits and protocol as PARSE (Appendix D).

C.3 Adaptations to the benchmark

The following settings adapt the authors' choices to the protocol and to the benchmark graphs.

- GStarX normalizes its payoff by the model's mean output over a set of reference graphs, here the first 200 training graphs.
- PGExplainer's mask network is trained once per dataset in its authors' inductive setting, on the first 200 training graphs with at least one edge, with its paper's setting on every dataset.
- GraphSHAP-IQ computes its index exactly by default, which is infeasible on almost every benchmark graph. It therefore runs its bounded order-2 approximation (k-SII over interactions of at most two nodes) with a cap of 100,000 model queries per graph, masking with the graph's mean node features as in its paper; its node scores are the Shapley values implied by these interactions.
- Some of MAGE's released configurations take the number of motifs from the ground truth, which an explainer may not read; it is set to 1, the authors' own value when no ground truth is used. This setting shapes only MAGE's stored motifs, not the node scores we evaluate.

D Evaluation protocol and statistics

This section describes how every number of Appendix E is produced: how explanations are scored, how PARSE is compared with the baselines on the graphs both explained, and which test graphs are explained.

D.1 Scoring

Every method is scored in the same way. The explained class $c^\star$ is the model's prediction on the full graph, and a masked node receives the training-mean features $\bar{\mathbf{x}}$ and loses its edges. At each sparsity $s \in \{0.5, 0.6, 0.7, 0.8, 0.9\}$ a method provides one set S of exactly k_s nodes. Its necessity $\mathrm{Fid}^+(S) = v(V) - v(V \setminus S)$ is the drop of the predicted-class probability when S is masked (higher is better), and its sufficiency error $\mathrm{Fid}^-(S) = v(V) - v(S)$ is the drop when every other node is masked (closer to zero is better). H-Fidelity combines the two with size weights, and characterization is the harmonic mean of $\max(0, \mathrm{Fid}^+)$ and $1 - \max(0, \mathrm{Fid}^-)$. Fid_α is computed on the set at $s = 0.5$: its necessity side masks each explanation node with probability $\alpha = 0.7$ and each other node with probability $1 - \alpha$, its sufficiency side swaps the two probabilities, and each side is the drop of the predicted-class probability from $v(V)$, averaged over 50 random masks. Each metric is averaged over the five sparsities within a graph and then over graphs. Because Fid^- is best close to zero, comparisons use the absolute value of the mean signed Fid^-; the per-method means of Appendix E.7 list the signed mean itself, and the masker check of Appendix E.4 the mean per-explanation $|\mathrm{Fid}^-|$.

Appendix E.4 re-scores every method's node sets under three other maskers.

D.2 Comparisons and statistics

For each dataset and metric, PARSE is compared with each baseline on the test graphs both explained, a standard paired design (Demšar, 2006). A difference of means no larger than one standard error of the paired per-graph differences is a tie; for Fid^- the standard error is the delta-method error of the difference of absolute means. PARSE wins a comparison if it beats every baseline, ties for best if no baseline beats it but at least one ties, and loses otherwise.

Three further quantities are reported for each comparison: the comparison baseline, which is the baseline that beats PARSE or else the closest one, whose margin exceeds its tie threshold by the least; a 95% percentile interval of PARSE's margin over it from 2,000 bootstrap resamples of the shared graphs; and a two-sided Wilcoxon signed-rank test (Wilcoxon, 1945) with Holm's correction (Holm, 1979) over all 65 comparisons. For Fid^-, the interval resamples the difference of absolute means, and the Wilcoxon test uses the per-graph differences after each method's values are multiplied by the sign of its mean.

D.3 Test draw, run and coverage

Each dataset contributes a random draw of up to 50 graphs from its test split (for the BA-House datasets, from the new generator sample of Appendix B.2). The draw holds 603 graphs in all; Table 5 gives the count per dataset.

Every method runs on each test graph under the same limits: 600 s and 5 GiB of memory per graph, 7,200 s per method and dataset, and 3 threads; a graph on which a method returns no explanation within these limits counts as not explained.

PARSE explained 602 of the 603 test graphs and the baselines between 555 and 603; the graphs column of Tables 12–14 gives each method's count per dataset. Because every comparison is paired, a graph that a method did not explain drops out of the comparisons that involve that method.

E Full benchmark results

This section gives every number behind the benchmark results: every comparison with its significance, PARSE's lead across explanation sizes, graph sizes and maskers, the cost of each method, typical explanations and, last, every method's mean on each metric. All numbers follow the protocol of Appendix D.

E.1 Every comparison

Tables 7 and 8 list all 65 comparisons: PARSE's margin over the comparison baseline, its paired standard error, its bootstrap interval and the Wilcoxon test. PARSE wins 58 comparisons, ties for best in 3 and loses 4. After Holm's correction, 51 of the 58 wins are significant, including all 39 on H-Fidelity, Fid^{+} and characterization. The four losses are on Fid^{-} on BACE (against GraphEXT), BBBP (GECo) and Graph-SST5 (SAME), and on Fid_{α} on BA-House-Or-Grid (GraphSHAP-IQ); only those on BBBP and Graph-SST5 are significant. In 64 of the 65 comparisons the comparison baseline is a game-theoretic explainer, most often GraphEXT or GraphSHAP-IQ.

Table 7: Every (dataset, metric) comparison of the public benchmark: PARSE's margin over the comparison baseline (vs.: the baseline that beats PARSE, else the closest one) on their n shared graphs (positive when PARSE is better; Fid$^-$: |baseline| − |PARSE|), its paired standard error (SE), its 95% bootstrap interval over graphs, and the two-sided Wilcoxon signed-rank p (Wilcoxon, 1945) before and after Holm's correction (Holm, 1979) over all 65 comparisons. *: Holm $p < 0.05$.

