STAR: A TRANSFERABLE EXPLANATORY APPROACH FOR GRAPH NEURAL NETWORKS

Anonymous authors

ABSTRACT

Graph neural networks (GNNs) integrate node attributes and relational structure through neighborhood aggregation to capture graph dependencies, but nonlinear computations obscure the decision evidence underlying individual predictions. To improve explainability, post-hoc explanatory methods identify decision evidence without modifying the pretrained GNN's architecture or parameters. Among these approaches, parameterized explanatory methods exploit cross-instance explanatory information to improve explanations for a given pretrained GNN. Extending these explanatory methods to other pretrained GNNs requires accounting for differences in predictive behavior. Although previously acquired explanatory information can guide explanation generation, directly applying the fitted explanation function risks misidentifying the new pretrained GNN's decision evidence. Refitting can require additional samples that are unavailable at explanation time. To address this challenge, this paper proposes STAR, a transferable explanatory approach for GNNs based on Source-informed explanatory prior generation and Target-Adaptive explanation Refinement. To exploit explanatory information across pretrained GNNs, STAR develops a response-conditioned explanatory prior generator that models explanatory relevance in relation to the queried pretrained GNN's behavior and input information. Prediction-aligned optimization across source pretrained GNNs enables explanatory guidance for new pretrained GNNs without generator refitting. To preserve explanation fidelity despite potential source-to-target mismatches, STAR develops target-adaptive explanation refinement that evaluates evidence guided by the explanatory prior alongside locally identified alternatives under the target pretrained GNN. This assessment allows useful explanatory prior guidance to inform the final explanation while permitting correction through target-specific evidence. Experiments across six datasets and three GNN architectures demonstrate competitive explanatory fidelity and effective transfer to target pretrained GNNs excluded from generator fitting.

1 INTRODUCTION

Graph neural networks (GNNs) integrate node attributes and relational structure, supporting diverse applications such as recommendation, social network analysis, and molecular property prediction (Gilmer et al., 2017; Li et al., 2025; Zhao et al., 2024). These predictive capabilities rely on repeated neighborhood aggregation and nonlinear transformations, which entangle node and edge contributions and make the decision-making process of GNNs difficult to explain. The resulting lack of explainability complicates deployment in high-stakes domains such as healthcare, where experts need to examine the evidence behind predictions (Huang et al., 2024a; Chen et al., 2024; Zhao et al., 2025). To address this problem, recent research investigates post-hoc explanatory methods that identify decision evidence for a pretrained GNN's predictions, often as explanatory subgraphs, without modifying the pretrained GNN's architecture or parameters (Ma et al., 2026; Huang et al., 2024b). These explanations support model debugging and expert assessment, contributing to the trustworthy application of GNNs (Azzolin et al., 2025). Among these explanatory methods, parameterized explanatory methods fit an explanation function using multiple inputs and the pretrained GNN's predictive behavior (Qiu et al., 2024). By capturing recurring relationships between input substructures and predictions, parameterized explanatory methods can improve explanation accuracy and support explanation generalization to previously unseen instances of the same pretrained GNN (Luo et al., 2024; Wang et al., 2021).

However, parameterized explanatory methods often have limited transferability to new pretrained GNNs. Differences in training data and optimization can cause pretrained GNNs to rely on different

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.

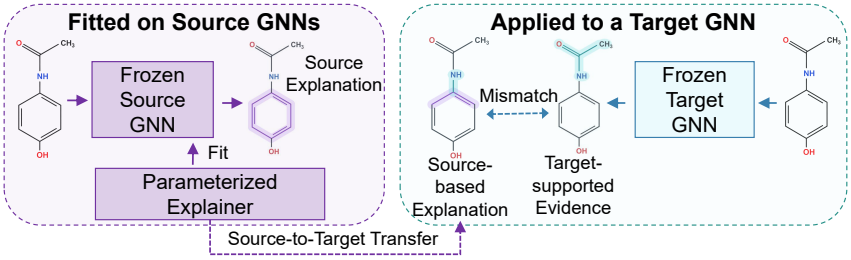

Figure 1: Schematic illustration of mismatch between a source-based explanation and decision evidence supporting the target pretrained GNN's prediction. For acetaminophen, the explanation emphasizes aromatic-ring bonds but omits part of the amide-containing fragment.

