RDS-EXPLAINER: RANDOM-REFERENCE DUAL SCORE FOR SIZE-ADAPTIVE GNN EXPLANATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Understanding why a graph neural network (GNN) makes a prediction is important for trustworthy graph learning. Many post-hoc GNN explainers identify important nodes or edges, but isolated importance values do not reveal which elements together form the prediction-relevant structure. A concrete explanatory subgraph makes this structure explicit, but its size is difficult to determine: too small may omit important evidence, whereas too large may include irrelevant structure, and the appropriate size varies across graphs. To address this problem, we propose RDS-Explainer (Random-Reference Dual-Score Explainer), which evaluates candidate subgraphs against references constructed from random subgraphs of different sizes. The resulting support and dispensability surpluses form the Dual Score, which identifies just-sufficient explanations by encouraging sufficient predictive evidence within the candidate subgraph and suppressing residual evidence outside it, without requiring a predefined explanation size. To search the exponential edge-subset space, Cooperative Edge Pruning removes multiple edges per step and coordinates parallel node-level pruning decisions through graph-level feedback. Once trained, the pruning policy applies directly to unseen graphs without per-instance optimization, enabling millisecond-level explanation generation. Across four molecular benchmarks, RDS-Explainer achieves the highest Charact and GEA, even when six baselines receive privileged ground-truth size information. Code will be publicly available after acceptance.

1 INTRODUCTION

Graph neural networks (GNNs) are widely used for relational prediction, from molecular modeling Gilmer et al. (2017) and materials modeling Merchant et al. (2023) to recommendation systems Fan et al. (2019); Pal et al. (2020) and fraud detection Li et al. (2025a). An accurate prediction, however, does not reveal which atoms, bonds, or relations drove it, and explainability has accordingly become a central dimension of trustworthy graph learning Zhang et al. (2024a): for users seeking to understand the basis of a prediction, a label or score alone is insufficient—what is needed is an explicit indication of the prediction-relevant substructure on which the model relied.

Post-hoc GNN explanation has developed into an active field for identifying the graph components or substructures relevant to a model's prediction Ma et al. (2026). Given a trained GNN and a prediction to explain, many post-hoc methods estimate the importance of nodes or edges to locate prediction-relevant structure. GNNExplainer learns an edge mask for each graph Ying et al. (2019), while PG-Explainer learns a parameterized generator of edge masks Luo et al. (2020). GraphEXT instead assigns importance to nodes Wu et al. (2025). For these importance-based methods, node- or edge-level importance does not by itself determine a unique explanatory subgraph. A concrete subgraph can be easier for users to interpret because it presents the important elements as a recognizable structure rather than as isolated importance values. For example,

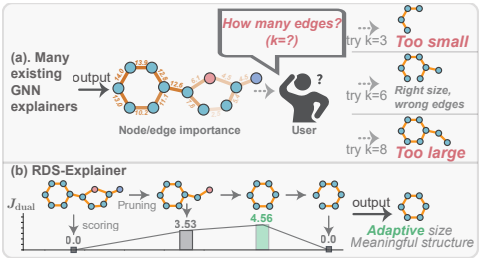


Figure 1: Comparison of explanation workflows on a Benzene example.

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.

in the Benzene task, a complete benzene ring has a direct interpretation because graphs containing such a ring are classified as positive. Forming such a subgraph therefore requires deciding how much of the graph the explanation should contain.

In practice, users often do not know in advance how large an explanation should be Zhang et al. (2024b). A recent survey likewise treats prior knowledge of explanation size as a distinct usability requirement Li et al. (2025b). The choice matters because an explanation that is too small can omit important elements, whereas one that is too large can include irrelevant information Amara et al. (2022). Figure 1(a) illustrates this problem with different choices of k on the same graph. Several methods have tried to avoid asking users to specify the explanation size in advance. AMExplainer learns an edge mask to produce an explanation without a predefined edge count, but it does not directly assess whether the resulting subgraph is too small or too large for that graph Zhang et al. (2024b). ZORRO grows an explanation until a user-defined fidelity threshold is reached Funke et al. (2023). The resulting explanation size therefore depends on how this threshold is set. EiG-Search goes further by searching the explanation size for each instance using candidates formed from an edge-importance ranking Lu et al. (2024). However, it considers only one candidate at each size, so a better subgraph with the same number of edges but a different composition may never be considered.

Determining the explanation size for each graph requires judging whether a subgraph provides an appropriate explanation. Prior work suggests that both the evidence retained in an explanation and the evidence left outside provide useful information about its quality Amara et al. (2022); Tan et al. (2022). We therefore propose RDS-Explainer (Random-Reference Dual-Score Explainer), which extends this two-sided view with size-dependent random references. Its support surplus evaluates the predictive evidence retained by a subgraph relative to a random-expansion reference, while its dispensability surplus evaluates the useful evidence left outside relative to a random-reduction reference. Together, they form the Dual Score used to evaluate and select an explanatory subgraph. The explanation and its size are therefore selected together rather than being specified in advance. By comparing the Dual Scores of candidate subgraphs, the most appropriate candidate in terms of both structure and size can be selected, as illustrated in Figure 1(b).

The Dual Score provides a way to evaluate a subgraph, but a graph with $|\mathcal{E}|$ edges has $2^{|\mathcal{E}|}$ possible edge subsets, making exhaustive evaluation infeasible. We therefore introduce Cooperative Edge Pruning, a reinforcement-learning-based search framework that starts from the full graph and can remove multiple edges at each step, reducing the search to a short sequence of subgraphs that serve as candidate explanations. At each step, the same actor network scores all nodes in parallel, and the scores of the two endpoints jointly determine whether an edge is retained. The Dual Score of the resulting subgraph provides graph-level feedback for training a critic network, which evaluates the combined pruning decisions and guides the actor toward better pruning strategies. Once trained, the actor can be applied directly to unseen graphs without per-instance optimization, enabling millisecond-level explanation generation.

Our contributions are summarized as follows.

- We propose a Random-Reference Dual Score that evaluates both retained and excluded predictive evidence, identifying just-sufficient explanations without requiring a user-specified explanation size.
- We develop Cooperative Edge Pruning, which removes multiple edges per step under graph-level feedback. Once trained, it requires no per-instance optimization and enables millisecond-level explanation generation on unseen graphs.
- We evaluate RDS-Explainer on four molecular benchmarks without prescribing its final explanation size. Even when six baselines are given privileged ground-truth size information, RDS-Explainer achieves the highest Charact and GEA on all four benchmarks.

2 Related Work

Post-hoc GNN Explanation. Post-hoc GNN explainers extract prediction-relevant evidence from a trained GNN without modifying the target model Longa et al. (2025); Ma et al. (2026). They can be grouped into importance estimation and structure construction. Importance-based methods score nodes, edges, or features using learned masks, local surrogates, or prediction decomposition; examples include GNNExplainer Ying et al. (2019), PGExplainer Luo et al. (2020), GraphLIME Huang

et al. (2023), PGM-Explainer Vu & Thai (2020), DEGREE Feng et al. (2022), and relevant-walk methods Schnake et al. (2022). These scores do not uniquely define a discrete subgraph, so a threshold, Top-K rule, or sparsity level is needed and determines the subgraph's size. Structure-construction methods form subgraphs directly through tree search (SubgraphX Yuan et al. (2021) and HM-Explainer Zhao et al. (2025)), sequential decisions (RG-Explainer Shan et al. (2021), RC-Explainer Wang et al. (2023), and CiRLExplainer Hu et al. (2025)), or GFlowNets (GFlowExplainer Li et al. (2023)). RDS-Explainer belongs to this latter family and evaluates candidate edge sets to identify the best explanatory subgraph.

Set-Level and Interaction-Aware GNN Explanation. When predictions depend on interactions, edge-set explainers model groups of edges, coalitions, or higher-order structures rather than isolated components. Game-theoretic methods estimate joint contributions with Shapley values or cooperative games, including SubgraphX Yuan et al. (2021), GNNShap Akkas & Azad (2024), GStarX Zhang et al. (2022), GISExplainer Xian et al. (2024), and higher-order interaction measures Bui et al. (2024); Muschalik et al. (2025); motif-based methods encode recurring or multi-node patterns Yu & Gao (2023); Ding et al. (2023); Sinha et al. (2025); and structured explainers such as RCExplainer Wang et al. (2023) model dependencies during subgraph construction. RDS-Explainer instead evaluates candidate edge sets with a graph-level Dual Score, while Cooperative Edge Pruning coordinates the joint pruning of interdependent edges and produces a nested candidate sequence.

Adaptive GNN Explanation. Recently, a few post-hoc GNN explainers have begun to address the problem of instance-dependent explanation size; a recent survey identifies prior knowledge of explanation size as a distinct usability requirement Li et al. (2025b). AMExplainer learns adversarially sparse masks, but does not directly test whether the resulting extent is appropriate Zhang et al. (2024b); ZORRO greedily adds components until a user-set RDT-Fidelity threshold is met, so its size depends on that threshold Funke et al. (2023); and EiG-Search ranks edges once and tests top-k prefixes, leaving only one composition at each size Lu et al. (2024). These methods control explanation size indirectly, whereas RDS-Explainer calibrates the *evidence–size trade-off* within each graph and compares each candidate edge set as a whole to select a just-sufficient explanation together with its size.

3 Preliminaries

Graph and GNN. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote an input graph with node set $\mathcal{V}$ and undirected edge set $\mathcal{E}$, where $N = |\mathcal{V}|$ and $M = |\mathcal{E}|$. A trained C-class GNN classifier f_θ produces the distribution $p_\theta(\cdot \mid \mathcal{G})$; its encoder gives node representations $\mathbf{H}^{\mathcal{G}} = g_{\text{enc}}(\mathcal{G})$, with $\mathbf{h}_i^{\mathcal{G}}$ denoting the representation of node i. The predicted label is $\hat{y}_{\mathcal{G}} = \arg\max_{c \in \{1,\dots,C\}} p_\theta(c \mid \mathcal{G})$.

Subgraphs and prediction response. For a nonempty edge set $\mathcal{E}_S \subseteq \mathcal{E}$, let $\mathcal{G}_S = (\mathcal{V}_S, \mathcal{E}_S)$ denote its edge-induced subgraph, where $\mathcal{V}_S$ contains the nodes incident to $\mathcal{E}_S$. To quantify how strongly an edge-induced subgraph $\mathcal{G}_S$ supports the original prediction $\hat{y}_{\mathcal{G}}$, we define its prediction response as $\Phi(\mathcal{E}_S) = \log p_\theta(\hat{y}_{\mathcal{G}} \mid \mathcal{G}_S)$.

Problem 1 (Size-Adaptive GNN Explanation)**.** Given a trained GNN classifier f_θ and an input graph $\mathcal{G}$, size-adaptive GNN explanation seeks an edge-induced subgraph that strongly supports the original prediction while allowing its size to depend on $\mathcal{G}$, with its edge set selected by maximizing a quality function J over nonempty candidate edge sets, $\mathcal{E}_S^\star = \arg\max_{\mathcal{E}_S \subseteq \mathcal{E},\, |\mathcal{E}_S| \geq 1} J(\mathcal{E}_S; \mathcal{G}, f_\theta)$, where $J(\mathcal{E}_S; \mathcal{G}, f_\theta)$ evaluates the quality of a candidate edge set for the target prediction, and $\mathcal{G}_S^\star$ is induced by $\mathcal{E}_S^\star$.

The key to solving Problem 1 is scoring candidate edge sets and searching among them efficiently. We address these two issues in the next two sections.