Dataset	Metric	vs.	n	margin	SE	95% CI	Wilcoxon p	Holm p
BACE	H-Fid.	GraphEXT	17	+0.0350	0.0032	[+0.0287, +0.0408]	$1.53{\cdot}10^{-5}$	0.000351^*
	Fid$^+$	GraphEXT	17	+0.1863	0.0314	[+0.1289, +0.2446]	$1.53{\cdot}10^{-5}$	0.000351^*
	Fid$^-$	GraphEXT	17	−0.1206	0.0407	[−0.1699, −0.0344]	0.0305	0.275
	Char.	GraphEXT	17	+0.1753	0.0238	[+0.1311, +0.2203]	$1.53{\cdot}10^{-5}$	0.000351^*
	Fid$_\alpha$	GraphEXT	17	+0.0480	0.0228	[+0.0059, +0.0931]	0.0505	0.354
BBBP	H-Fid.	GraphEXT	36	+0.0355	0.0029	[+0.0302, +0.0413]	$2.91{\cdot}10^{-11}$	$1.28{\cdot}10^{-9*}$
	Fid$^+$	GraphEXT	36	+0.2306	0.0238	[+0.1857, +0.2793]	$2.91{\cdot}10^{-11}$	$1.28{\cdot}10^{-9*}$
	Fid$^-$	GECo	36	−0.0905	0.0182	[−0.1076, +0.0285]	$2.91{\cdot}10^{-10}$	$1.19{\cdot}10^{-8*}$
	Char.	GraphEXT	36	+0.2627	0.0280	[+0.2130, +0.3165]	$2.91{\cdot}10^{-11}$	$1.28{\cdot}10^{-9*}$
	Fid$_\alpha$	GraphEXT	36	+0.0305	0.0065	[+0.0189, +0.0437]	$9.76{\cdot}10^{-6}$	0.000244^*
Benzene	H-Fid.	GraphSVX	50	+0.0140	0.0027	[+0.0094, +0.0199]	$1.78{\cdot}10^{-15}$	$1.15{\cdot}10^{-13*}$
	Fid$^+$	GraphSVX	50	+0.0458	0.0146	[+0.0230, +0.0792]	$1.78{\cdot}10^{-15}$	$1.15{\cdot}10^{-13*}$
	Fid$^-$	SubgraphX	50	+0.0287	0.0065	[+0.0169, +0.0424]	$3.38{\cdot}10^{-6}$	$9.13{\cdot}10^{-5*}$
	Char.	GraphSVX	50	+0.0809	0.0159	[+0.0535, +0.1153]	$1.18{\cdot}10^{-9}$	$4.61{\cdot}10^{-8*}$
	Fid$_\alpha$	SAME	50	+0.0096	0.0045	[+0.0006, +0.0188]	0.00294	0.0383^*
ENZYMES	H-Fid.	GraphEXT	50	+0.0174	0.0013	[+0.0149, +0.0200]	$1.78{\cdot}10^{-15}$	$1.15{\cdot}10^{-13*}$
	Fid$^+$	GraphEXT	50	+0.0385	0.0053	[+0.0288, +0.0493]	$1.77{\cdot}10^{-11}$	$7.96{\cdot}10^{-10*}$
	Fid$^-$	SAME	50	+0.0329	0.0081	[+0.0174, +0.0493]	$8.24{\cdot}10^{-5}$	0.00148^*
	Char.	GraphEXT	50	+0.0531	0.0050	[+0.0435, +0.0625]	$2.49{\cdot}10^{-14}$	$1.19{\cdot}10^{-12*}$
	Fid$_\alpha$	GraphEXT	50	+0.0011	0.0036	[−0.0060, +0.0087]	0.706	1
Graph-SST2	H-Fid.	GraphSVX	50	+0.0093	0.0018	[+0.0060, +0.0129]	$1.68{\cdot}10^{-7}$	$5.38{\cdot}10^{-6*}$
	Fid$^+$	GraphSVX	50	+0.0547	0.0108	[+0.0347, +0.0766]	$1.31{\cdot}10^{-6}$	$3.79{\cdot}10^{-5*}$
	Fid$^-$	SAME	40	+0.0016	0.0082	[−0.0131, +0.0178]	0.645	1
	Char.	GraphSVX	50	+0.0591	0.0124	[+0.0366, +0.0865]	$5.85{\cdot}10^{-7}$	$1.81{\cdot}10^{-5*}$
	Fid$_\alpha$	GraphEXT	50	+0.0237	0.0051	[+0.0143, +0.0346]	$1.37{\cdot}10^{-5}$	0.000328^*
Graph-SST5	H-Fid.	GraphSVX	50	+0.0053	0.0007	[+0.0040, +0.0068]	$3.52{\cdot}10^{-9}$	$1.27{\cdot}10^{-7*}$
	Fid$^+$	GraphEXT	50	+0.0253	0.0047	[+0.0165, +0.0347]	$3.73{\cdot}10^{-6}$	$9.71{\cdot}10^{-5*}$
	Fid$^-$	SAME	50	−0.0158	0.0039	[−0.0230, −0.0085]	0.000331	0.00563^*
	Char.	GraphEXT	50	+0.0298	0.0046	[+0.0210, +0.0391]	$1.59{\cdot}10^{-8}$	$5.25{\cdot}10^{-7*}$
	Fid$_\alpha$	GraphSVX	50	+0.0105	0.0030	[+0.0051, +0.0167]	0.00174	0.0243^*
Graph-Twitter	H-Fid.	GraphSHAP-IQ	50	+0.0053	0.0005	[+0.0044, +0.0063]	$1.78{\cdot}10^{-15}$	$1.15{\cdot}10^{-13*}$
	Fid$^+$	GraphSHAP-IQ	50	+0.0286	0.0027	[+0.0235, +0.0338]	$1.24{\cdot}10^{-14}$	$6.22{\cdot}10^{-13*}$
	Fid$^-$	GraphSHAP-IQ	50	+0.0056	0.0023	[+0.0006, +0.0100]	0.0119	0.131
	Char.	GraphSHAP-IQ	50	+0.0403	0.0040	[+0.0325, +0.0478]	$1.78{\cdot}10^{-15}$	$1.15{\cdot}10^{-13*}$
	Fid$_\alpha$	GraphSHAP-IQ	50	+0.0209	0.0028	[+0.0160, +0.0267]	$8.87{\cdot}10^{-9}$	$3.02{\cdot}10^{-7*}$
Mutagenicity	H-Fid.	SAME	50	+0.0566	0.0053	[+0.0464, +0.0671]	$3.55{\cdot}10^{-15}$	$1.81{\cdot}10^{-13*}$
	Fid$^+$	GraphEXT	50	+0.2681	0.0206	[+0.2299, +0.3112]	$1.78{\cdot}10^{-15}$	$1.15{\cdot}10^{-13*}$
	Fid$^-$	SAME	50	+0.0988	0.0237	[+0.0371, +0.1435]	0.000816	0.0122^*
	Char.	GraphEXT	50	+0.3022	0.0217	[+0.2623, +0.3451]	$1.78{\cdot}10^{-15}$	$1.15{\cdot}10^{-13*}$
	Fid$_\alpha$	SAME	50	+0.1415	0.0359	[+0.0750, +0.2171]	0.000527	0.00843^*
PROTEINS	H-Fid.	SAME	46	+0.0203	0.0018	[+0.0171, +0.0238]	$2.84{\cdot}10^{-14}$	$1.34{\cdot}10^{-12*}$
	Fid$^+$	GraphEXT	49	+0.0813	0.0095	[+0.0637, +0.1011]	$1.78{\cdot}10^{-14}$	$8.7{\cdot}10^{-13*}$
	Fid$^-$	SAME	46	+0.0363	0.0072	[+0.0223, +0.0499]	$2.56{\cdot}10^{-6}$	$7.16{\cdot}10^{-5*}$
	Char.	GraphEXT	49	+0.1043	0.0124	[+0.0819, +0.1307]	$1.74{\cdot}10^{-9}$	$6.6{\cdot}10^{-8*}$
	Fid$_\alpha$	GraphEXT	49	+0.0168	0.0059	[+0.0056, +0.0281]	0.0106	0.128
BA-2Motifs	H-Fid.	GraphSVX	50	+0.0131	0.0010	[+0.0112, +0.0150]	$1.78{\cdot}10^{-15}$	$1.15{\cdot}10^{-13*}$
	Fid$^+$	SAME	50	+0.0632	0.0082	[+0.0478, +0.0799]	$1.78{\cdot}10^{-15}$	$1.15{\cdot}10^{-13*}$
	Fid$^-$	GraphSVX	50	+0.0103	0.0055	[−0.0006, +0.0198]	0.0176	0.176
	Char.	SAME	50	+0.0501	0.0057	[+0.0397, +0.0622]	$1.78{\cdot}10^{-15}$	$1.15{\cdot}10^{-13*}$
	Fid$_\alpha$	GraphEXT	50	+0.0030	0.0017	[−0.0003, +0.0063]	0.04	0.32
BA-House-And-Grid	H-Fid.	GraphSHAP-IQ	50	+0.0037	0.0007	[+0.0025, +0.0051]	$1.78{\cdot}10^{-15}$	$1.15{\cdot}10^{-13*}$
	Fid$^+$	GraphSHAP-IQ	50	+0.0286	0.0054	[+0.0186, +0.0390]	$4.52{\cdot}10^{-9}$	$1.58{\cdot}10^{-7*}$
	Fid$^-$	GraphSHAP-IQ	50	−0.0005	0.0014	[−0.0033, +0.0022]	0.774	1
	Char.	GraphSHAP-IQ	50	+0.0225	0.0051	[+0.0135, +0.0327]	$2.06{\cdot}10^{-9}$	$7.61{\cdot}10^{-8*}$
	Fid$_\alpha$	GraphSHAP-IQ	50	+0.0062	0.0031	[+0.0007, +0.0128]	0.0595	0.357

Table 8: Every comparison of the public benchmark (continued; columns as in Table 7).