decision evidence, even when their architectures are identical. An explanation function fitted to a source pretrained GNN may consequently require refitting to explain a target pretrained GNN. Such refitting requires additional samples and corresponding target-GNN predictions, but the necessary samples may be unavailable when only the target pretrained GNN and the instance being explained are accessible. Although explanatory information acquired from the source pretrained GNN can guide the identification of candidate decision evidence for the target prediction, directly applying the source-fitted explanation function risks selecting evidence that does not adequately support that prediction. For example, Figure 1 schematically illustrates a source-fitted explanatory method emphasizing aromatic-ring bonds in acetaminophen while overlooking part of the amide-containing fragment supporting the target pretrained GNN's prediction. This mismatch can reduce explanation fidelity and undermine the reliability of explanations for model assessment and debugging. Instance-specific explanatory methods provide an alternative by directly optimizing an explanation using the instance to be explained and the target pretrained GNN's predictive responses (Ying et al., 2019; Lu et al., 2024). However, identifying decision evidence requires assessing how input components jointly support the prediction (Muschalik et al., 2025; Hu et al., 2025). The numerous possible combinations make exhaustive evaluation impractical, and approximate optimization can overlook evidence with stronger predictive support (Luo et al., 2020). The central challenge is therefore to exploit source-acquired explanatory information to identify decision evidence supporting the target pretrained GNN's prediction, without refitting the explanation function or introducing source-to-target mismatches.

To address this challenge, this paper proposes STAR, a transferable explanatory approach for GNNs based on Source-informed explanatory prior generation and Target-Adaptive explanation Refinement. To exploit source-acquired explanatory information while accounting for differences among pretrained GNNs, STAR constructs a response-conditioned explanatory prior generator that models explanatory relevance through the relationship between input structure and the queried pretrained GNN's representations and predictive responses. Prediction-aligned optimization across source pretrained GNNs fits this conditional relationship, allowing the frozen generator to produce target-dependent explanatory priors without refitting. To address potential source-to-target mismatches, STAR develops target-adaptive explanation refinement that assesses explanatory-prior-guided evidence under the target pretrained GNN and permits locally identified alternatives, adapting evidence selection to the queried prediction. By improving explanation transferability to new pretrained GNNs, this design broadens the applicability of post-hoc explanatory methods and supports the trustworthy use of GNNs across application settings. The main contributions are as follows.

- This paper proposes source-informed explanatory prior generation through a response-conditioned generator, enabling explanatory information acquired across source pretrained GNNs to guide target decision-evidence identification without target-specific refitting.
- This paper develops target-adaptive explanation refinement to address source-to-target mismatches through target-specific assessment of explanatory-prior-guided and locally identified evidence.
- This paper demonstrates STAR's explanatory fidelity and transferability through comparative and held-out evaluations across six datasets and three GNN architectures, complemented by component ablations.

2 Related Work

To improve the explainability of pretrained GNNs, existing research investigates post-hoc GNN explanatory methods that identify decision evidence without modifying the pretrained GNN's architecture or parameters, supporting model inspection and debugging (Longa et al., 2025). GNNExplainer establishes a general framework that optimizes structural and feature masks to identify compact evidence supporting individual predictions (Ying et al., 2019). PGM-Explainer further characterizes dependencies between input components and predictive responses through a probabilistic graphical model constructed from input perturbations (Vu & Thai, 2020). To account for the joint contributions of graph components, SubgraphX assesses candidate subgraphs through Shapley values (Yuan et al., 2021), while MAGE incorporates graph connectivity into the attribution of node interactions (Bui et al., 2024). Beyond evidence attribution, subsequent research improves the criteria governing explanation generation. GEAR addresses conflicts among explanation objectives through gradient adjustment (Zhang et al., 2024), and IDEA incorporates structural information by aligning an input and its explanatory subgraph in a prototypical representation space (Yin et al., 2026). Curvature-based explanation optimization additionally exploits edge curvature and effective resistance to improve robustness under graph perturbations (Liu et al., 2025). To extend explanation generation beyond independent instance-specific optimization, parameterized explanatory methods exploit explanatory information acquired across multiple instances. PGExplainer fits a parameterized edge-selection function to a pretrained GNN's predictive behavior, supporting explanation generalization to unseen instances of that pretrained GNN (Luo et al., 2020). ReFine further uses class-wise explanatory patterns to guide individual explanations, adapting its explanatory modules through instance-specific fine-tuning (Wang et al., 2021). TAGE extends explanation generation to unseen downstream tasks by explaining a fixed GNN embedding model, addressing changes in downstream tasks while retaining the underlying embedding model (Xie et al., 2022).