4 Quality Function Definition

Directly comparing subgraph prediction responses is insufficient for selecting an appropriate explanation because the response is confounded by subgraph size. We therefore first construct size-

conditioned random references in Subsection 4.1 to estimate the response expected at each size under random structural selection. After this size adjustment, Subsection 4.2 evaluates each subgraph from two complementary views: the predictive evidence it retains and the evidence left in its complement. It combines these two views into the Dual Score, which jointly selects the explanatory subgraph and its size. Appendix B gives a worked example of this computation on a Benzene graph.

4.1 Size-Conditioned Random References

The raw prediction response $\Phi(\mathcal{E}_S)$ is not directly comparable across subgraph sizes, because a larger subgraph may receive a higher response simply because it contains more edges, allowing redundant context to overshadow a compact subgraph that preserves the original prediction. We therefore construct size-conditioned random references to estimate the response expected at each size under random structural selection.

To calibrate the prediction response against subgraph size, we first estimate the typical response of randomly selected connected subgraphs at each size. For a given r, the reference response is defined as the expected response of a connected subgraph containing r structural edges under this random selection process. We estimate this expectation by drawing L random connected structural edge sets $\{\mathcal{R}_{r,j}\}_{j=1}^{L}$ with $|\mathcal{R}_{r,j}| = r$, and define the size-conditioned random response as

$$\mu_{\mathcal{G}}(r) = \frac{1}{L}\sum_{j=1}^{L}\Phi(\mathcal{R}_{r,j}). \tag{1}$$

Connectedness keeps each sampled subgraph structurally coherent, so the reference reflects local graph structure at the specified size; sampling details are given in Appendix A.

The size-conditioned response curve provides a random baseline at each edge-set size. For a candidate edge set of size $k = |\mathcal{E}_S|$, we summarize this curve in two directions: expansion from a one-edge random anchor and reduction from the full graph. The former models the response gain associated with retaining additional edges, whereas the latter models the response loss associated with omitting edges from the full graph.

Expansion reference. For $r \geq 2$, $(\mu_{\mathcal{G}}(r) - \mu_{\mathcal{G}}(1))/(r-1)$ measures the average response gain per added edge relative to the one-edge random anchor. We use these gains to define the random-expansion gain rate and, using the one-edge response as the anchor, the reference response at candidate size k:

$$\bar{\delta}^{+} = \max\left(0, \frac{1}{M-1}\sum_{r=2}^{M}\frac{\mu_{\mathcal{G}}(r) - \mu_{\mathcal{G}}(1)}{r-1}\right), \qquad \Phi^{+}_{\mathrm{ref}}(k) = \mu_{\mathcal{G}}(1) + (k-1)\bar{\delta}^{+}. \tag{2}$$

We truncate negative gain rates at zero because a decreasing reference would increase the support surplus of larger subgraphs even when their responses do not improve.

Reduction reference. For $r < M$, $(\Phi(\mathcal{E}) - \Phi(\mathcal{R}_{r,j}))/(M-r)$ measures the average response loss per omitted edge when a random r-edge set is retained from the full graph. We use these losses to define the random-reduction loss rate and, using the full-graph response as the anchor, the reference response at candidate size k:

$$\bar{\delta}^{-} = \frac{1}{M-1}\sum_{r=1}^{M-1}\frac{1}{L}\sum_{j=1}^{L}\frac{\Phi(\mathcal{E}) - \Phi(\mathcal{R}_{r,j})}{M-r}, \qquad \Phi^{-}_{\mathrm{ref}}(k) = \Phi(\mathcal{E}) - (M-k)\bar{\delta}^{-}. \tag{3}$$

Together, these references provide the size-matched baselines for the two surpluses below.

4.2 Random-Reference Dual Score

With the two size-matched reference responses established above, we evaluate a nonempty candidate edge set $\mathcal{E}_S \subseteq \mathcal{E}$ from two complementary perspectives. Let $k = |\mathcal{E}_S|$ and $\mathcal{E}_S^c = \mathcal{E} \setminus \mathcal{E}_S$. The retained edges should provide more predictive response than expected under random expansion at size k, while the omitted edges should cause less response loss than expected under random reduction.

Support surplus. We define the support surplus $J_{\text{sup}}(\mathcal{E}_S)$ as the response of $\mathcal{E}_S$ above the random-expansion reference at the same size:

$$J_{\text{sup}}(\mathcal{E}_S) = \Phi(\mathcal{E}_S) - \Phi^{+}_{\text{ref}}(k) = \Phi(\mathcal{E}_S) - \left[\mu_{\mathcal{G}}(1) + (k-1)\bar{\delta}^{+}\right]. \tag{4}$$

A positive $J_{\text{sup}}(\mathcal{E}_S)$ indicates that $\mathcal{E}_S$ produces a higher response than the random-expansion reference at size k.

Dispensability surplus. The support surplus does not determine whether the excluded edges are dispensable. When $\mathcal{E}_S^c$ is omitted, the actual response loss is $\Phi(\mathcal{E}) - \Phi(\mathcal{E}_S)$. We therefore define the dispensability surplus as the candidate response above the random-reduction reference, or equivalently as the expected response loss under random reduction minus the actual loss:

$$J_{\text{disp}}(\mathcal{E}_S) = \Phi(\mathcal{E}_S) - \Phi^{-}_{\text{ref}}(k) = (M-k)\bar{\delta}^{-} - [\Phi(\mathcal{E}) - \Phi(\mathcal{E}_S)]. \tag{5}$$

A positive $J_{\text{disp}}(\mathcal{E}_S)$ indicates that the actual response loss is smaller than the random-reduction loss expected when $M-k$ edges are omitted, suggesting that $\mathcal{E}_S^c$ is relatively dispensable.

Dual Score aggregation. A just-sufficient explanation should retain enough evidence for the original prediction while leaving little predictive evidence in its complement. We therefore require both surpluses to be positive. We apply the positive-part operation to each surplus and combine them with a stabilized harmonic mean, so that a failed criterion yields zero and one surplus cannot compensate for a weakness in the other. For $k > 1$, we define

$$J_{\text{dual}}(\mathcal{E}_S) = \frac{2[J_{\text{sup}}(\mathcal{E}_S)]_+[J_{\text{disp}}(\mathcal{E}_S)]_+}{[J_{\text{sup}}(\mathcal{E}_S)]_+ + [J_{\text{disp}}(\mathcal{E}_S)]_+ + \epsilon}, \qquad k > 1, \tag{6}$$

where $[x]_+ = \max(0, x)$ and $\epsilon > 0$ provides numerical stability. For $k = 1$, we set $J_{\text{dual}}(\mathcal{E}_S) = [J_{\text{sup}}(\mathcal{E}_S)]_+$, so a one-edge candidate receives a positive Dual Score only when $\Phi(\mathcal{E}_S) > \mu_{\mathcal{G}}(1)$.

For nonempty explanatory edge sets with $k > 1$, J_{dual} satisfies the following property.

Property 1 (Joint positivity and full-graph suppression)**.** *For any edge set $\mathcal{E}_S \subseteq \mathcal{E}$ with $k = |\mathcal{E}_S| > 1$: (i) $J_{\text{dual}}(\mathcal{E}_S) > 0$ if and only if $J_{\text{sup}}(\mathcal{E}_S) > 0$ and $J_{\text{disp}}(\mathcal{E}_S) > 0$; and (ii) $J_{\text{dual}}(\mathcal{E}) = 0$.*

The proof is provided in Appendix I. Since the same score compares candidates across edge-set sizes, maximizing J_{dual} selects the explanatory structure and its size jointly; as shown in Appendix B, J_{dual} peaks at the complete six-edge benzene ring on a Benzene graph and is lower for incomplete sets, redundant extensions, and the full graph.

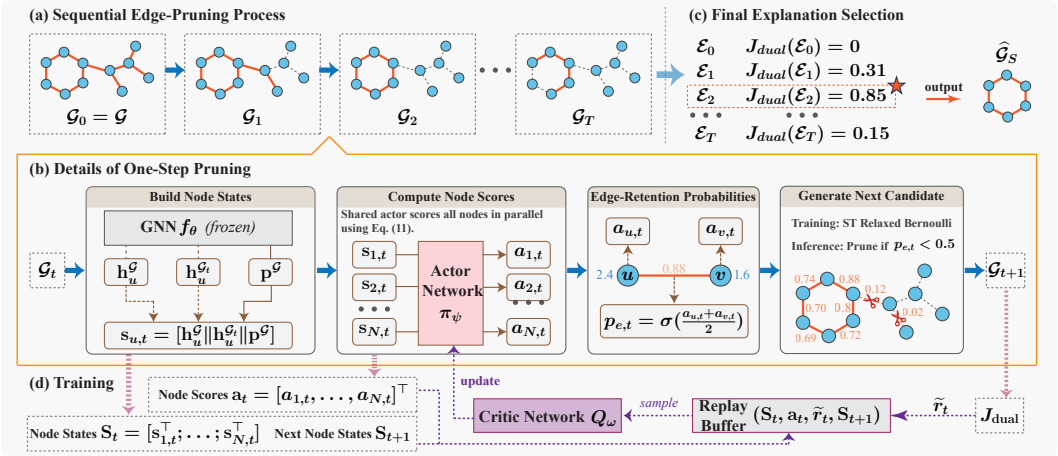


Figure 2: Framework of Cooperative Edge Pruning in RDS-Explainer.

5 COOPERATIVE EDGE PRUNING

The Dual Score $J_{\mathrm{dual}}(\cdot)$ defined in Section 4 evaluates any given edge set, and the explanation induced by the highest-scoring edge set $\widehat{\mathcal{E}}_S$ is taken as the final explanation; we now describe how to find $\widehat{\mathcal{E}}_S$. We formulate the search as sequential edge pruning from the full graph (Fig. 2): starting from $\mathcal{G}_0 = \mathcal{G}$, each step prunes part of the current edge set of $\mathcal{G}_{t-1} = (\mathcal{V}, \mathcal{E}_{t-1})$ to obtain $\mathcal{G}_t = (\mathcal{V}, \mathcal{E}_t)$, yielding the nested candidate sequence $\mathcal{E}_0 \supseteq \mathcal{E}_1 \supseteq \cdots \supseteq \mathcal{E}_T$. This sequence reduces the candidate space from all $2^{|\mathcal{E}|}$ edge subsets to $T+1$ visited candidates, making the combinatorial search in Problem 1 tractable. The pruning process only determines which candidates are visited, while J_{dual} determines which one is returned: the explanatory structure and its size are both decided by the score, not by any budget or stopping rule (Subsection 5.3).

5.1 NODE AGENTS AND EDGE RETENTION

At each step of the sequence, the graph nodes act as agents that collaboratively edit the graph structure, with the goal of progressively pruning the full graph into a compact subgraph that still preserves the original GNN prediction. At step t, each node u obtains a score $a_{u,t}$ from its state $\mathbf{s}_{u,t}$, and the retention probability $p_{e,t}$ of an edge $e = (u, v) \in \mathcal{E}_t$ is determined by the scores of its two endpoints:

$$\mathbf{s}_{u,t} = [\mathbf{h}_u^{\mathcal{G}} \| \mathbf{h}_u^{\mathcal{G}_t} \| \mathbf{p}^{\mathcal{G}}], \tag{7}$$

$$a_{u,t} = \pi_\psi(\mathbf{s}_{u,t}; \hat{y}_{\mathcal{G}}) - \frac{1}{N} \sum_{w \in \mathcal{V}} \pi_\psi(\mathbf{s}_{w,t}; \hat{y}_{\mathcal{G}}), \tag{8}$$

$$p_{e,t} = \sigma\left(\frac{a_{u,t} + a_{v,t}}{2}\right), \qquad e = (u, v) \in \mathcal{E}_t. \tag{9}$$

Here, $\mathbf{h}_u^{\mathcal{G}}$ and $\mathbf{h}_u^{\mathcal{G}_t}$ are computed by the frozen target-GNN encoder g_{enc} and serve as a fixed reference for the original prediction and a description of the structure remaining at step t, respectively; $\mathbf{p}^{\mathcal{G}}$ is the full-graph class-probability vector. The shared actor π_ψ maps the state to a scalar, and the second term in Eq. 8 subtracts the mean score over all nodes, removing the bias shared by every node, so that $a_{u,t}$ expresses the *relative* tendency of node u to retain its incident edges. The retention probability averages the two endpoint scores (σ is the sigmoid function), giving both endpoints equal influence on the edge. This construction further couples the node tendencies through the graph structure: changing the score of any single node changes the retention probabilities of all its incident edges, so whether an edge is retained reflects a joint decision of its two endpoints rather than an independent importance value of either side.