Dataset	Metric	vs.	n	margin	SE	95% CI	Wilcoxon p	Holm p
BA-House-Grid	H-Fid.	GraphEXT	50	$+0.0041$	0.0004	$[+0.0032, +0.0049]$	$1.78{\cdot}10^{-15}$	$1.15{\cdot}10^{-13*}$
	Fid^{+}	GraphEXT	50	$+0.0093$	0.0019	$[+0.0059, +0.0134]$	$1.21{\cdot}10^{-6}$	$3.62{\cdot}10^{-5*}$
	Fid^{-}	GraphSHAP-IQ	50	$+0.0040$	0.0029	$[-0.0045, +0.0097]$	0.215	0.86
	Char.	GraphEXT	50	$+0.0161$	0.0020	$[+0.0124, +0.0204]$	$4.44{\cdot}10^{-14}$	$2.04{\cdot}10^{-12*}$
	Fid_{α}	GraphEXT	50	$+0.0126$	0.0026	$[+0.0078, +0.0179]$	$2.43{\cdot}10^{-5}$	0.000486^{*}
BA-House-Or-Grid	H-Fid.	GraphSHAP-IQ	50	$+0.0260$	0.0023	$[+0.0218, +0.0304]$	$1.78{\cdot}10^{-15}$	$1.15{\cdot}10^{-13*}$
	Fid^{+}	GraphSHAP-IQ	50	$+0.0724$	0.0170	$[+0.0388, +0.1040]$	$7.5{\cdot}10^{-5}$	0.00143^{*}
	Fid^{-}	GStarX	50	$+0.0902$	0.0141	$[+0.0627, +0.1167]$	$4.31{\cdot}10^{-10}$	$1.73{\cdot}10^{-8*}$
	Char.	GraphSHAP-IQ	50	$+0.1379$	0.0092	$[+0.1204, +0.1558]$	$1.78{\cdot}10^{-15}$	$1.15{\cdot}10^{-13*}$
	Fid_{α}	GraphSHAP-IQ	50	-0.0158	0.0106	$[-0.0358, +0.0047]$	0.0823	0.411

E.2 Explanation size

Figures 3 and 4 show the mean H-Fidelity and Fid^{+} of every method against sparsity on all thirteen datasets. PARSE's mean H-Fidelity is the highest of all methods at every sparsity on every dataset, and its mean Fid^{+} on eleven. The four datasets shown in the main text are those on which PARSE's 95% interval lies above every baseline mean at the most sparsities, counted over both metrics.

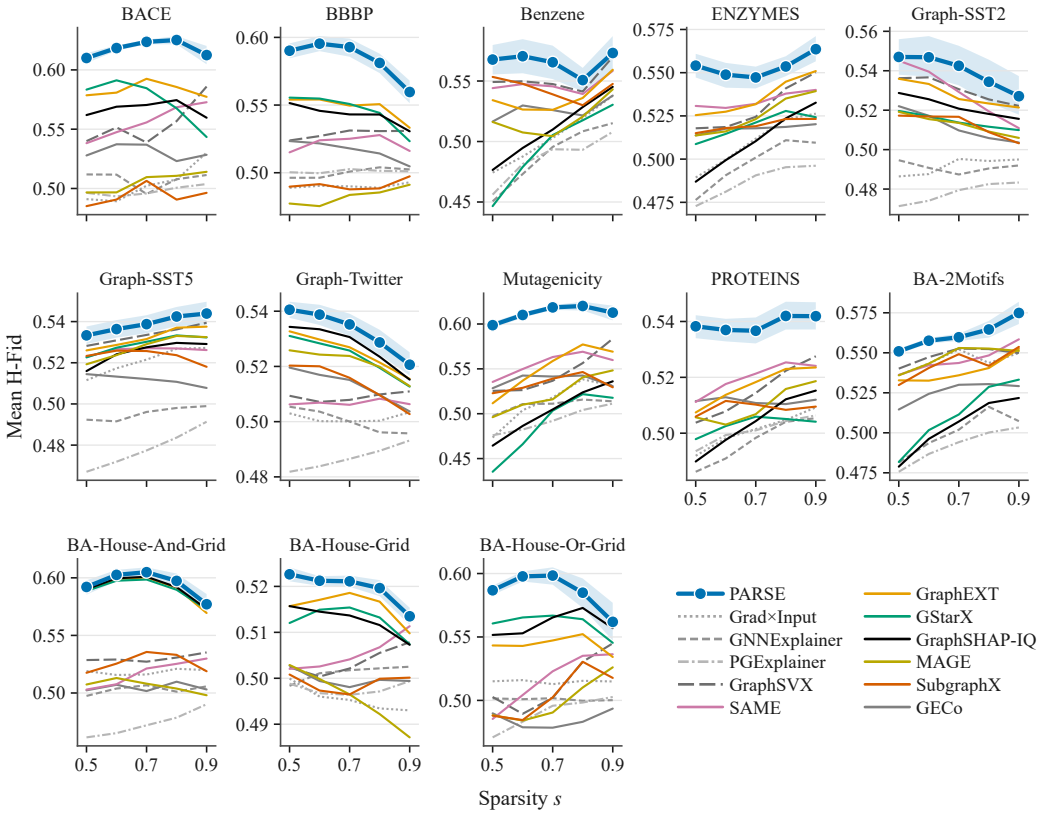


Figure 3: Mean H-Fidelity against sparsity on all thirteen datasets. Shaded: PARSE's 95% bootstrap interval.