Existing explanatory methods improve decision-evidence identification, but transferring acquired explanatory information to separately pretrained GNNs requires accounting for differences in predictive behavior. STAR addresses this requirement through source-informed explanatory prior generation and target-adaptive refinement, extending explanation generation to new pretrained GNNs without refitting the generator.

3 Method

STAR exploits source-acquired explanatory information through source-informed explanatory prior generation and target-adaptive explanation refinement, as illustrated in Figure 2. Given a pretrained GNN f_θ and input graph $G = (V, E, X)$, STAR identifies decision evidence for the queried prediction as an explanatory subgraph $G[S]$. Here, θ denotes the GNN parameters, V, E, and X denote the node set, edge set, and node attributes, and $S \subseteq E$ denotes the retained edges. The subgraph preserves the original nodes and attributes, with retained-edge fraction $\rho \in (0, 1)$. Let $p_0 = f_\theta(G)$ denote the predictive distribution and $c_0 = \arg\max_c p_{0,c}$ the predicted class. To relate source-acquired information to the queried prediction, the explanatory prior generator g_ϕ estimates edge relevance from the input and the pretrained GNN's representations and predictive responses. Prediction-aligned optimization fits its parameters ϕ across source pretrained GNNs. During target explanation, the generated explanatory prior guides candidate construction. Target-adaptive refinement addresses potential mismatches by assessing these candidates alongside locally optimized alternatives. STAR accesses the pretrained GNN's intermediate representations and gradients with respect to message masks. Both the generator and pretrained GNN remain frozen while target predictive support determines the final explanatory subgraph. Details are described in the following subsections.

3.1 Source-Informed Explanatory Prior Generation

To exploit source-acquired explanatory information for the queried prediction, STAR constructs a response-conditioned explanatory prior generator. **Input and response descriptors** summarize the input and the pretrained GNN's predictive behavior. The generator then **conditions neighborhood aggregation on the GNN response** and **estimates edge relevance**. The resulting continuous edge scores form the explanatory prior that guides candidate subgraph construction.

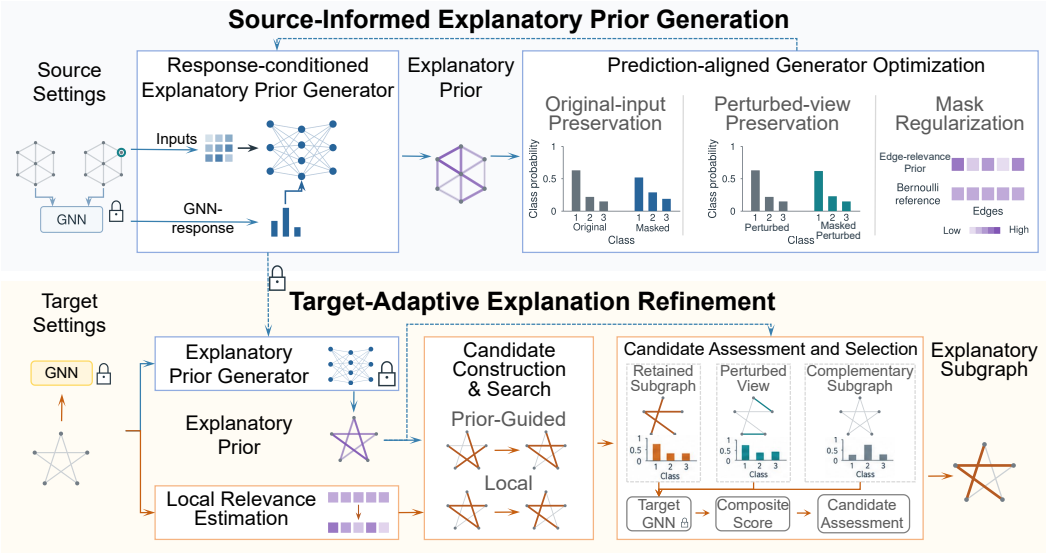

Figure 2: Framework of STAR. Purple denotes edge relevance; orange denotes selected edges.