At inference, all current edges are processed simultaneously according to their retention probabilities: edges whose $p_{e,t}$ is no smaller than a threshold τ_{keep} are retained as the next candidate (we use $\tau_{\mathrm{keep}} = 0.5$):

$$\mathcal{B}_t = \{e \mid e \in \mathcal{E}_t,\ p_{e,t} \geq \tau_{\mathrm{keep}}\}, \qquad \mathcal{E}_{t+1} = \mathcal{B}_t. \tag{10}$$

During training, a straight-through relaxed Bernoulli mask is sampled from the same probabilities, producing discrete pruning in the forward pass while letting gradients flow through the continuous relaxation. In both cases, only edges within $\mathcal{E}_t$ can be retained, so the candidate sequence stays nested at every step.

5.2 CENTRALIZED TRAINING OF NODE AGENTS

All nodes share the same reward, namely the Dual Score of the resulting candidate, $r_t = J_{\mathrm{dual}}(\mathcal{E}_{t+1})$. Since this reward evaluates the candidate edge set as a whole rather than the independent contribution of any single edge, we adopt a centralized actor–critic framework Lowe et al. (2017): the critic Q_ω jointly receives the states and scores of all nodes (denoted $\mathbf{S}_t$ and $\mathbf{a}_t$) and estimates the long-term value of the joint scoring; after training, the critic is discarded and only the actor is used at inference, i.e., centralized training with decentralized execution. The online actor and critic (π_ψ, Q_ω) are each paired with a target copy ($\pi_{\psi'}$, $Q_{\omega'}$), which tracks the online parameters by Polyak averaging. The reward $\widetilde{r}_t$ actually used in training may additionally apply size-change shaping on top of r_t, followed by soft clipping and per-predicted-class normalization; these training-only transformations do not affect candidate selection, which always uses the original J_{dual}. The

above training details—the construction of $\mathbf{S}_t, \mathbf{a}_t$, the target-network update rule, and the reward processing—are collected in Appendix C.

The critic and the actor are updated with the following objectives:

$$\mathcal{L}_Q = \mathbb{E}\Big[\big(Q_\omega(\mathbf{S}_t, \mathbf{a}_t) - \widetilde{r}_t - \gamma \mathbf{1}\{t < T-1\} Q_{\omega'}(\mathbf{S}_{t+1}, \mathbf{a}'_{t+1})\big)^2\Big], \tag{11}$$

$$\mathcal{L}_\pi = -\mathbb{E}_{\mathbf{S}_t}\big[Q_\omega\big(\mathbf{S}_t, \mathbf{a}_\psi(\mathbf{S}_t)\big)\big]. \tag{12}$$

Here, γ is the discount factor; $\mathbf{1}\{t < T-1\}$ removes the future value at the final pruning step; and $\mathbf{a}_\psi(\cdot)$ applies π_ψ node-wise with the same centering as in Eq. 8, while $\mathbf{a}'_{t+1}$ is the corresponding score given by the target actor on the next state. When updating the actor, ω is fixed and gradients pass through the critic to ψ. Because the critic evaluates the scores of all nodes under the same reward, the actor learns coordinated scores rather than independent node or edge importance values.

5.3 Candidate Selection and Inference

At inference, the trained actor executes a single pruning rollout for each test graph, the frozen target GNN and J_{dual} evaluate every candidate visited along the sequence, and the final edge set is chosen by

$$\hat{t} = \max\left(\arg\max_{0 \le t \le T} J_{\mathrm{dual}}(\mathcal{E}_t)\right), \qquad \widehat{\mathcal{E}}_S = \mathcal{E}_{\hat{t}}. \tag{13}$$

If several candidates attain the same maximal score, the latest one is selected; nestedness guarantees that it is no larger than any earlier tied candidate, favoring more compact explanations at equal quality. The edge-induced subgraph of $\widehat{\mathcal{E}}_S$ is returned as the final explanation. The whole inference involves neither the critic nor any per-instance re-optimization of the explainer parameters: trained once, RDS-Explainer applies directly to unseen graphs (the training procedure is summarized in Algorithm 2 in Appendix C).

6 Experiments

Experimental Setup. We evaluate RDS-Explainer on four molecular graph datasets from GraphXAI Agarwal et al. (2023) (Benzene, FluorideCarbonyl, AlkaneCarbonyl, and Mutagenicity) with frozen GIN target models, comparing against six post-hoc explainers plus two size-adaptive baselines, and we report Characterization (Charact), Graph Explanation Accuracy (GEA), and Sparsity; higher Charact and GEA and lower Sparsity are better. The complete setup—including dataset descriptions and statistics, the target GNN architectures, the baseline scope, the metric definitions, and all implementation settings—is provided in Appendix D.

6.1 Explanation Quality

We use two baseline-size protocols, both supplying a graph-specific explanation size to the six baselines while RDS-Explainer receives none: the first grants the baselines privileged ground-truth size information, setting the size to that of the smallest ground-truth explanation of each graph, $k_i^{\mathrm{base}} = \min_{\mathcal{E} \in \zeta_i} |\mathcal{E}|$ (ζ_i is the set of valid ground-truth explanations of graph i); the second evaluates all methods at exactly the same explanation size, setting it to the number of edges selected by RDS-Explainer on the same graph and seed, $k_{i,s}^{\mathrm{base}} = |\widehat{\mathcal{E}}_{i,s}^{\mathrm{RDS}}|$.

The first block of Table 1 shows that even without any size information, RDS-Explainer achieves the highest Charact and GEA on all four datasets; the largest Charact gap appears on Benzene (94.84 versus 13.40 for the strongest baseline), so privileged size information does not translate into comparable faithfulness for the baselines. The second block unifies the explanation size across methods, so the remaining differences reflect only which edges are selected: RDS-Explainer leads by 44.56 and 7.72 Charact points on Benzene and FluorideCarbonyl, ties GraphEXT at 100.00 on AlkaneCarbonyl, and narrowly leads GraphEXT on Mutagenicity; it also leads in GEA on Benzene and Mutagenicity and ranks second on FluorideCarbonyl. Notably, all six baselines attain higher Charact in the second block than in the first—the sizes selected by RDS-Explainer are themselves a more suitable explanation budget, a point that Subsection 6.2 examines directly. One dataset behaves differently: on AlkaneCarbonyl, RDS-Explainer attains perfect Charact but low GEA, and

Table 1: Explanation quality on four molecular datasets with GIN target models.

$k_i^{\text{base}} = \min_{\mathcal{E} \in \zeta_i} |\mathcal{E}|$

	Benzene		FluorideCarbonyl		AlkaneCarbonyl		Mutagenicity	
Method	Charact	GEA	Charact	GEA	Charact	GEA	Charact	GEA
GNNExplainer	0.02 ± 0.04	8.70 ± 0.15	8.51 ± 1.41	3.49 ± 0.49	5.48 ± 3.21	1.00 ± 0.75	43.26 ± 1.54	42.48 ± 2.22
PGExplainer	0.23 ± 0.15	11.86 ± 1.05	62.02 ± 6.11	49.65 ± 3.14	1.37 ± 0.00	0.00 ± 0.00	50.81 ± 0.84	59.71 ± 4.27
GraphEXT	11.09 ± 1.02	50.83 ± 0.52	41.84 ± 3.03	29.14 ± 0.56	52.05 ± 0.00	0.00 ± 0.00	50.07 ± 0.41	62.55 ± 2.13
IDEA	0.56 ± 1.04	16.28 ± 13.15	14.74 ± 7.82	16.77 ± 20.84	29.59 ± 25.88	0.00 ± 0.00	45.04 ± 5.48	41.86 ± 15.04
ProxyExplainer	13.40 ± 18.35	27.01 ± 34.94	24.47 ± 4.34	44.42 ± 1.53	29.59 ± 26.15	0.27 ± 0.61	46.96 ± 4.14	61.77 ± 8.31
RCExplainer	7.16 ± 3.92	50.52 ± 5.53	45.26 ± 4.96	54.32 ± 3.24	30.96 ± 25.91	0.00 ± 0.00	51.41 ± 0.84	44.61 ± 4.93
RDS-Explainer	**94.84 ± 1.46**	**74.33 ± 0.52**	**79.21 ± 4.05**	**55.29 ± 8.06**	**100.00 ± 0.00**	**4.25 ± 1.67**	**93.78 ± 2.26**	**67.49 ± 2.51**

$k_{i,s}^{\text{base}} = |\widehat{\mathcal{E}}_{i,s}^{\text{RDS}}|$

	Benzene		FluorideCarbonyl		AlkaneCarbonyl		Mutagenicity	
Method	Charact	GEA	Charact	GEA	Charact	GEA	Charact	GEA
GNNExplainer	0.79 ± 0.14	16.12 ± 0.54	16.93 ± 1.19	4.60 ± 0.50	74.52 ± 7.48	4.68 ± 0.87	76.74 ± 2.65	44.17 ± 1.45
PGExplainer	5.39 ± 0.99	18.95 ± 1.04	71.49 ± 3.71	54.72 ± 5.47	97.26 ± 0.97	1.08 ± 0.48	91.85 ± 1.89	56.98 ± 4.16
GraphEXT	46.47 ± 1.64	57.63 ± 0.43	49.65 ± 4.57	30.50 ± 0.41	**100.00 ± 0.00**	14.57 ± 0.61	93.33 ± 2.51	63.51 ± 1.87
IDEA	11.09 ± 14.52	27.40 ± 18.30	16.67 ± 14.11	19.87 ± 25.12	98.63 ± 0.97	7.18 ± 1.55	80.89 ± 13.96	46.68 ± 12.37
ProxyExplainer	34.38 ± 46.80	31.78 ± 37.59	46.14 ± 0.95	53.65 ± 2.32	89.86 ± 14.39	6.61 ± 6.92	89.48 ± 10.28	66.51 ± 2.46
RCExplainer	50.28 ± 5.08	62.56 ± 1.44	66.58 ± 4.18	**63.22 ± 4.38**	93.70 ± 11.81	**16.57 ± 1.46**	84.89 ± 2.94	41.19 ± 3.18
RDS-Explainer	**94.84 ± 1.46**	**74.33 ± 0.52**	**79.21 ± 4.05**	55.29 ± 8.06	**100.00 ± 0.00**	4.25 ± 1.67	**93.78 ± 2.26**	**67.49 ± 2.51**

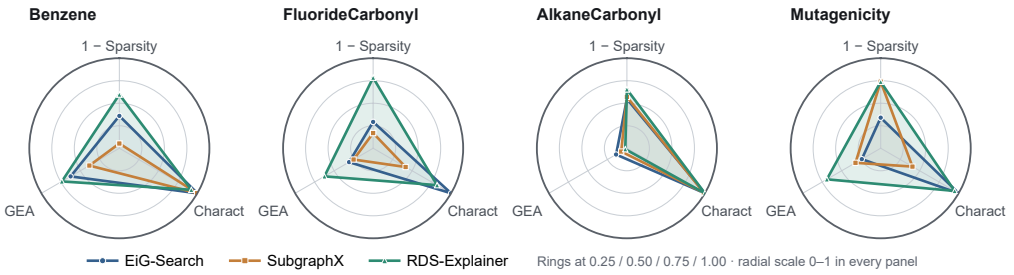


Figure 3: Comparison of RDS-Explainer with two size-adaptive baselines.

even the dataset-provided ground-truth explanation itself reaches a Charact of only 24.66 under the same procedure, showing that these motifs do not consistently support the target GNN's predictions; the GNN most likely exploits shortcut evidence beyond the annotated motifs rather than the motifs themselves. Qualitative explanations for representative molecules are provided in Appendix F.