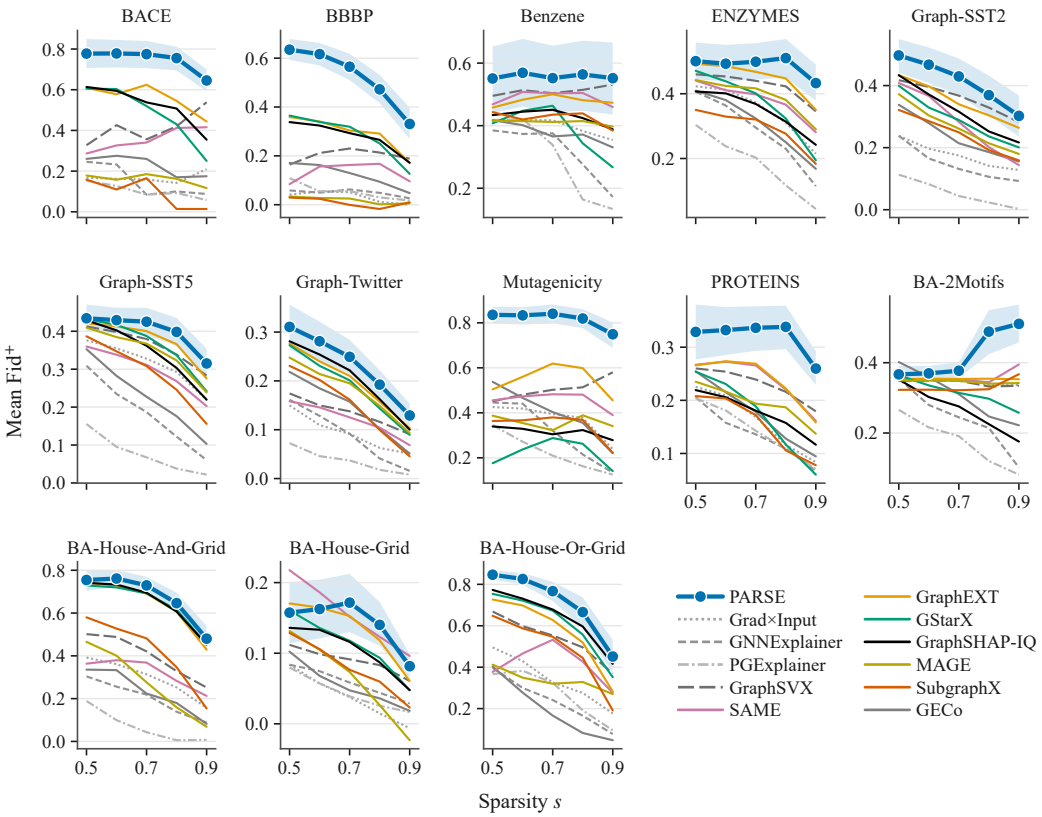


Figure 4: Mean Fid^{+} against sparsity on all thirteen datasets. Shaded: PARSE's 95% bootstrap interval.

E.3 Graph size

To test whether PARSE's lead depends on graph size, each dataset's test graphs are split into thirds by node count, and PARSE's margin over the comparison baseline of Appendix E.1 is recomputed on each third (Figure 5). The 95% interval of the margin lies above zero in all 39 thirds for H-Fidelity and in 38 of the 39 for Fid^{+}.

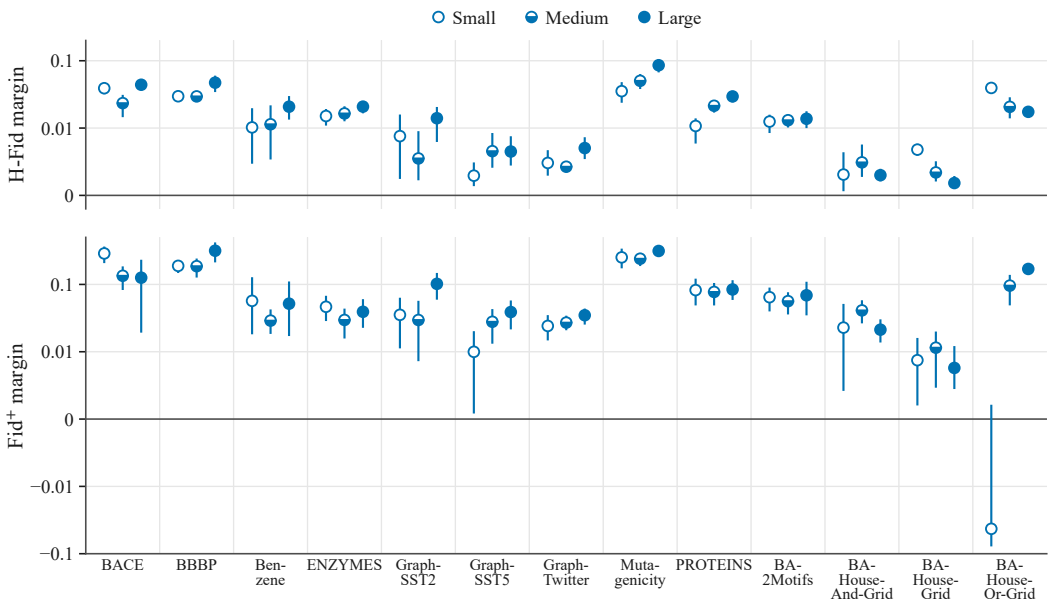


Figure 5: PARSE's margin over the comparison baseline of Tables 7 and 8 on the smallest, middle and largest third of each dataset's test graphs (open, half-filled and filled circles), with 95% bootstrap intervals. Top: H-Fidelity; bottom: Fid^{+}. Symmetric-log axis.

E.4 Other maskers

Since fidelity can vary with the masker (Zheng et al., 2024), the same node sets are re-scored under three other maskers: zero features with edges removed; training-mean features with edges kept (mean blur); and the features of randomly drawn training nodes with edges kept (reference blur). Table 9 gives every method's mean Fid^{+} and $|\mathrm{Fid}^{-}|$ under each masker. PARSE has the highest mean Fid^{+} and the lowest mean $|\mathrm{Fid}^{-}|$ under the benchmark masker, zero masking and reference blur, and ranks in the top two on both under all four maskers.

Table 9: Mean Fid^{+} and mean per-explanation $|\mathrm{Fid}^{-}|$ of every method's test node sets under four maskers, averaged over datasets: the benchmark's (training-mean features, edges removed), zero features with edges removed, and training-reference or training-mean features with edges kept (blur). Best per column in bold (highest Fid^{+}, lowest $|\mathrm{Fid}^{-}|$).

	benchmark		zero, hard		reference, blur		mean, blur	
Method	Fid^{+}	$\lvert\mathrm{Fid}^{-}\rvert$	Fid^{+}	$\lvert\mathrm{Fid}^{-}\rvert$	Fid^{+}	$\lvert\mathrm{Fid}^{-}\rvert$	Fid^{+}	$\lvert\mathrm{Fid}^{-}\rvert$
PARSE	**0.4959**	**0.1958**	**0.4100**	**0.2219**	**0.3026**	**0.1990**	0.3120	0.2015
GraphEXT	0.3988	0.2440	0.3465	0.2694	0.2881	0.2062	0.3072	0.2128
GraphSHAP-IQ	0.3527	0.2826	0.3061	0.3008	0.2952	0.2096	**0.3380**	**0.2005**
GraphSVX	0.3468	0.2696	0.3129	0.2912	0.2467	0.2265	0.2623	0.2483
GStarX	0.3403	0.2719	0.2969	0.2954	0.2511	0.2184	0.2757	0.2233
SAME	0.3077	0.2464	0.2836	0.2817	0.2118	0.2318	0.2194	0.2558
MAGE	0.2571	0.2762	0.2576	0.2962	0.1941	0.2393	0.1902	0.2634
SubgraphX	0.2540	0.2643	0.2610	0.2874	0.1949	0.2292	0.2092	0.2587
Grad×Input	0.2367	0.2929	0.2341	0.3078	0.1661	0.2589	0.1645	0.2857
GECo	0.2306	0.2577	0.2176	0.2850	0.1769	0.2337	0.1799	0.2654
GNNExplainer	0.1842	0.3147	0.1828	0.3360	0.1287	0.2766	0.1323	0.3042
PGExplainer	0.1324	0.3371	0.1341	0.3532	0.0985	0.2936	0.1033	0.3244

E.5 COST

Table 10 gives each method's time and peak memory on the test graphs, averaged over datasets. PARSE takes 2.51 s per graph (2.07 s on the graphs every method explained) and peaks at 4.58 GB of memory. PARSE's one-off calibration is given in Appendix A.2; the per-dataset setup of PGExplainer and GStarX, in parentheses, is spread over the graphs they explain.