Construct input and response descriptors. Pretrained GNNs can differ in their input-feature, hidden-representation, and output dimensions. To accommodate these differences, STAR retains the input graph's connectivity and constructs descriptors from structural statistics, pretrained GNN representations, and predictive responses. The input descriptor for node v is defined as

$$a_v = \left[s_v(G) \parallel r_v(e_v, e_*, \bar{e}, \bar{e}_{\mathcal{N}(v)}) \right] \in \mathbb{R}^{d_a}, \tag{1}$$

where $\parallel$ denotes concatenation and d_a denotes the input-descriptor dimension. The vector $s_v(G)$ contains normalized incoming, outgoing, and total degrees, graph size, and query-relative structural information. The vector e_v denotes the pretrained GNN's final-layer representation of node v. The references $\bar{e}$ and $\bar{e}_{\mathcal{N}(v)}$ denote mean representations over the graph and the neighbor set $\mathcal{N}(v)$, respectively. The reference e_* is the queried node's representation for a node prediction and the graph mean for a graph prediction.

The vector r_v contains the node-representation norm, distances and cosine similarities to these three references, and local representation dispersion. Node attributes contribute through the pretrained GNN's representations. Query-relative structural information records query membership, distance, and reachability for node predictions, with zero entries for graph predictions. An isolated node uses its own representation as the neighborhood mean.

The response descriptor summarizes predictive behavior on original and perturbed inputs:

$$b = \left[s_{\text{pred}}(p_0, o_0) \parallel s_{\text{pert}}\left(p_0, \{p_j\}_{j=1}^{K}\right) \parallel s_{\text{graph}}(\{e_v\}_{v \in V}, e_*) \right] \in \mathbb{R}^{d_b}, \tag{2}$$

where d_b denotes the response-descriptor dimension, o_0 denotes the logits corresponding to p_0, and $p_j = f_\theta(G_j)$ denotes the pretrained GNN's predictive distribution on the j-th of K locally perturbed inputs. The vector s_{pred} contains ranked class probabilities and standardized logits, together with confidence, predictive margin, entropy, probability concentration, and class-count summaries. The vector s_{pert} aggregates prediction divergence, confidence in the original predicted class, predictive margin, and entropy across perturbations. The vector s_{graph} supplies graph-level context through relative representation norms, dispersion, and similarities between connected nodes.

Ranked classwise values are truncated or zero-padded to a prescribed length, while scalar summaries accommodate different representation dimensions. These computed statistics provide descriptors of consistent dimensions across pretrained GNNs. Appendix B.1 specifies descriptor coordinates and normalizations; Appendix B.2 describes perturbation probes.

Condition neighborhood aggregation on the GNN response. To process local structure according to the prediction being explained, STAR combines Graph Isomorphism Network (GIN) aggregation (Xu et al., 2019) with response-conditioned feature-wise affine modulation (Perez et al., 2018).

Projection weights W_a and bias c_a initialize the generator states as $h_v^{(0)} = W_a a_v + c_a$. At layer l, the generator computes

$$u_v^{(l)} = F_l\left(h_v^{(l-1)} + \sum_{u\in\mathcal{N}(v)} h_u^{(l-1)}\right), \tag{3}$$

$$\widetilde{u}_v^{(l)} = (1 + \tanh\gamma^{(l)}(b)) \odot u_v^{(l)} + \zeta^{(l)}(b), \tag{4}$$

$$h_v^{(l)} = \mathrm{LN}\Big(h_v^{(l-1)} + \mathrm{Drop}(\mathrm{ReLU}(\widetilde{u}_v^{(l)}))\Big), \tag{5}$$

where F_l denotes a multilayer perceptron and $u_v^{(l)}$ denotes the aggregated node state. Functions $\gamma^{(l)}$ and $\zeta^{(l)}$ map b to feature-wise scaling and shifting coefficients, producing the response-conditioned state $\widetilde{u}_v^{(l)}$. The update uses elementwise multiplication $\odot$, rectified linear activation ReLU, dropout Drop, and layer normalization LN.