6.2 Adaptive Sizing

Figure 3 first compares RDS-Explainer with the native adaptive procedures of EiG-Search and SubgraphX, none of which receives an external explanation size (compactness is reported as $1-\text{Sparsity}$, so larger values indicate more edges removed): RDS-Explainer removes more edges than EiG-Search on all four datasets and than SubgraphX on three (nearly tied on Mutagenicity), and attains the highest GEA on Benzene, FluorideCarbonyl, and Mutagenicity. EiG-Search reaches a Charact of 100 on all four datasets, yet its GEA falls below that of RDS-Explainer on three, with AlkaneCarbonyl as the only exception (all three methods score low there, with EiG-Search highest). A high Charact alone thus implies neither a compact nor a ground-truth-aligned explanation (exact values are given in Appendix G).

A more direct, controlled comparison replaces the per-graph explanation sizes with a single size K shared by all test graphs while keeping the learned edge scores unchanged (K is selected by validation Charact, and the Top-K edge groups are retained), testing whether one common size can reproduce the quality and compactness of the per-graph explanations. Table 2 shows that it cannot: on Benzene and Mutagenicity, the fixed-size variant retains 43.37 and 25.29 more percentage points of edges, yet loses 0.94 and 3.71 percentage points in Charact and 32.23 and 36.71 percentage points in GEA, and it loses a further 4.74 and 5.75 Charact points on FluorideCarbonyl and AlkaneCar-

Table 2: Per-graph versus fixed explanation sizes using the same learned edge scores.

	RDS-Explainer			Fixed-size RDS-Explainer		
Dataset	Sparsity ↓	Charact ↑	GEA ↑	Sparsity ↓	Charact ↑	GEA ↑
Benzene	41.78 ± 1.03	94.84 ± 1.46	74.33 ± 0.52	85.15 ± 0.00	93.90 ± 0.11	42.10 ± 0.18
FluorideCarbonyl	23.56 ± 1.72	79.21 ± 4.05	55.29 ± 8.06	26.82 ± 2.07	74.47 ± 4.01	51.76 ± 5.82
AlkaneCarbonyl	34.07 ± 1.73	100.00 ± 0.00	4.25 ± 1.67	35.67 ± 16.81	94.25 ± 1.15	2.82 ± 2.29
Mutagenicity	26.08 ± 2.16	93.78 ± 2.26	67.49 ± 2.51	51.37 ± 7.70	90.07 ± 2.32	30.78 ± 5.08

Table 3: Ablation study on Benzene with GIN target models (percentages, mean $\pm$ std over five explainer seeds). The other three datasets are reported in Appendix E.1.

Dataset	Metric	RDS-Explainer	V1 (J_{sup})	V2 (J_{disp})	V3 (Φ)	V4 (1 step)	V5 (1 edge/step)
Benzene	Charact ↑	94.84 ± 1.46	92.44 ± 3.07	90.53 ± 10.52	99.90 ± 0.22	91.55 ± 0.47	78.33 ± 16.85
	Sparsity ↓	41.78 ± 1.03	41.52 ± 2.22	51.71 ± 19.21	90.80 ± 20.57	45.57 ± 0.47	48.95 ± 4.25
	GEA ↑	74.33 ± 0.52	73.14 ± 1.00	64.90 ± 20.88	42.79 ± 14.25	74.23 ± 0.29	58.79 ± 12.66

bonyl. One fixed size applied to the same edge scores therefore cannot preserve both quality and compactness: it either enlarges the explanations, degrades their quality, or does both.

Appendix H provides additional statistical evidence that the selected sizes vary across graphs, and the full fixed-Sparsity frontier comparison in Appendix E further confirms that RDS-Explainer preserves high quality at more compact explanation sizes.

6.3 Ablation Study

To examine the contributions of the Dual Score and Cooperative Edge Pruning, we design five variants: V1–V3 replace the scoring objective (using only J_{sup}, only J_{disp}, or the raw $\Phi(\mathcal{E}_S)$ without random-reference correction), while V4–V5 modify the pruning (T independent one-step transitions and one focal edge per step, respectively; full definitions in Appendix E.1). Table 3 reports the Benzene results, and the other three datasets show consistent patterns (Table 5). Dropping either surplus degrades Charact and GEA, whereas using Φ directly achieves the highest Charact (99.90) only by retaining most of the graph (Sparsity $41.78 \rightarrow 90.80$, GEA $74.33 \rightarrow 42.79$), confirming that J_{sup} and J_{disp} constrain different aspects of the explanation and that only their combination balances prediction support, explanation size, and the evidence left outside. Both pruning variants are weaker: V4 raises Sparsity on FluorideCarbonyl from 23.56 to 33.74, and V5 drops to 78.33 Charact on Benzene with substantially larger seed variance, showing that parallel multi-edge updates outperform edge-at-a-time updates. Additional perturbation, parameter, cross-architecture, and runtime analyses are reported in Appendix E.

7 Conclusion

This paper presented RDS-Explainer for size-adaptive post-hoc GNN explanation, where the appropriate explanation size is unknown in advance. RDS-Explainer evaluates candidate subgraphs through the Dual Score, which uses size-dependent random references to assess retained predictive support and residual evidence outside the candidate, thereby identifying just-sufficient explanations without requiring a predefined explanation size. Cooperative Edge Pruning further turns complete-candidate quality into coordinated structural pruning, enabling efficient search over a short sequence of candidate subgraphs and direct inference on unseen graphs without per-instance optimization. Across four molecular benchmarks, RDS-Explainer achieves the highest Charact and GEA even when six baselines are given privileged ground-truth size information. Additional experiments support the benefits of graph-specific sizing, just-sufficient explanation selection, and coordinated multi-edge pruning. Future work will extend RDS-Explainer to larger and more heterogeneous graph settings.

AI USE STATEMENT

In this work, we did not use generative AI tools for tasks requiring disclosure under the ICLR 2027 AI Policy for Authors. Additionally, we used generative AI tools to assist with manuscript preparation, including improving grammar, clarity, readability, and academic writing style. All AI-assisted text was carefully reviewed and revised by the authors to ensure technical accuracy and faithful representation of the proposed methodology and experimental results. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REFERENCES

Chirag Agarwal, Owen Queen, Himabindu Lakkaraju, and Marinka Zitnik. Evaluating explainability for graph neural networks. *Sci. Data*, 10(1):144, 2023. doi: 10.1038/s41597-023-01974-x. URL https://www.nature.com/articles/s41597-023-01974-x.

Selahattin Akkas and Ariful Azad. GNNShap: Scalable and accurate GNN explanation using Shapley values. In *WWW*, pp. 827–838, 2024. doi: 10.1145/3589334.3645599. URL https://dl.acm.org/doi/10.1145/3589334.3645599.

Kenza Amara, Zhitao Ying, Zitao Zhang, Zhichao Han, Yang Zhao, Yinan Shan, Ulrik Brandes, Sebastian Schemm, and Ce Zhang. GraphFramEx: Towards systematic evaluation of explainability methods for graph neural networks. In *LoG*, pp. 44:1–44:23, 2022. URL https://proceedings.mlr.press/v198/amara22a.html.

Ngoc Bui, Hieu Trung Nguyen, Viet Anh Nguyen, and Rex Ying. Explaining graph neural networks via structure-aware interaction index. In *ICML*, pp. 4854–4883, 2024.

Zhuomin Chen, Jiaxing Zhang, Jingchao Ni, Xiaoting Li, Yuchen Bian, Md Mezbahul Islam, Ananda Mondal, Hua Wei, and Dongsheng Luo. Generating in-distribution proxy graphs for explaining graph neural networks. In *ICML*, pp. 7712–7730, 2024. URL https://proceedings.mlr.press/v235/chen24bd.html.

Feng Ding, Naiwen Luo, Shuo Yu, Tingting Wang, and Feng Xia. MEGA: Explaining graph neural networks with network motifs. In *IJCNN*, pp. 1–9, 2023. doi: 10.1109/IJCNN54540.2023.10191684. URL https://ieeexplore.ieee.org/document/10191684/.

Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. Graph neural networks for social recommendation. In *WWW*, pp. 417–426, 2019. doi: 10.1145/3308558.3313488.

Qizhang Feng, Ninghao Liu, Fan Yang, Ruixiang Tang, Mengnan Du, and Xia Hu. DEGREE: Decomposition based explanation for graph neural networks. In *ICLR*, 2022.

Thorben Funke, Megha Khosla, Mandeep Rathee, and Avishek Anand. ZORRO: Valid, sparse, and stable explanations in graph neural networks. *IEEE Trans. Knowl. Data Eng.*, 35(8):8687–8698, 2023. doi: 10.1109/TKDE.2022.3201170. URL https://ieeexplore.ieee.org/document/9866587/.

Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *ICML*, pp. 1263–1272, 2017.

Wenya Hu, Jia Wu, and Quan Qian. CiRLExplainer: Causality-inspired explainer for graph neural networks via reinforcement learning. *IEEE Trans. Neural Netw. Learn. Syst.*, 36(6):9970–9984, 2025. doi: 10.1109/TNNLS.2025.3543070. URL https://ieeexplore.ieee.org/document/10925220/.

Qiang Huang, Makoto Yamada, Yuan Tian, Dinesh Singh, and Yi Chang. GraphLIME: Local interpretable model explanations for graph neural networks. *IEEE Trans. Knowl. Data Eng.*, 35(7):6968–6972, 2023. doi: 10.1109/TKDE.2022.3187455. URL https://ieeexplore.ieee.org/document/9811416/.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017. URL https://openreview.net/forum?id=SJU4ayYgl.

Pengbo Li, Hang Yu, and Xiangfeng Luo. Context-aware graph neural network for graph-based fraud detection with extremely limited labels. In *AAAI*, pp. 12112–12120, 2025a. doi: 10.1609/aaai.v39i11.33319.

Wenqian Li, Yinchuan Li, Zhigang Li, Jianye Hao, and Yan Pang. DAG matters! GFlowNets enhanced explainer for graph neural networks. In *ICLR*, 2023.

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

Antonio Longa, Steve Azzolin, Gabriele Santin, Giulia Cencetti, Pietro Liò, Bruno Lepri, and Andrea Passerini. Explaining the explainers in graph neural networks: a comparative study. *ACM Comput. Surv.*, 57(5):120:1–120:37, 2025. doi: 10.1145/3696444. URL https://dl.acm.org/doi/10.1145/3696444.

Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. In *NeurIPS*, pp. 6379–6390, 2017.

Shengyao Lu, Bang Liu, Keith G. Mills, Jiao He, and Di Niu. EiG-Search: Generating edge-induced subgraphs for GNN explanation in linear time. In *ICML*, pp. 33069–33088, 2024.