Table 10: Cost on the public test benchmark, averaged over datasets (peak memory: the largest over datasets). (+x): per-dataset setup (PGExplainer training, GStarX payoff reference), amortized per explained graph; common: time on the graphs every method explained.

Method	s/graph	peak memory (GB)	s/graph (common)
SubgraphX	56.98	2.62	59.32
GraphEXT	19.42	2.56	16.23
GStarX	17.50 (+0.01)	2.57	15.98
SAME	13.88	2.63	12.95
MAGE	9.03	4.96	6.97
PARSE	2.51	4.58	2.07
PGExplainer	0.01 (+1.92)	2.53	0.01
GNNExplainer	0.72	2.57	0.71
GraphSVX	0.61	2.53	0.61
GECo	0.11	2.55	0.11
GraphSHAP-IQ	0.09	2.57	0.08
Grad$\times$Input	0.01	2.53	0.01

E.6 TYPICAL EXPLANATIONS

Figure 6 shows the nodes that PARSE and four interaction- and structure-aware baselines keep at sparsity 0.7 on Mutagenicity, BBBP and BA-House-Grid. The graph shown for each dataset is the test graph whose H-Fidelity margin of PARSE over the comparison baseline is closest to the dataset's median. Table 11 gives the fidelity of these node sets: on all three graphs, removing PARSE's nodes lowers the predicted-class probability more than removing the nodes of any of the eleven baselines, for example by 0.958 on the Mutagenicity graph against at most 0.865 for any baseline (GraphEXT) and 0.663 for the four shown.

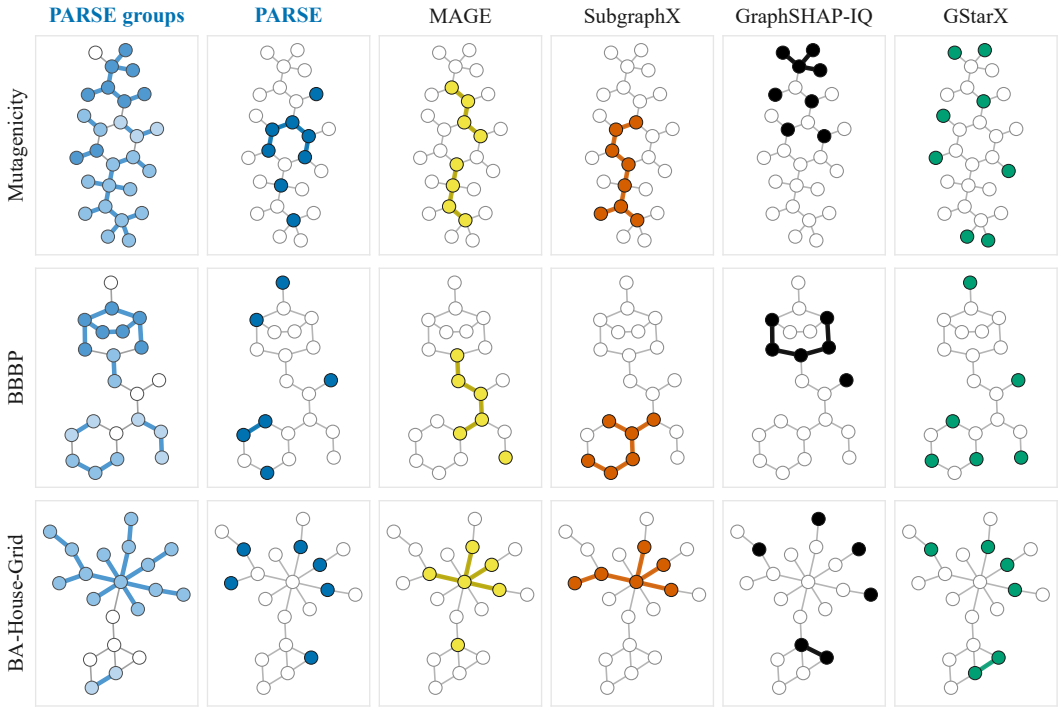


Figure 6: Nodes kept at sparsity 0.7 (filled) by PARSE and four baselines, on the test graph of each dataset where PARSE's H-Fidelity margin is closest to its median. First column: PARSE's partition, each group of more than one node tinted blue with thick inner edges.

Table 11: Fidelity of the node sets of Figure 6 (k kept nodes of n). Fid^+: higher is better, negative when removing the set raises the predicted-class probability; Fid^-: closer to zero is better.

	Mutagenicity ($k = 8, n = 27$)		BBBP ($k = 6, n = 21$)		BA-House-Grid ($k = 6, n = 20$)	
Method	Fid^+	Fid^-	Fid^+	Fid^-	Fid^+	Fid^-
PARSE	0.958	0.083	0.460	−0.230	0.267	0.269
MAGE	0.532	0.184	−0.230	0.439	0.239	0.381
SubgraphX	0.663	0.041	−0.157	−0.146	0.234	0.337
GraphSHAP-IQ	0.005	0.699	0.134	−0.230	0.190	0.284
GStarX	−0.002	0.597	−0.049	−0.221	0.232	0.249

E.7 Per-method means

Tables 12–14 give every method's mean on each metric over all the graphs that method explained; bold marks the best of these means. Where coverage differs, this own-graph view can disagree with the paired view of Appendix E.1: SAME explained 29 of the 50 BA-House-Grid graphs and has the higher own-graph Fid^+ and characterization there, while PARSE is better on the graphs both explained.

Table 12: Per-method means on the public test benchmark, each method on all the graphs it explained; Fid$^-$ is the signed mean and comp. the mean number of connected components of the node sets. Bold: best in the dataset (highest; for Fid$^-$, closest to zero).