The resulting states $h_v^{(l)}$ combine neighborhood information with the queried GNN's response. They belong to the explanatory prior generator and are distinct from the pretrained GNN's representations e_v.

Estimate edge relevance. To score edges for explanatory subgraph construction, the decoder combines endpoint states, their differences, and elementwise products. Encodings of input and response descriptors retain information available before aggregation:

$$z_{uv} = [h_u^{(L)}, h_v^{(L)}, |h_u^{(L)} - h_v^{(L)}|, h_u^{(L)} \odot h_v^{(L)}, \psi(a_u, a_v), \eta(b)], \tag{6}$$

$$m_{uv}^G = \mathrm{sigmoid}(d_{\phi,1}(z_{uv}) - d_{\phi,2}(z_{uv})), \tag{7}$$

where L denotes the number of generator layers, z_{uv} denotes the edge descriptor, and brackets denote concatenation. Mappings ψ and η encode the endpoint input descriptors and the response descriptor, respectively. The decoder outputs two logits, $d_{\phi,1}$ and $d_{\phi,2}$, whose difference determines the edge-relevance score through the logistic sigmoid. Source fitting uses the raw generated mask. During target refinement, opposite-direction probabilities are averaged for undirected relations, whose directions are retained together during subgraph extraction.

The mask $m^G \in (0,1)^{|E|}$ forms the explanatory prior for candidate construction and search. The generator is fitted across source pretrained GNNs through prediction alignment (Section 3.2) and remains frozen during target explanation.

3.2 Prediction-Aligned Generator Optimization

To make the explanatory prior informative for identifying decision evidence, STAR optimizes the generator using predictive feedback from source pretrained GNNs. Generated edge scores act as message masks, and the objective encourages preservation of the pretrained GNN's predictions under these masks. Locally perturbed inputs further assess prediction preservation, with greater penalties for deterioration when the unmasked predictions remain similar.

Assess prediction preservation. To relate generated edge scores to the queried prediction, STAR compares predictions with and without the generated mask. Let $f_\theta(G; m)$ denote the prediction obtained by scaling edge messages with m. For a locally perturbed input G_ξ, let $p_\xi = f_\theta(G_\xi)$ denote its unmasked prediction and m_ξ^G the recomputed explanatory prior. Perturbations modify edge endpoints or interpolate attributes between structurally similar nodes (Appendix B.2). Class-normalized Kullback–Leibler (KL) divergence (Cover & Thomas, 2006) is $D_C(p\|q) = C^{-1}\sum_c p_c \log(p_c/q_c)$ for C classes. The alignment losses are

$$\ell_0 = aD_C(p_0 \| f_\theta(G; m^G)), \qquad \ell_\xi = aD_C(p_\xi \| f_\theta(G_\xi; m_\xi^G)), \tag{8}$$

where $a > 0$ denotes the alignment scale. Minimizing these losses encourages the generated masks to preserve the pretrained GNN's predictive distributions on their respective inputs.

Weight alignment by prediction similarity. Local input changes can affect both the prediction and its supporting evidence. STAR therefore penalizes increased alignment loss more strongly when the unmasked predictions remain similar, using

$$\alpha_\xi = 1 - \exp\{-\lambda e^{-D_C(p_\xi\|p_0)/\tau_g}\}, \qquad \mathcal{R}_\xi = \ell_0 + \alpha_\xi[\ell_\xi - \ell_0]_+, \tag{9}$$

where $\tau_g > 0$ denotes prediction-change tolerance, $\lambda \geq 0$ controls penalty strength, and $[x]_+ = \max(x, 0)$. The difference $\ell_\xi - \ell_0$ measures deterioration in prediction preservation after perturbation. Similar unmasked predictions increase α_ξ and strengthen the penalty when this difference is positive. Generated masks may vary with the input. Reference predictions and α_ξ remain constant during differentiation. Appendix B.5 derives the loss properties.