Dongsheng Luo, Wei Cheng, Dongkuan Xu, Wenchao Yu, Bo Zong, Haifeng Chen, and Xiang Zhang. Parameterized explainer for graph neural network. In *NeurIPS*, pp. 19620–19631, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/e37b08dd3015330dcbb5d6663667b8b8-Abstract.html.

Wenzheng Ma, Xiaofeng Liu, Yihu Liu, and Yinglong Ma. Post-hoc explainability of graph neural networks: A comprehensive survey. *Inf. Sci.*, 740:123202, 2026. doi: 10.1016/j.ins.2026.123202. URL https://linkinghub.elsevier.com/retrieve/pii/S0020025526001337.

Amil Merchant, Simon Batzner, Samuel S Schoenholz, Muratahan Aykol, Gowoon Cheon, and Ekin Dogus Cubuk. Scaling deep learning for materials discovery. *Nature*, 624(7990):80–85, 2023.

Maximilian Muschalik, Fabian Fumagalli, Paolo Frazzetto, Janine Strotherm, Luca Hermes, Alessandro Sperduti, Eyke Hüllermeier, and Barbara Hammer. Exact computation of any-order Shapley interactions for graph neural networks. In *ICLR*, 2025.

Aditya Pal, Chantat Eksombatchai, Yitong Zhou, Bo Zhao, Charles Rosenberg, and Jure Leskovec. PinnerSage: Multi-modal user embedding framework for recommendations at Pinterest. In *KDD*, pp. 2311–2320, 2020. doi: 10.1145/3394486.3403280.

Thomas Schnake, Oliver Eberle, Jonas Lederer, Shinichi Nakajima, Kristof T. Schütt, Klaus-Robert Müller, and Grégoire Montavon. Higher-order explanations of graph neural networks via relevant walks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(11):7581–7596, 2022. doi: 10.1109/TPAMI.2021.3115452. URL https://ieeexplore.ieee.org/document/9547794/.

Caihua Shan, Yifei Shen, Yao Zhang, Xiang Li, and Dongsheng Li. Reinforcement learning enhanced explainer for graph neural networks. In *NeurIPS*, pp. 22523–22533, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/be26abe76fb5c8a4921cf9d3e865b454-Abstract.html.

Akshit Sinha, Sreeram Vennam, Charu Sharma, and Ponnurangam Kumaraguru. Higher order structures for graph explanations. In *AAAI*, pp. 20514–20521, 2025. doi: 10.1609/AAAI.V39I19.34260. URL https://doi.org/10.1609/aaai.v39i19.34260.

Juntao Tan, Shijie Geng, Zuohui Fu, Yingqiang Ge, Shuyuan Xu, Yunqi Li, and Yongfeng Zhang. Learning and evaluating graph neural network explanations based on counterfactual and factual reasoning. In *WWW*, pp. 1018–1027, 2022. doi: 10.1145/3485447.3511948. URL https://dl.acm.org/doi/10.1145/3485447.3511948.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *ICLR*, 2018. URL https://openreview.net/forum?id=rJXMpikCZ.

Minh N. Vu and My T. Thai. PGM-Explainer: Probabilistic graphical model explanations for graph neural networks. In *NeurIPS*, pp. 12225–12235, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/8fb134f258b1f7865a6ab2d935a897c9-Abstract.html.

Xiang Wang, Yingxin Wu, An Zhang, Fuli Feng, Xiangnan He, and Tat-Seng Chua. Reinforced causal explainer for graph neural networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(2):2297–2309, 2023. doi: 10.1109/TPAMI.2022.3170302. URL https://ieeexplore.ieee.org/abstract/document/9763330.

Lijun Wu, Dong Hao, and Zhiyi Fan. Explainable graph neural networks via structural externalities. In *IJCAI*, pp. 583–591, 2025. doi: 10.24963/IJCAI.2025/66. URL https://www.ijcai.org/proceedings/2025/66.

Xingping Xian, Jianlu Liu, Chao Wang, Tao Wu, Shaojie Qiao, Xiaochuan Tang, and Qun Liu. GIS-Explainer: On explainability of graph neural networks via game-theoretic interaction subgraphs. arXiv preprint arXiv:2409.15698, 2024. URL http://arxiv.org/abs/2409.15698.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *ICLR*, 2019. URL https://openreview.net/forum?id=ryGs6iA5Km.

Jun Yin, Senzhang Wang, Ziluowen Luo, Peng Huo, Hao Yan, Hao Miao, Chaozhuo Li, Shirui Pan, and Chengqi Zhang. Paradigm shift of GNN explainer from label space to prototypical representation space. In *ICLR*, 2026. URL https://openreview.net/forum?id=X7eYISNf01.

Zhitao Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. GNNExplainer: Generating explanations for graph neural networks. In *NeurIPS*, pp. 9240–9251, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/d80b7040b773199015de6d3b4293c8ff-Abstract.html.

Zhaoning Yu and Hongyang Gao. MotifExplainer: a motif-based graph neural network explainer. arXiv preprint arXiv:2202.00519, 2023. URL http://arxiv.org/abs/2202.00519.

Hao Yuan, Haiyang Yu, Jie Wang, Kang Li, and Shuiwang Ji. On explainability of graph neural networks via subgraph explorations. In *ICML*, pp. 12241–12252, 2021. URL http://proceedings.mlr.press/v139/yuan21c.html.

He Zhang, Bang Wu, Xingliang Yuan, Shirui Pan, Hanghang Tong, and Jian Pei. Trustworthy graph neural networks: Aspects, methods, and trends. *Proc. IEEE*, 112(2):97–139, 2024a.

Shichang Zhang, Yozen Liu, Neil Shah, and Yizhou Sun. GStarX: Explaining graph neural networks with structure-aware cooperative games. In *NeurIPS*, pp. 19810–19823, 2022.

Wei Zhang, Xiaofan Li, and Wolfgang Nejdl. Adversarial mask explainer for graph neural networks. In *WWW*, pp. 861–869, 2024b. doi: 10.1145/3589334.3645608. URL https://doi.org/10.1145/3589334.3645608.

Yibowen Zhao, Yonghui Xu, Di Wang, Yixin Zhang, Qingzhong Li, and Lizhen Cui. A multi-objective explanation framework for graph neural networks. *IEEE Trans. Knowl. Data Eng.*, 37 (12):6765–6781, 2025. doi: 10.1109/TKDE.2025.3611170. URL https://ieeexplore.ieee.org/document/11168917/.

A Random-Reference Sampling

The random references are constructed from structural undirected edges. Reciprocal directed entries in the GNN's message-passing representation are treated as one structural edge and are retained or removed together. For a target size r, a draw starts from a uniformly selected edge. It then repeatedly

Algorithm 1 Random structural edge-set sampling

Require: Structural edge set $\mathcal{E}$, target size $1 \le r \le M$, requested sample count L
Ensure: Distinct random edge sets of size r

if $r = M$ **then**
  return $\{\mathcal{E}\}$
end if
$\mathcal{S} \leftarrow \varnothing$, $a \leftarrow 0$, $A \leftarrow \max(100, 40L)$
while $|\mathcal{S}| < L$ and $a < A$ **do**
  $a \leftarrow a + 1$; draw e uniformly from $\mathcal{E}$
  $\mathcal{R} \leftarrow \{e\}$; let $\mathcal{V}_{\mathcal{R}}$ contain the endpoints of e
  while $|\mathcal{R}| < r$ **do**
    Concatenate all unselected edges incident to nodes in $\mathcal{V}_{\mathcal{R}}$ into a list F
    if $F \neq \varnothing$ **then**
      Draw e uniformly from the entries of F
    else
      Draw e uniformly from $\mathcal{E} \setminus \mathcal{R}$
    end if
    $\mathcal{R} \leftarrow \mathcal{R} \cup \{e\}$; add the endpoints of e to $\mathcal{V}_{\mathcal{R}}$
  end while
  if $\mathcal{R} \notin \mathcal{S}$ **then**
    Add $\mathcal{R}$ to $\mathcal{S}$
  end if
end while
return $\mathcal{S}$

selects an unselected edge adjacent to a node incident to the selected edges. This favors locally coherent structures over sets assembled from unrelated parts of the graph. If no adjacent unselected edge remains before r edges have been selected, the process restarts from a uniformly selected unselected edge. On a connected input graph, every drawn edge set is connected; on a disconnected input graph, a draw may contain edges from multiple components. The restart guarantees that a draw can reach any size $1 \le r \le M$, but does not make it a connected edge set in that case.

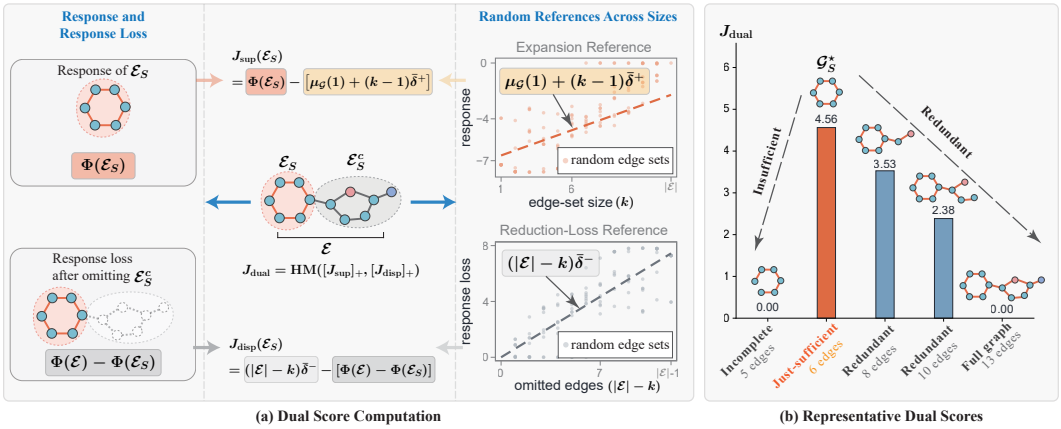
Algorithm 1 specifies the sampling rule. The adjacency list is formed by concatenating the unselected incident edges of all nodes reached so far; an edge incident to two reached nodes can therefore appear twice. Selection is uniform over this list, so the induced distribution is not uniform over distinct connected subgraphs. Repeated edge sets are discarded, with at most $\max(100, 40L)$ draws attempted per size. If fewer than L distinct sets are obtained, the implementation averages the responses of those available; at $r = M$, it directly uses the full edge set as the sole sample. Thus, the L in Eq. 1 denotes the number of retained samples for a size when that number is attainable; otherwise its empirical average uses the actual sample count. Sampling uses a graph-specific pseudorandom generator initialized with seed $42 + 17i$, where i is the graph index. The response of each retained edge set is evaluated on its edge-induced subgraph.

B Worked Example of the Dual Score

Figure 4(a) presents a worked example of the Dual Score computation on a representative Benzene graph: it shows the size-conditioned random references, the resulting support and dispensability surpluses, and the aggregation into J_{dual}.

C Additional Details for Cooperative Edge Pruning

Algorithm 2 summarizes the complete training loop of Cooperative Edge Pruning: for each training graph, it runs a T-step pruning rollout, stores one transition per step, and periodically updates the critic, the actor, and their target copies. The remainder of this section provides the definitions and implementation details involved in this loop. At inference, the trained actor generates the trajectory without exploration or parameter updates, the frozen target GNN and the original J_{dual} score the visited candidates, and Eq. 13 selects the explanation.



(a) Dual Score Computation (b) Representative Dual Scores

Figure 4: Worked example of the Dual Score for a representative Benzene graph. Panel (a) traces the computation of the size-conditioned random references and the two surpluses; panel (b) shows J_{dual} over candidate edge sets, peaking at the complete six-edge benzene ring and decreasing for incomplete sets, redundant extensions, and the full graph.