Dataset	Method	graphs	H-Fid.	Fid$^+$	Fid$^-$	Char.	Fid$_\alpha$	comp.
BACE	**PARSE**	17	**0.6180**	**0.7463**	−0.1446	**0.8433**	**0.4023**	5.14
	Grad×Input	17	0.5041	0.1679	0.1870	0.2323	−0.0283	3.29
	GNNExplainer	17	0.5076	0.1500	0.1493	0.2384	0.0659	4.51
	PGExplainer	17	0.4980	0.1030	0.1858	0.1739	−0.0048	2.28
	GraphSVX	17	0.5546	0.4136	0.0745	0.4648	0.1633	5.36
	SAME	17	0.5565	0.3562	−0.0320	0.4535	0.1181	2.16
	GraphEXT	17	0.5829	0.5600	**−0.0240**	0.6680	0.3543	6.45
	GStarX	17	0.5740	0.4817	−0.0451	0.5880	0.3422	6.20
	GraphSHAP-IQ	17	0.5671	0.5215	0.0578	0.5984	0.2755	5.14
	MAGE	17	0.5055	0.1600	0.1682	0.2512	0.0064	2.65
	SubgraphX	17	0.4938	0.0921	0.1713	0.1702	−0.0286	1.00
	GECo	17	0.5307	0.2280	0.0377	0.3028	0.1184	1.56
BBBP	**PARSE**	36	**0.5838**	**0.5238**	−0.0929	**0.6437**	**0.2377**	4.94
	Grad×Input	36	0.4902	0.0317	0.1001	0.1001	−0.0143	2.50
	GNNExplainer	36	0.5000	0.0491	0.0453	0.1144	0.0134	3.46
	PGExplainer	36	0.5010	0.0533	0.0578	0.1135	0.0282	2.17
	GraphSVX	36	0.5288	0.2017	0.0133	0.2547	0.0927	3.45
	SAME	36	0.5216	0.1333	−0.0102	0.2041	0.0495	1.54
	GraphEXT	36	0.5483	0.2932	−0.0747	0.3810	0.2072	5.47
	GStarX	36	0.5457	0.2801	−0.0749	0.3657	0.1994	5.72
	GraphSHAP-IQ	36	0.5428	0.2785	−0.0421	0.3455	0.1922	4.50
	MAGE	36	0.4825	0.0187	0.1660	0.1161	−0.0301	1.88
	SubgraphX	36	0.4910	0.0090	0.0785	0.0945	−0.0100	1.02
	GECo	36	0.5164	0.1210	**−0.0024**	0.1814	0.0717	1.47
Benzene	**PARSE**	50	**0.5655**	**0.5575**	**0.1473**	**0.4622**	**0.0720**	2.58
	Grad×Input	50	0.5044	0.3981	0.3654	0.1743	0.0155	2.19
	GNNExplainer	50	0.4886	0.3165	0.4051	0.1319	−0.0022	2.97
	PGExplainer	50	0.4865	0.2949	0.3935	0.1163	0.0046	1.70
	GraphSVX	50	0.5515	0.5117	0.2058	0.3814	0.0562	2.60
	SAME	50	0.5470	0.4887	0.2047	0.3505	0.0624	1.18
	GraphEXT	50	0.5364	0.4792	0.2695	0.3086	0.0477	2.84
	GStarX	50	0.4958	0.3858	0.4135	0.1520	0.0019	3.99
	GraphSHAP-IQ	50	0.5109	0.4284	0.3672	0.1879	0.0319	2.78
	MAGE	50	0.5184	0.4107	0.3052	0.2277	0.0339	1.94
	SubgraphX	50	0.5435	0.4243	0.1761	0.3254	0.0576	1.02
	GECo	50	0.5263	0.3774	0.2381	0.2587	0.0451	1.37
ENZYMES	**PARSE**	50	**0.5535**	**0.4871**	**0.2252**	**0.5261**	**0.1168**	2.64
	Grad×Input	50	0.5100	0.3466	0.3952	0.3529	0.0377	2.93
	GNNExplainer	50	0.4977	0.2820	0.4170	0.2997	0.0065	3.07
	PGExplainer	50	0.4872	0.1811	0.4113	0.2353	−0.0168	1.52
	GraphSVX	50	0.5304	0.4243	0.3339	0.4416	0.0679	2.15
	SAME	50	0.5341	0.3793	0.2582	0.4399	0.1043	1.30
	GraphEXT	50	0.5361	0.4486	0.3119	0.4730	0.1157	2.73
	GStarX	50	0.5193	0.3656	0.3454	0.3934	0.0920	3.08
	GraphSHAP-IQ	50	0.5106	0.3465	0.3992	0.3588	0.0292	2.91
	MAGE	50	0.5254	0.3908	0.3323	0.4116	0.0840	2.18
	SubgraphX	45	0.5197	0.2920	0.2793	0.3527	0.0640	1.01
	GECo	50	0.5176	0.3063	0.3075	0.3657	0.0661	1.47
Graph-SST2	**PARSE**	50	**0.5396**	**0.4132**	0.1562	**0.5077**	**0.1871**	1.51
	Grad×Input	50	0.4917	0.1757	0.2885	0.2556	0.0016	1.18
	GNNExplainer	47	0.4910	0.1467	0.2901	0.2026	0.0244	2.00
	PGExplainer	40	0.4781	0.0530	0.3264	0.0897	−0.0627	1.38
	GraphSVX	50	0.5302	0.3584	0.1833	0.4486	0.1542	1.42
	SAME	40	0.5288	0.2807	**0.1535**	0.3654	0.1820	1.18
	GraphEXT	50	0.5279	0.3478	0.1820	0.4291	0.1634	1.75
	GStarX	50	0.5144	0.2909	0.2237	0.3554	0.1141	2.34
	GraphSHAP-IQ	47	0.5217	0.3179	0.2173	0.3903	0.1363	2.15
	MAGE	47	0.5127	0.2663	0.2251	0.3414	0.1043	1.60
	SubgraphX	50	0.5126	0.2415	0.1838	0.3219	0.1094	1.06
	GECo	50	0.5117	0.2344	0.1914	0.3098	0.1156	1.40

Table 13: Per-method means on the public test benchmark (continued), each method on all the graphs it explained; Fid$^-$ is the signed mean and comp. the mean number of connected components of the node sets. Bold: best in the dataset (highest; for Fid$^-$, closest to zero).

Dataset	Method	graphs	H-Fid.	Fid$^+$	Fid$^-$	Char.	Fid$_\alpha$	comp.
Graph-SST5	**PARSE**	50	**0.5390**	**0.4004**	0.2494	**0.5007**	**0.1377**	1.86
	Grad×Input	50	0.5209	0.3142	0.2968	0.4090	0.0798	1.36
	GNNExplainer	50	0.4954	0.1826	0.3578	0.2609	0.0237	3.58
	PGExplainer	50	0.4783	0.0756	0.3837	0.1328	−0.0704	2.19
	GraphSVX	50	0.5336	0.3624	0.2448	0.4677	0.1272	1.48
	SAME	50	0.5259	0.2961	**0.2336**	0.4077	0.1059	1.12
	GraphEXT	50	0.5322	0.3751	0.2738	0.4708	0.1238	2.12
	GStarX	50	0.5291	0.3630	0.2790	0.4569	0.1152	2.46
	GraphSHAP-IQ	50	0.5253	0.3432	0.2826	0.4378	0.0968	2.81
	MAGE	50	0.5275	0.3447	0.2719	0.4432	0.1053	1.98
	SubgraphX	50	0.5233	0.2886	0.2395	0.3950	0.1133	1.00
	GECo	50	0.5117	0.2282	0.2726	0.3262	0.0788	1.74
Graph-Twitter	**PARSE**	50	**0.5328**	**0.2325**	**0.0512**	**0.3461**	**0.1476**	3.70
	Grad×Input	50	0.5014	0.0928	0.1616	0.1665	0.0269	2.16
	GNNExplainer	50	0.5002	0.0876	0.1683	0.1554	0.0311	3.54
	PGExplainer	50	0.4870	0.0363	0.2332	0.0826	−0.0437	1.89
	GraphSVX	50	0.5090	0.1338	0.1584	0.2158	0.0441	3.77
	SAME	48	0.5068	0.1204	0.1605	0.1955	0.0311	2.71
	GraphEXT	50	0.5253	0.1975	0.0732	0.2946	0.1218	4.36
	GStarX	50	0.5235	0.1877	0.0744	0.2849	0.1170	4.59
	GraphSHAP-IQ	50	0.5274	0.2039	0.0568	0.3058	0.1267	3.18
	MAGE	50	0.5213	0.1798	0.0841	0.2730	0.0994	2.54
	SubgraphX	50	0.5137	0.1479	0.1140	0.2370	0.0840	1.00
	GECo	50	0.5130	0.1427	0.1219	0.2275	0.0781	1.75
Mutagenicity	**PARSE**	50	**0.6120**	**0.8155**	**0.0136**	**0.8464**	**0.2786**	5.93
	Grad×Input	50	0.5130	0.3709	0.3486	0.3566	−0.0038	4.26
	GNNExplainer	50	0.5098	0.3113	0.3284	0.3311	0.0548	5.32
	PGExplainer	50	0.4932	0.2229	0.3766	0.2566	0.0128	3.35
	GraphSVX	50	0.5457	0.5055	0.2896	0.5111	0.0519	3.98
	SAME	50	0.5554	0.4563	0.1124	0.5345	0.1371	1.48
	GraphEXT	50	0.5507	0.5474	0.2343	0.5443	0.1133	6.75
	GStarX	50	0.4889	0.2208	0.3880	0.2554	−0.0967	7.60
	GraphSHAP-IQ	50	0.5031	0.3149	0.4067	0.3176	−0.0044	5.82
	MAGE	49	0.5222	0.3588	0.2974	0.3921	0.0140	2.03
	SubgraphX	47	0.5333	0.3402	0.1776	0.4004	0.0709	1.02
	GECo	50	0.5371	0.3967	0.1991	0.4643	0.1353	1.76
PROTEINS	**PARSE**	49	**0.5392**	**0.3191**	**0.1373**	**0.4215**	**0.0826**	3.55
	Grad×Input	50	0.5011	0.1606	0.2414	0.2421	0.0128	3.53
	GNNExplainer	50	0.4970	0.1352	0.2521	0.2101	−0.0029	4.35
	PGExplainer	50	0.5009	0.1413	0.2234	0.2305	0.0142	2.40
	GraphSVX	50	0.5152	0.2297	0.2219	0.3108	0.0379	3.90
	SAME	47	0.5199	0.2374	0.1969	0.3316	0.0628	1.64
	GraphEXT	49	0.5172	0.2378	0.2102	0.3172	0.0658	3.82
	GStarX	50	0.5031	0.1701	0.2360	0.2391	0.0317	4.90
	GraphSHAP-IQ	49	0.5038	0.1759	0.2414	0.2507	0.0096	3.06
	MAGE	48	0.5100	0.1938	0.2135	0.2797	0.0446	2.24
	SubgraphX	36	0.5091	0.1535	0.1548	0.2278	0.0309	1.01
	GECo	50	0.5115	0.1741	0.1794	0.2579	0.0470	1.71