Regularize and optimize the generator. To discourage extreme generated probabilities, STAR adds Bernoulli-reference mask regularization (Miao et al., 2022), giving

$$\mathcal{L}_{\text{shared}}(\phi) = \mathbb{E}_{(G,f_\theta),\xi,r}\left[\mathcal{R}_\xi + \beta\mathcal{B}(m^G, r)\right], \tag{10}$$

where $\beta \geq 0$ denotes the regularization weight and $\mathcal{B}(m, r) = |E|^{-1}\sum_e D_{\text{KL}}(\text{Ber}(m_e)\|\text{Ber}(r))$. Here, Ber denotes a Bernoulli distribution, D_{KL} denotes ordinary KL divergence, and $r \in (0, 1)$ is an independently sampled reference rate. Sampling r symmetrically about 0.5 produces the stated regularization effect in expectation (Appendix B.6). The retained-edge fraction controls explanation size separately. The expectation covers source pretrained GNN–input pairs, perturbations, and reference rates. Gradients through masked predictions update only ϕ. Original-input alignment and regularization select the generator on source validation pairs (Appendix B.3). The selected generator remains frozen during target-adaptive refinement.

3.3 Target-Adaptive Explanation Refinement

The explanatory prior supplies source-informed guidance, but its edge ranking may not identify the evidence supporting a particular target prediction. STAR addresses this mismatch through instance-specific refinement (Wang et al., 2021). Local relevance estimation supplies alternative scores, candidate construction and search refine explanatory subgraphs, and candidate assessment selects the final explanation under the target pretrained GNN. Both the explanatory prior generator and target pretrained GNN remain frozen.

Local relevance estimation. To identify evidence beyond the explanatory prior's ranking, STAR optimizes an instance-specific mask $m = m(z) = \text{Sym}_G(\mathcal{C}(\text{sigmoid}(z)))$ from uniform logits $z^{(0)} = 0$. Here, $\mathcal{C}$ clips probabilities to $[\varepsilon_{\text{num}}, 1 - \varepsilon_{\text{num}}]$, and Sym_G averages opposite directions for undirected relations and is the identity otherwise. The processed explanatory prior is $\widetilde{m}^G = \text{Sym}_G(\mathcal{C}(m^G))$ (Appendix B.4). The objective favors prediction preservation and reduced target-class confidence under the complementary mask (Amara et al., 2022). Let $q^+ = f_\theta(G; m)$, $q^- = f_\theta(G; 1 - m)$, and $\chi_0(q) = q_{c_0}/\pi_0$, where $\pi_0 = p_{0,c_0} > 0$. The local objective is

$$\mathcal{J}_t(z) = D_C(p_0\|q^+) + \mu\chi_0(q^-) + \nu g^{\text{loc}}_{\xi_t} D_C(p_{\xi_t}\|f_\theta(G_{\xi_t}; A_{\xi_t} m)) + \omega\overline{H}(m), \tag{11}$$

where t indexes updates, and $\mu, \nu, \omega \geq 0$ weight complementary confidence, perturbed-input prediction preservation, and mean binary entropy $\overline{H}(m)$, respectively. Entropy minimization sharpens edge scores. The weight $g^{\text{loc}}_\xi = \text{clip}(e^{-D_C(p_0\|p_\xi)/\tau_{\text{loc}}}, g_{\min}, 1)$ emphasizes perturbed inputs with similar predictions, where $\tau_{\text{loc}} > 0$ controls tolerance and $g_{\min} \in [0, 1]$ is the minimum weight. Mapping A_ξ retains scores for matching edges and assigns the fixed mean generated-mask value to new edges (Appendix B.4). After T Adam updates (Kingma & Ba, 2015), $z^{(T)}$ yields local scores.

Candidate construction and search. To assess the contribution of source-informed guidance, STAR constructs two candidate pools. The local pool contains optimized and uniform masks. The explanatory-prior-guided pool contains the generated mask, optimized local mask, and their logit interpolations,

$$m_\upsilon = m((1 - \upsilon)z^G + \upsilon z^{(T)}), \qquad \upsilon \in \mathcal{V}, \tag{12}$$

where $z^G = \text{logit}(\mathcal{C}(\widetilde{m}^G))$, logit denotes inverse sigmoid, and $\mathcal{V} \subseteq [0, 1]$ contains endpoint and intermediate coefficients. Each pool selects an initialization by assessing prediction preservation under continuous and thresholded masks, complementary confidence, and entropy (Appendix B.4).