Algorithm 2 Training Cooperative Edge Pruning

Require: Training graphs $\mathcal{D}_{\text{train}}$, frozen target GNN f_θ, pruning horizon T
Ensure: Trained actor π_ψ

```
Initialize actor π_ψ, critic Q_ω, their target copies, and replay buffer.
repeat
    Sample a batch of training graphs and initialize candidates to their full edge sets.
    for t = 0, ..., T − 1 do
        Compute node states, actor scores, and edge actions using Eqs. 7–9.
        Sample a straight-through mask, retain at least one edge group, and project the mask
    onto the current edge set to form the next candidate.
        Evaluate J_dual(E_{t+1}) on its edge-induced graph and store the transition (Eq. 15).
        if an update is scheduled then
            Sample transitions from replay; update the critic and actor by Eqs. 11–12 and the
    target networks by Eq. 16.
        end if
    end for
until the training budget is exhausted
return π_ψ.
```

Structural edges and candidate graphs. The message-passing representation may store an undirected bond as two reciprocal directed entries. We treat these entries as one structural edge unit and apply the same keep/remove decision to both; candidate size and J_{dual} are computed over structural units. During pruning, the state graph retains all original nodes and masks edges. To evaluate a candidate, we form the edge-induced subgraph on the endpoints of its retained edges and run the frozen target GNN on that subgraph. This keeps the node-aligned policy state well-defined while ensuring that the reward measures the explanation actually returned.

Nonempty candidate guarantee. In each step of the algorithm, at least one edge group is retained after sampling the straight-through mask, so that the next candidate $\mathcal{G}_{t+1}$ is nonempty. The same guarantee applies to the deterministic threshold decoding at inference (Eq. 10): if the thresholded set $\mathcal{B}_t$ is empty, the edge group in $\mathcal{E}_t$ with the highest retention probability is retained.

Training reward processing. By configuration, the training reward may additionally include size-change shaping on top of r_t, namely a shaping term proportional to the candidate-size change between two consecutive steps, which guides the search toward more compact candidates. The reward

is soft-clipped before being written into replay storage and is normalized per predicted class when sampled for an update. The processed value is denoted by $\widetilde{r}_t$ and is the reward used in Eq. 11. These transformations apply only to training; candidate selection and the final explanation always use the original J_{dual} score.

Critic inputs. The critic input is formed by stacking the states and scores of all nodes (d_s is the node-state dimension):

$$\begin{aligned} \mathbf{S}_t &= [\mathbf{s}_{1,t}^\top; \ldots; \mathbf{s}_{N,t}^\top] \in \mathbb{R}^{N \times d_s}, \\ \mathbf{a}_t &= [a_{1,t}, \ldots, a_{N,t}]^\top \in \mathbb{R}^N, \end{aligned} \tag{14}$$

For batched training, $\mathbf{S}_t$ and $\mathbf{a}_t$ are zero-padded to a common maximum number of nodes and then flattened into fixed-length vectors; padded rows are excluded from the graph-wise centering in Eq. 8.

Experience storage. During training, every pruning step is stored in the replay buffer as a transition

$$(\mathbf{S}_t, \mathbf{a}_t, \widetilde{r}_t, \mathbf{S}_{t+1}). \tag{15}$$

These transitions accumulate across training graphs and pruning steps; whenever an update is scheduled, a batch of transitions is sampled from the buffer to update the critic and the actor. Each stored transition additionally records a terminal indicator and the original predicted class, the latter used by the per-class reward normalization described above.

Target networks. The temporal-difference target in Eq. 11 requires the joint score at the next state $\mathbf{S}_{t+1}$, which is supplied by the target actor as $\mathbf{a}'_{t+1} = \mathbf{a}_{\psi'}(\mathbf{S}_{t+1}; \hat{y}_{\mathcal{G}})$, where $\mathbf{a}_{\psi'}(\cdot)$ applies $\pi_{\psi'}$ row-wise to $\mathbf{S}_{t+1}$ and centers the valid node scores exactly as in Eq. 8, so that the online and target scores are calibrated on the same scale and the temporal-difference target remains comparable. After every update of the online networks, their target copies $\pi_{\psi'}$ and $Q_{\omega'}$ track the online parameters by Polyak averaging Lowe et al. (2017):

$$\omega' \leftarrow \tau_{\mathrm{tgt}}\omega + (1 - \tau_{\mathrm{tgt}})\omega', \qquad \psi' \leftarrow \tau_{\mathrm{tgt}}\psi + (1 - \tau_{\mathrm{tgt}})\psi', \tag{16}$$

where $0 < \tau_{\mathrm{tgt}} \leq 1$ controls the tracking rate: the smaller τ_{tgt} is, the more slowly the target networks change, and the more stable the temporal-difference target.

Computational cost. With fixed-depth sparse message passing, a T-step rollout costs $O(T(N + M)d)$ for N nodes, M structural edges, and representation dimension d. Constructing L random-reference samples for each edge-set size costs $O(LM(N + M)d)$ once per graph. After these graph-specific statistics are cached, inference follows only the short pruning trajectory.

D Experimental Details

This section provides the dataset descriptions, metric definitions, baseline scope, and implementation settings used in the experiments.

Moved experimental setup. The experiments use the four molecular graph datasets released with GraphXAI Agarwal et al. (2023), which provide chemically defined ground-truth motifs of varying sizes, and a 70%/10%/20% training/validation/test split. The target models are three-layer GINs Xu et al. (2019) trained separately on each dataset, plus a three-layer GCN Kipf & Welling (2017) and a two-layer GAT Veličković et al. (2018) on Benzene for the cross-architecture evaluation; all target GNNs are trained beforehand and kept frozen during explanation. We compare against six post-hoc GNN explainers—GNNExplainer Ying et al. (2019), PGExplainer Luo et al. (2020), GraphEXT Wu et al. (2025), IDEA Yin et al. (2026), ProxyExplainer Chen et al. (2024), and RCExplainer Wang et al. (2023)—covering mask optimization, learned scoring, representation- and proxy-based explanation, and sequential structural search, and we additionally add EiG-Search Lu et al. (2024) and SubgraphX Yuan et al. (2021) for the adaptive-sizing comparison.

Table 4: Dataset statistics.

Dataset	#Graphs	Avg. $\lvert\mathcal{V}\rvert$	Avg. $\lvert\mathcal{E}\rvert$	Feature Dim.	#Classes
Benzene	12,000	20.58	21.83	14	2
FluorideCarbonyl	8,671	21.36	22.68	14	2
AlkaneCarbonyl	1,125	21.48	22.78	14	2
Mutagenicity	1,768	29.15	30.42	14	2

Datasets. Benzene classifies whether a molecule contains a benzene ring, with each ring serving as a ground-truth explanation. FluorideCarbonyl classifies the co-occurrence of a fluoride atom and a carbonyl group, while AlkaneCarbonyl considers an unbranched alkane together with a carbonyl group. Mutagenicity classifies molecules as mutagenic or non-mutagenic, using NH_2, NO_2, aliphatic halide, nitroso, and azo groups as ground-truth toxicophores. All experiments use a 70%/10%/20% training/validation/test split.

Baseline scope. GNNExplainer optimizes an instance-specific mask, whereas PGExplainer learns a parameterized mask generator. GraphEXT aggregates node scores over edge endpoints; IDEA optimizes explanations in representation space; ProxyExplainer uses proxy graphs; and RCExplainer performs sequential reinforcement-learning-based edge selection.

Evaluation metrics. Charact is the harmonic mean of Fid^+ and $1 - \mathrm{Fid}^-$. Here Fid^+ measures necessity and is high when removing the explanation changes the prediction, while Fid^- measures prediction failure when only the explanation is retained. GEA measures the Jaccard overlap between the explanatory edges and the union of the ground-truth explanations. For explainer seed s, Sparsity is the mean proportion of input-graph edges included in the explanations:

$$\mathrm{Sparsity}_s = \frac{1}{n_{\mathrm{test}}} \sum_{i=1}^{n_{\mathrm{test}}} \frac{\left|\widehat{\mathcal{E}}_{i,s}\right|}{|\mathcal{E}_i|},$$

where n_{test} is the number of test graphs, $\mathcal{E}_i$ is the edge set of test graph i, and $\widehat{\mathcal{E}}_{i,s}$ is its explanatory edge set under seed s. Sparsity is reported as a percentage, with lower values indicating more compact explanations.

Implementation details. For Benzene, FluorideCarbonyl, AlkaneCarbonyl, and Mutagenicity, RDS-Explainer is trained for 3, 5, 3, and 2 epochs, respectively, with $T = 3, 2, 10$, and 1 pruning steps. The actor learning rate is 10^{-3} on Benzene and FluorideCarbonyl and 5×10^{-5} on AlkaneCarbonyl and Mutagenicity; the critic learning rate is 10^{-3} for all datasets. We use $L = 10$ random reference subgraphs per edge-set size, except $L = 5$ on AlkaneCarbonyl. The Benzene GCN and GAT experiments reuse the GIN settings. Inference uses a 0.5 edge-retention threshold and retains at least one edge group. Results are reported as mean $\pm$ sample standard deviation over five explainer seeds unless otherwise stated.

E Additional Experimental Results

E.1 Ablation on Additional Datasets

Table 5 reports the component analysis on FluorideCarbonyl, AlkaneCarbonyl, and Mutagenicity, complementing the Benzene results in Table 3; the variant patterns are consistent with those on Benzene. This section provides the complete variant definitions and the detailed discussion summarized in Subsection 6.3 of the main text.

Variant definitions. V1–V3 replace the scoring objective: V1 uses only J_{sup}, V2 uses only J_{disp}, and V3 directly uses $\Phi(\mathcal{E}_S)$ without random-reference correction or any size penalty, in both training and final selection. V4 and V5 modify the pruning: V4 replaces the candidate-conditioned T-step

Table 5: Ablation study on FluorideCarbonyl, AlkaneCarbonyl, and Mutagenicity with GIN target models (percentages, mean ± sample standard deviation over five explainer seeds). V4 is not applicable to Mutagenicity because the default search on it already uses one step.

Dataset	Metric	RDS-Explainer	V1 (J_{sup})	V2 (J_{disp})	V3 (Φ)	V4 (1 step)	V5 (1 edge/step)
Fluoride Carbonyl	Charact ↑	79.21 ± 4.05	76.14 ± 4.35	92.46 ± 1.22	94.12 ± 1.47	77.98 ± 5.70	35.88 ± 13.99
	Sparsity ↓	23.56 ± 1.72	22.74 ± 1.50	34.92 ± 1.50	40.00 ± 2.81	33.74 ± 5.04	45.45 ± 5.54
	GEA ↑	55.29 ± 8.06	55.14 ± 5.62	55.81 ± 3.35	53.60 ± 1.47	56.19 ± 4.02	24.42 ± 14.03
Alkane Carbonyl	Charact ↑	100.00 ± 0.00	98.36 ± 2.25	98.63 ± 1.68	99.45 ± 0.75	99.45 ± 0.75	95.07 ± 3.15
	Sparsity ↓	34.07 ± 1.73	42.57 ± 27.82	33.06 ± 5.14	33.78 ± 3.31	33.57 ± 2.26	39.83 ± 5.29
	GEA ↑	4.25 ± 1.67	1.58 ± 1.94	5.04 ± 4.28	1.64 ± 1.42	3.17 ± 2.80	4.46 ± 4.75
Mutagenicity	Charact ↑	93.78 ± 2.26	92.74 ± 2.25	93.04 ± 2.85	94.67 ± 2.42	–	81.33 ± 13.64
	Sparsity ↓	26.08 ± 2.16	25.87 ± 1.45	26.30 ± 1.75	40.15 ± 33.15	–	31.36 ± 4.79
	GEA ↑	67.49 ± 2.51	67.67 ± 2.22	67.09 ± 2.71	57.59 ± 23.48	–	55.61 ± 10.13

sequence with T independent one-step training transitions, each starting from the full graph, and prunes only once at inference (keeping the same training-transition and update budget, unlike the plain $T{=}1$ setting in the sensitivity analysis, which also reduces training transitions); V5 updates a single focal edge per step via its two endpoints and re-encodes the current graph, retaining the shared actor, endpoint aggregation, graph-level state, J_{dual}, and sequential training.