Table 14: Per-method means on the public test benchmark (continued), each method on all the graphs it explained; Fid$^-$ is the signed mean and comp. the mean number of connected components of the node sets. Bold: best in the dataset (highest; for Fid$^-$, closest to zero).

Dataset	Method	graphs	H-Fid.	Fid$^+$	Fid$^-$	Char.	Fid$_\alpha$	comp.
BA-2Motifs	**PARSE**	50	**0.5615**	**0.4226**	**0.0736**	**0.5393**	**0.0501**	2.46
	Grad×Input	50	0.5450	0.3448	0.1263	0.4772	0.0453	1.10
	GNNExplainer	50	0.5001	0.2390	0.3495	0.3095	0.0001	4.07
	PGExplainer	50	0.4920	0.1750	0.3439	0.2278	−0.0135	2.64
	GraphSVX	50	0.5484	0.3426	0.0839	0.4860	0.0464	3.09
	SAME	50	0.5458	0.3594	0.1347	0.4892	0.0464	1.12
	GraphEXT	50	0.5389	0.3537	0.2019	0.4787	0.0470	2.78
	GStarX	50	0.5114	0.3138	0.3442	0.3790	0.0041	3.82
	GraphSHAP-IQ	50	0.5045	0.2662	0.3528	0.3505	−0.0023	4.62
	MAGE	50	0.5472	0.3459	0.0989	0.4887	0.0452	1.37
	SubgraphX	50	0.5430	0.3319	0.1270	0.4564	0.0351	1.00
	GECo	50	0.5257	0.3081	0.2076	0.3965	0.0402	1.46
BA-House-And-Grid	**PARSE**	50	**0.5948**	**0.6744**	0.0213	**0.7643**	**0.4197**	5.16
	Grad×Input	50	0.5183	0.2953	0.2845	0.3675	0.0935	2.95
	GNNExplainer	50	0.5030	0.2011	0.3025	0.2455	0.0324	3.80
	PGExplainer	50	0.4735	0.0688	0.3971	0.0950	−0.0657	2.74
	GraphSVX	50	0.5302	0.3982	0.3047	0.4511	0.1303	2.30
	SAME	50	0.5174	0.3214	0.3269	0.3765	0.0567	1.32
	GraphEXT	50	0.5900	0.6395	0.0228	0.7357	0.4048	5.18
	GStarX	50	0.5894	0.6422	0.0356	0.7311	0.3972	5.32
	GraphSHAP-IQ	50	0.5911	0.6459	**0.0207**	0.7418	0.4135	4.76
	MAGE	50	0.5061	0.2740	0.3386	0.3573	0.0797	1.80
	SubgraphX	50	0.5261	0.4181	0.3331	0.4626	0.1349	1.00
	GECo	50	0.5048	0.2309	0.3161	0.2835	0.0627	1.59
BA-House-Grid	**PARSE**	50	**0.5196**	0.1427	**0.0382**	0.2133	**0.0951**	5.28
	Grad×Input	50	0.4955	0.0375	0.0974	0.1522	0.0113	3.27
	GNNExplainer	50	0.5012	0.0579	0.0939	0.1154	0.0089	3.76
	PGExplainer	50	0.4977	0.0436	0.1157	0.0994	0.0057	2.11
	GraphSVX	50	0.5036	0.0891	0.1284	0.1614	0.0227	2.18
	SAME	29	0.5054	**0.1549**	0.2401	**0.2510**	0.0300	1.26
	GraphEXT	50	0.5156	0.1334	0.0482	0.1972	0.0825	5.36
	GStarX	50	0.5126	0.1110	0.0431	0.1719	0.0703	5.52
	GraphSHAP-IQ	50	0.5126	0.1040	0.0422	0.1662	0.0744	5.23
	MAGE	50	0.4957	0.0626	0.1382	0.1655	0.0225	1.83
	SubgraphX	50	0.4989	0.0788	0.1394	0.1687	0.0168	1.02
	GECo	50	0.4999	0.0544	0.1026	0.1311	0.0200	1.31
BA-House-Or-Grid	**PARSE**	50	**0.5859**	**0.7114**	**0.1664**	**0.7523**	0.3942	4.48
	Grad×Input	50	0.5148	0.3406	0.3701	0.3658	0.1064	2.56
	GNNExplainer	50	0.5008	0.2362	0.3929	0.2823	0.0628	3.62
	PGExplainer	50	0.4902	0.2721	0.5013	0.2458	0.0140	1.54
	GraphSVX	50	0.5139	0.5371	0.5822	0.4371	0.1411	2.47
	SAME	38	0.5168	0.4157	0.4392	0.3998	0.0509	1.59
	GraphEXT	50	0.5440	0.5708	0.3491	0.5446	0.3069	3.86
	GStarX	50	0.5604	0.6116	0.2567	0.6113	0.3959	4.95
	GraphSHAP-IQ	50	0.5599	0.6389	0.2693	0.6144	**0.4100**	4.64
	MAGE	50	0.4998	0.3361	0.5168	0.3482	0.0464	1.48
	SubgraphX	50	0.5046	0.4838	0.5823	0.3974	0.1101	1.00
	GECo	50	0.4848	0.1952	0.4673	0.2016	0.0402	1.47

F CONTROLLED TASKS

The controlled tasks plant two motifs whose interaction is known and links distant parts of a graph. This section describes how the graphs are built, why the predictors follow the planted rule, how PARSE's two rankings are set, and how each reported quantity is computed.

F.1 THE GRAPHS

Each graph starts from a *scaffold*, an unlabelled tree with 9 or 10 nodes whose diameter is large enough for the separation rule below. A 7-node gadget is attached by one edge to each of two leaves at the ends of a longest path of the tree. A gadget holds a 3-cycle (motif A), a 4-cycle (motif B) or a 6-cycle (a decoy), plus a tail that brings it to 7 nodes, so every graph has 23 or 24 nodes whichever motifs are present, and all nodes carry the same feature. Because the gadgets sit at the two ends of a longest path, any connected node set that holds both motifs covers more than half of the graph.

Each placement of the gadgets yields the four presence states of A and B, with A on either side, and isomorphic duplicates are removed. The graphs built on one scaffold form a *family*, and whole families are assigned to training, validation or test. The three splits hold 40, 14 and 24 families with 269, 86 and 163 graphs. The test split holds 46 graphs with A only, 46 with B only, 46 with both and 25 with neither, and on the 46 graphs with both motifs the motifs are 6 to 10 hops apart. The three tasks share these graphs and differ only in their label rule: a graph is in class 1 under AND when both motifs are present, under OR when at least one is, and under XOR when exactly one is; the decoy never affects the label.

F.2 PREDICTORS

Each task has its own predictor: a GIN-type network (Xu et al., 2019) with random-walk structural encoding (Dwivedi et al., 2022), with 8 layers of width 48 and 6 random-walk steps, chosen on validation graphs among three candidates that differ in depth and encoding. It is trained with Adam (learning rate 10^{-3}, weight decay 10^{-5}, batch size 64) for up to 240 epochs. The training data are the training graphs and masked copies of them, each labeled by the task's rule applied to the motifs that remain intact. The predictor's response to masking therefore follows the rule, so the planted interaction is also the interaction in the game that PARSE explains, which answers the concern of Faber et al. (2021) that a model need not use the structure planted in its data. A predictor qualifies only if, on the validation graphs, it reaches a balanced accuracy of at least 0.9 with a false-positive rate of at most 0.1, on full and on masked graphs, and if masking A, B or both brings its class-1 probability within 0.2 of the value the rule prescribes. All three selected predictors qualify, and each classifies every one of the 163 test graphs correctly.

F.3 PARSE ON THE TASKS

On these tasks PARSE's interactions are read at two levels. The group ranking asks which pair of connected groups interacts most, and the extended node ranking asks which nodes matter once their pair and group effects are credited to them. Both start from the matrix $\hat{\mathbf{M}}$ of Phase 1, whose diagonal holds the nodes' Shapley values and whose off-diagonal holds the pairwise SII, the same matrix from which the public-benchmark explanations are built.

PARSE runs with $m = 50$ orderings and the partition settings $\theta = (\gamma_H, \gamma_S, p_{\mathrm{cov}}, \mu_c, \mu_\beta) = (0.4, 0.3, 1, 1, 0.5)$; since all nodes share one feature, a masked node gets feature 0 and loses its edges. The explained class is the predictor's decision on the full graph; on graphs with both motifs it is class 1 under AND and OR and class 0 under XOR, which fixes the expected sign $\sigma^\star$ of the interaction between A and B: positive under AND, negative under OR, and positive under XOR, where the two motifs together restore class 0. The motif sign reads the Phase-1 matrix $\hat{\mathbf{M}}$ alone.

The group ranking partitions the graph again from $\hat{\mathbf{M}}$ with its own settings (Table 15; $p_{\mathrm{cov}} = 1$ and the annealing of Table 3), splits every group of more than $\lfloor c_{\max} n \rfloor$ nodes into connected pieces, and plays a game whose players are the groups. The SII of each pair of groups is computed exactly when there are at most 8 groups and is otherwise estimated from 32 random contexts per pair. Pairs are ranked by $\max(0, |\hat{I}| - \widehat{\mathrm{se}})$, with $\widehat{\mathrm{se}} = 0$ for an exact value.

The extended node ranking is a weighted sum of four channels, each divided by its maximum over the nodes: $|\hat{\phi}_i|$ and the effects of single nodes, of pairs and of groups. These effects are re-estimated within a budget of 2,048 masked graphs per graph: node effects along random orderings of all nodes, and the effects of the adjacent pairs with the largest $|\hat{M}_{ij}|$ and of small connected groups on random subsets of the other nodes. Each effect is shrunk to a lower confidence bound, its absolute mean minus ζ standard errors (at least 0), mixed with weight λ with the same bound on its signed mean, and scaled by the consistency of its sign across samples. A pair's effect is shared by its two nodes and a group's effect is divided by its size to the power η; each node averages what it receives.

Both rankings are calibrated on the 14 validation families with the optimizer of Appendix A.2, and one setting is shared by the three tasks. The group objective, on the 24 validation graphs with both motifs, rewards strict and exact matches at the top of the ranking and penalizes coalescence, fragmentation and mixing (Appendix F.4); the node objective rewards the node AUC on the validation

graphs whose class the motifs decide. Table 15 gives the search ranges and the adopted values; the pair channel receives the largest weight.

Table 15: Settings of PARSE's two rankings on the controlled tasks: search ranges and adopted values, shared by the three tasks. Symbols as in Appendix F.3.

Ranking	Setting	Range	Adopted
Group ranking	γ_H	$[0.1, 1]$	0.570
	γ_S	$[0.1, 3]$	1.250
	μ_c	$[0.5, 2]$	1.541
	μ_β	$[0.2, 0.8]$	0.262
	$c_{\max}$	$[0.2, 0.5]$ or no cap	0.258
Node ranking	weights: $\lvert\hat{\phi}_i\rvert$, node, pair, group	simplex	$0.347, 0.219, 0.417, 0.017$
	shrinkage ζ	$[0, 3]$	0.435
	share λ of the signed bound	$[0, 1]$	0.131
	group-size power η	$[0, 1.5]$	0.111

F.4 Scored quantities

A pair of groups (C, C') is *correct* if its rank score is positive and its estimated SII has sign $\sigma^\star$. Its *overlap* is the better, over the two ways of assigning the groups to the motifs, of the smaller intersection-over-union, for example $\min(\mathrm{IoU}(C, A), \mathrm{IoU}(C', B))$. A correct pair is a *soft* match if its overlap is at least 0.5, a *strict* match if, in addition, one group contains all of A and the other all of B, and an *exact* match if its overlap is 1. Top-1 counts the graphs whose highest-ranked pair is a match. The motif sign is correct if $\sum_{i\in A,\, j\in B} \hat{M}_{ij}$ has sign $\sigma^\star$. Node AUC is the area under the ROC curve of the extended node ranking against the nodes of the motifs present, and the motif nodes are ranked first on a graph when this AUC is 1.

Three quantities describe the partition that the group ranking uses: *coalescence* is 1 if some group contains both whole motifs; *mixing* is the largest, over groups, of the smaller of the group's recall of A and of B; and *fragmentation* is one minus the largest, over two distinct groups, of the smaller of the first group's recall of A and the second's recall of B, so it is 0 exactly when A and B lie whole in two different groups. All averages weight scaffold families equally.

F.5 Further results

On the XOR task, the summed pairwise SII between the two motifs is positive on all 46 graphs with both motifs and averages $+4.20$. Whenever the top pair holds both whole motifs, its two groups are 2 to 10 hops apart and never adjacent. On graphs with a single motif, that motif alone decides the class under OR and XOR, and PARSE ranks its nodes first on every such graph (node AUC 1.000).