Detailed results. V1 retains almost the same Sparsity as RDS-Explainer but consistently lower Charact and GEA, and V2 offers no consistent gain, being worse on all three metrics with an order-of-magnitude larger seed variance on Benzene. V3 attains the highest Charact (99.90) on Benzene, but its Sparsity rises from 41.78 to 90.80 and its GEA falls from 74.33 to 42.79: using Φ directly preserves a high response by keeping much of the graph, but fails to identify a compact explanation. The three variants thus show that J_{sup} and J_{disp} constrain different aspects of the explanation, and that only their combination balances prediction support, explanation size, and the evidence left outside.

V4 lowers Charact by 3.29 points on Benzene and raises Sparsity from 23.56 to 33.74 on FluorideCarbonyl, suggesting that multi-step feedback helps produce more compact explanations when quality changes little, although its benefit varies across datasets. V5 attains lower Charact, higher Sparsity, and larger seed variance on all four datasets (on Benzene, 78.33 versus 94.84 Charact), showing that parallel multi-edge updates outperform edge-at-a-time updates even though V5 retains the shared actor, graph-level context, and J_{dual}.

E.2 Sparsity Frontier

We then compare RDS-Explainer with baselines evaluated at fixed Sparsity values from 0.1 to 0.9. This complements the two baseline-size protocols in Subsection 6.1 by showing how baseline quality changes as the retained proportion increases. In Figure 5, each star summarizes RDS-Explainer's explanations, and the horizontal dashed line marks its corresponding quality level. On Benzene, RDS-Explainer attains higher Charact and GEA with comparatively compact explanations. On FluorideCarbonyl, it has higher Charact than baseline explanations of similar size, while its GEA is comparable to RCExplainer at Sparsity values of 20% and 30%. On Mutagenicity, larger baseline explanations can exceed its Charact, but their GEA decreases markedly as the retained proportion approaches the full graph. AlkaneCarbonyl remains a boundary case: Charact is high at many explanation sizes, whereas GEA remains low because the target GNN's prediction-relevant evidence does not fully align with the dataset-provided ground-truth chemical motifs. Overall, RDS-Explainer achieves competitive explanation quality with relatively compact explanations in these comparisons.

E.3 Just-Sufficient Explanations

To examine whether the selected explanations are just-sufficient, we perturb the selected and excluded edges in opposite directions. A just-sufficient explanatory edge set should have high quality before perturbation and contain prediction-relevant evidence, while the excluded edges should be comparatively dispensable: removing selected edges should reduce explanation quality, whereas

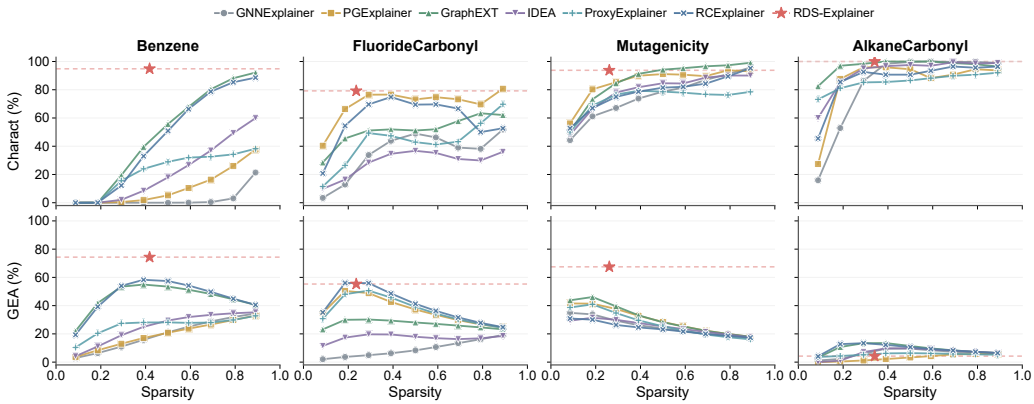


Figure 5: Explanation quality across Sparsity values. Baseline curves use fixed Sparsity values from 0.1 to 0.9, whereas each RDS-Explainer star summarizes its graph-specific explanations.

adding excluded edges should provide little benefit. We denote the explanation itself by $\widehat{\mathcal{E}}_S^0 = \widehat{\mathcal{E}}_S$, and the sets obtained after removing or adding k edges by $\widehat{\mathcal{E}}_S^{-k}$ and $\widehat{\mathcal{E}}_S^{+k}$, respectively. We first randomly remove one to three selected edges or add one to three edges from $\mathcal{E} \setminus \widehat{\mathcal{E}}_S$, repeating each setting 20 times. We then perform score-aware perturbations on RDS-Explainer: selected edges are removed in ascending score order, whereas excluded edges are added in descending score order. As a complementary diagnostic of the learned ranking, we also remove selected edges in descending score order.

The top row of Figure 6 reports the random perturbations. At zero perturbation, RDS-Explainer obtains Charact values of 94.8, 79.2, and 93.8 on Benzene, FluorideCarbonyl, and Mutagenicity, respectively. Randomly removing one selected edge reduces these values by 50.0, 26.1, and 40.5 points, the largest one-edge decreases among the compared methods on these datasets. Adding up to three non-selected edges provides no improvement. By contrast, RCExplainer gains about 19 points on Benzene after one edge is added, and several baselines on Mutagenicity also improve after additions, indicating that useful predictive evidence remains outside their selected sets. AlkaneCarbonyl is less discriminative because RDS-Explainer and several baselines are close to Charact saturation: one-edge removal lowers RDS-Explainer by only 9.4 points, and edge additions still provide essentially no benefit.

The bottom row reports the score-aware perturbations. Removing the lowest-scored selected edge reduces Charact by 6.9, 11.5, and 3.6 points on Benzene, FluorideCarbonyl, and Mutagenicity, while adding even the highest-scored excluded edges does not improve the zero-perturbation result. Removing the highest-scored selected edge causes much larger decreases of 58.5, 36.8, and 72.0 points, showing that the learned scores concentrate prediction-relevant evidence near the top of the ranking. AlkaneCarbonyl again shows a weaker local response under saturation: the corresponding decreases are 1.4 and 18.6 points for removing the lowest- and highest-scored selected edge.

Together, the random and score-aware perturbations show an asymmetric pattern that is clearest on Benzene, FluorideCarbonyl, and Mutagenicity: removing selected edges harms explanation quality, whereas adding excluded edges offers no benefit. This pattern is consistent with the selected explanations being just-sufficient, while the response strength varies across datasets and does not by itself establish strict non-additive interactions.

E.4 Parameter Sensitivity

We vary the number of random reference samples per size $L \in \{5, 10, 20\}$, the reward-shaping coefficient $\lambda_s \in \{0, 0.5, 1, 1.5\}$, and the number of pruning steps $T \in \{1, 2, 3, 5, 10\}$ one at a time while keeping the remaining parameters at their dataset-specific defaults. As shown in Figure 7,

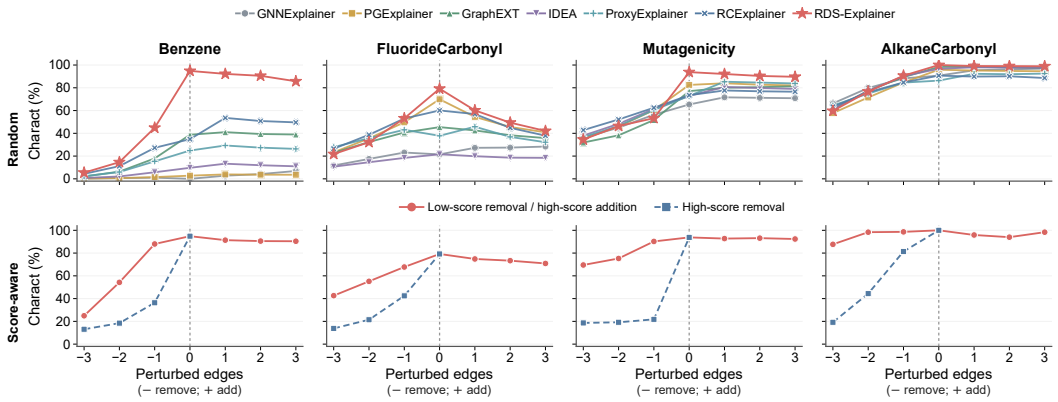


Figure 6: Charact under random and score-aware edge perturbations on four molecular datasets with GIN target models.

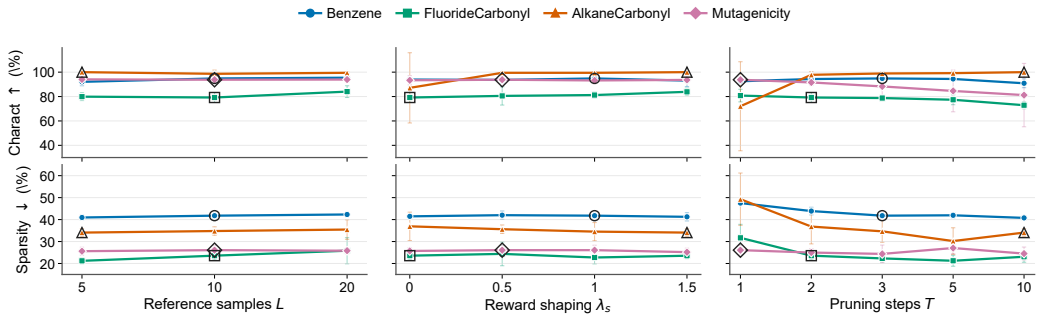


Figure 7: Sensitivity of Charact and Sparsity to L, λ_s, and T. Means and sample standard deviations over five explainer seeds are shown, with outlined markers denoting dataset-specific defaults. Tested values are equally spaced on each horizontal axis.

L has the smallest effect, with Charact varying by at most 4.82 points within each dataset. The effect of λ_s is more dataset dependent: it changes Benzene and Mutagenicity only slightly, while inappropriate values can increase variation on AlkaneCarbonyl. The influence of T is stronger, affecting both Charact and Sparsity, particularly on FluorideCarbonyl and AlkaneCarbonyl. Overall, the selected defaults lie in comparatively stable regions, while λ_s and especially T reflect a dataset-dependent quality–size trade-off.

E.5 Evaluation across GNN Architectures

To assess whether RDS-Explainer performs consistently across target GNN architectures, we evaluate it on Benzene using GIN-3Layer, GCN-3Layer, and GAT-2Layer. We use the second baseline-size protocol from Subsection 6.1: for each target GNN, every baseline is assigned the number of edges selected by RDS-Explainer on the same graph and seed, while RDS-Explainer receives no size information (Table 6). For Charact, RDS-Explainer leads on GIN and GCN and is only 0.06 points below RCExplainer on GAT. Its GEA is highest on GIN and GAT and 1.78 points below PG-Explainer on GCN. These results show that RDS-Explainer performs consistently across the three evaluated architectures on Benzene.

Table 6: Explanation quality across target GNN architectures on Benzene. Results are percentages (mean ± sample standard deviation over five seeds); bold and underlined values indicate the best and second-best result in each column.

	GIN-3Layer		GCN-3Layer		GAT-2Layer	
Method	Charact	GEA	Charact	GEA	Charact	GEA
GNNExplainer	0.79 ± 0.14	16.12 ± 0.54	15.19 ± 1.40	25.47 ± 0.56	40.33 ± 3.33	36.56 ± 0.82
PGExplainer	5.39 ± 0.99	18.95 ± 1.04	83.61 ± 3.78	**71.56 ± 1.08**	87.25 ± 1.77	63.57 ± 0.72
GraphEXT	46.47 ± 1.64	57.63 ± 0.43	27.67 ± 0.80	39.64 ± 0.89	50.64 ± 1.17	38.11 ± 0.76
IDEA	11.09 ± 14.52	27.40 ± 18.30	8.30 ± 6.16	12.15 ± 5.32	48.18 ± 3.51	47.16 ± 1.56
ProxyExplainer	34.38 ± 46.80	31.78 ± 37.59	37.14 ± 26.50	37.02 ± 19.74	78.28 ± 7.77	57.41 ± 5.34
RCExplainer	50.28 ± 5.08	62.56 ± 1.44	45.72 ± 5.01	35.95 ± 13.45	**89.21 ± 2.08**	61.35 ± 0.52
RDS-Explainer	**94.84 ± 1.46**	**74.33 ± 0.52**	**90.73 ± 2.40**	69.78 ± 1.96	89.15 ± 1.01	**64.89 ± 1.17**

E.6 Runtime Analysis

Table 7 reports the average runtime across all evaluated datasets. RDS-Explainer is faster at test time than GNNExplainer, GraphEXT, RCExplainer, and EiG-Search, although PGExplainer, IDEA, and ProxyExplainer remain faster. Its total run time of 46.7 minutes is close to that of RCExplainer and substantially lower than that of GraphEXT, but higher than those of the other methods. Overall, RDS-Explainer trades greater offline computation for efficient explanation of new graphs.

F Qualitative Case Study

Figure 8 presents representative GIN explanations from the four datasets under the size-controlled setting of Subsection 6.1. RDS-Explainer recovers the ground-truth structures in the selected Benzene, FluorideCarbonyl, and Mutagenicity examples, whereas several baselines place some of their equally many edges in different molecular regions. AlkaneCarbonyl illustrates the metric discrepancy discussed in Subsection 6.1: the ground-truth motif has zero Charact for this prediction, whereas RDS-Explainer attains Charact 100 but zero GEA, indicating that the target GNN relies on predictive structure that differs from the dataset-provided motif.

G Native-Adaptive Baseline Comparison

Table 8 reports the values summarized in Figure 3. EiG-Search uses its author-defined full-budget prefix search, with no external target size; SubgraphX and RDS-Explainer likewise select explanations under their native procedures. All methods use the same correctly predicted test graphs, frozen GIN target models, and project evaluator. Sparsity is the retained-edge fraction (lower is more compact), matching the manuscript definition; the figure plots its complement $1 - \text{Sparsity}$. These native outputs have different actual sizes, so Charact and GEA differences should not be interpreted as size-matched effects.

Table 7: Average runtime across all evaluated datasets.

Method	Test time / graph (s) ↓	Total run time (min) ↓
GNNExplainer	0.608	7.0
PGExplainer	**0.005**	13.2
GraphEXT	9.968	86.1
IDEA	0.008	**2.3**
ProxyExplainer	0.012	4.7
RCExplainer	0.489	44.5
EiG-Search	0.295	2.7
RDS-Explainer	0.073	46.7

Table 8: Native-adaptive explanation quality on four molecular datasets. Sparsity is the retained-edge fraction; Charact and GEA are reported using the project evaluator. All values are percentages.

Dataset	Method	Sparsity ↓	Charact ↑	GEA ↑
Benzene	EiG-Search	64.06	100.00	62.52
	SubgraphX	94.73	99.81	38.43
	RDS-Explainer	40.33	92.30	73.73
FluorideCarbonyl	EiG-Search	70.75	100.00	30.84
	SubgraphX	83.14	41.67	25.04
	RDS-Explainer	21.60	82.02	62.44
AlkaneCarbonyl	EiG-Search	45.23	100.00	13.87
	SubgraphX	42.94	98.63	7.30
	RDS-Explainer	34.61	100.00	2.10
Mutagenicity	EiG-Search	66.05	100.00	24.20
	SubgraphX	26.02	40.74	32.30
	RDS-Explainer	26.16	94.81	69.02

Table 9: Variation in the explanation sizes selected by RDS-Explainer across test graphs and five explainer seeds.

Dataset	Mean (IQR)	Distinct	Graph std	Seed std	Graph/Seed
Benzene	9.72 (5.00)	20	2.72	0.45	6.05
FluorideCarbonyl	5.18 (2.00)	13	1.27	1.20	1.06
AlkaneCarbonyl	7.90 (3.00)	12	1.97	0.71	2.77
Mutagenicity	5.80 (4.00)	19	2.97	0.77	3.83

H Adaptive Size Variability

We summarize how the explanation sizes selected by RDS-Explainer vary across test graphs and explainer seeds. The mean and interquartile range (IQR) are computed over all graph–seed pairs. Distinct counts the number of different sizes observed over the same pairs. Graph std is the sample standard deviation across graphs after averaging each graph's selected size over the five seeds, whereas seed std is the mean across graphs of the within-graph sample standard deviation over seeds. The Graph/Seed ratio compares graph-to-graph variation with variation across seeds.

The selected sizes vary across graphs, although the extent of variation differs by dataset. Benzene shows the clearest graph-to-graph variation, which is more than six times the variation across seeds. Variation across graphs also exceeds variation across seeds on AlkaneCarbonyl and Mutagenicity. FluorideCarbonyl shows 13 distinct selected sizes, but its graph std and seed std are similar, so evidence for consistent graph-to-graph differences is weaker. Overall, RDS-Explainer selects different sizes across graphs, with dataset-dependent strength and stability.

I Proof of Joint Positivity and Full-Graph Suppression

Proof. Let

$$a = [J_{\mathrm{sup}}(\mathcal{E}_S)]_+, \qquad b = [J_{\mathrm{disp}}(\mathcal{E}_S)]_+.$$

For $k > 1$, Eq. 6 gives

$$J_{\mathrm{dual}}(\mathcal{E}_S) = \frac{2ab}{a + b + \epsilon},$$

where $\epsilon > 0$. The denominator is strictly positive.

If either $J_{\mathrm{sup}}(\mathcal{E}_S) \leq 0$ or $J_{\mathrm{disp}}(\mathcal{E}_S) \leq 0$, then $a = 0$ or $b = 0$, respectively. The numerator is therefore zero, which gives $J_{\mathrm{dual}}(\mathcal{E}_S) = 0$. Conversely, if both $J_{\mathrm{sup}}(\mathcal{E}_S) > 0$ and $J_{\mathrm{disp}}(\mathcal{E}_S) > 0$, then $a > 0$ and $b > 0$, so the numerator is strictly positive and $J_{\mathrm{dual}}(\mathcal{E}_S) > 0$. This proves part (i).

For the full graph, $\mathcal{E}_S = \mathcal{E}$ and $k = M$. By the definition of the random-reduction reference,

$$\Phi^{-}_{\mathrm{ref}}(M) = \Phi(\mathcal{E}) - \big(M - M\big)\bar{\delta}^{-} = \Phi(\mathcal{E}).$$

Therefore,

$$J_{\mathrm{disp}}(\mathcal{E}) = \Phi(\mathcal{E}) - \Phi^{-}_{\mathrm{ref}}(M) = 0.$$

Its positive part is zero, so Eq. 6 gives $J_{\mathrm{dual}}(\mathcal{E}) = 0$. This proves part (ii). □

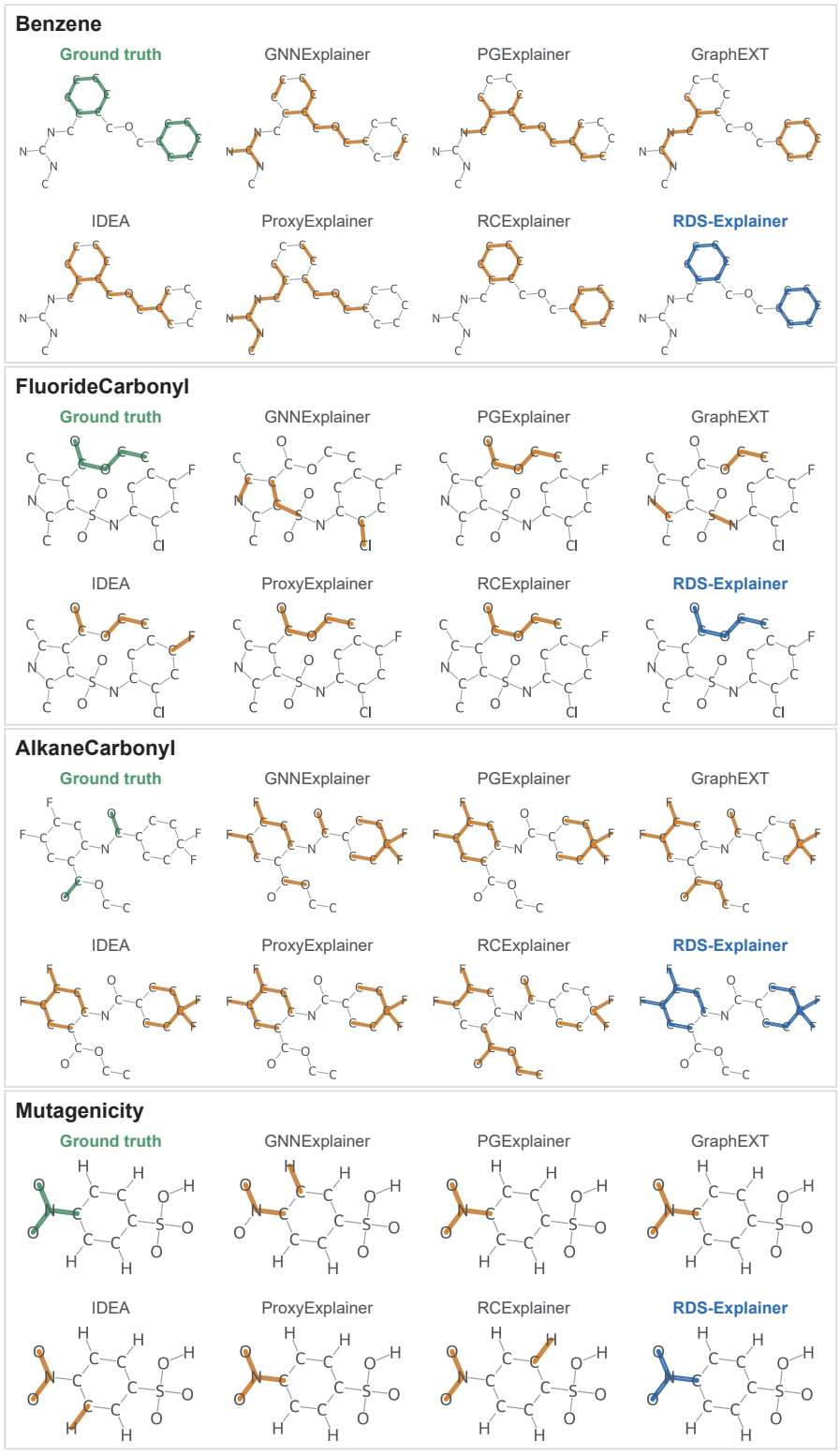


Figure 8: Representative explanations on four molecular datasets. All explainers retain the same number of edges as RDS-Explainer for each displayed molecule; the ground-truth explanations are shown as references.