Continuous message weighting and discrete edge removal can affect predictions differently. Each selected initialization therefore undergoes explicit subgraph search using a fixed set of perturbed inputs $\mathcal{Q}$, minimizing

$$\begin{aligned}\mathcal{S}_\mathcal{Q}(S) = {} & D_C(p_0\|f_\theta(G[S])) + \mu\chi_0(f_\theta(G[E \setminus S])) \\ & + \frac{\nu_s}{\max\{1, |\mathcal{Q}|\}}\sum_{\xi\in\mathcal{Q}} g^{\text{loc}}_\xi D_C(p_\xi\|f_\theta(G_\xi[S_\xi])),\end{aligned} \tag{13}$$

where $\nu_s \geq 0$ weights perturbed-input prediction preservation and S_ξ contains retained edges with matching endpoints in G_ξ. The terms assess prediction preservation after retaining the candidate, target-class confidence after removing it, and prediction preservation under perturbation. The set $\mathcal{Q}$ contains inputs with similar predictions and original-edge matches; its loss contribution is zero when empty.

Search exchanges selected and unselected edges while preserving the retained-edge count. Initialization scores guide proposals, with generated scores additionally guiding explanatory-prior-guided search. At each retained fraction, search evaluates at most W exchanges per iteration for B iterations, accepting only score reductions. Fractions are processed in ascending order, preserving edges finalized at smaller fractions and producing a refined edge ordering (Appendix B.4).

Candidate assessment and selection. To determine whether source-informed guidance improves the explanation under the target pretrained GNN, STAR uses the refined local ordering as the initial reference. The generated and refined explanatory-prior-guided orderings are compared with this reference at common retained fractions using additional fixed perturbed inputs. At each fraction, STAR computes the candidate's composite score minus the reference score. Replacement requires a lower mean score after penalizing variation in these differences, while limiting deterioration at individual fractions. An accepted candidate becomes the reference for the next comparison. Appendix B.4 specifies the comparison rule, and Appendix B.7 derives its score-control properties. The selected ordering yields the final explanatory subgraph by retaining approximately ρM of the M undirected relations or directed edges.

4 Experiments

4.1 Experimental Setup

Datasets and target GNNs. Experiments use MUTAG, PROTEINS (Morris et al., 2020), BBBP, BACE, HIV (Wu et al., 2018), and Benzene (Agarwal et al., 2023), with graph convolutional networks (GCN), GIN, and graph attention networks (GAT) (Kipf & Welling, 2017; Xu et al., 2019; Veličković et al., 2018). Three seeds use label-stratified random 70/10/20% train/validation/test splits. Explanatory methods share frozen checkpoints and up to 100 correctly predicted test inputs per configuration.

Baselines and transfer protocol. GNNExplainer (Ying et al., 2019), PGExplainer (Luo et al., 2020), GraphMask (Schlichtkrull et al., 2021), GEAR (Zhang et al., 2024), IDEA (Yin et al., 2026), and Graph Curvatures with PGExplainer (Liu et al., 2025) represent instance-specific, parameterized, multiobjective, representation-aligned, and robustness-oriented explanations. *Joint-fit* fits one STAR generator per seed across all datasets and their pretrained GNNs. *Held-out* excludes the evaluated dataset and its pretrained GNNs from generator fitting and checkpoint validation, retaining the architecture families. Both use frozen generators and identical refinement, with inference hyperparameters fixed on pooled benchmark-validation cells including evaluated datasets. Parameterized baselines fit on their target GNN's training split.

Evaluation targets and metrics. Fidelity measures target-class confidence on explanatory subgraphs, and Robust Fidelity measures confidence after perturbing and re-explaining inputs (Amara et al., 2022; Agarwal et al., 2023; Liu et al., 2025). Normalized area under the curve (AUC) over retained fractions 0.1–0.9 summarizes predictive support across explanation sizes. Higher is better, and each input uses its own predicted class. Appendix D.1 provides formulas and complementary preservation/removal measurements.

Implementation details. The generator uses five layers of width 384 and validation early stopping. Appendix D.2 specifies fitting, refinement, and baseline settings. Appendix E reports conditioning controls, target-quality sensitivity, seed variability, fidelity curves, parameter sweeps, and Cora transfer. Code is available at `https://anonymous.4open.science/r/STAR-ICLR`.

4.2 Explanation Fidelity and Transfer

Table 1 compares explanatory fidelity under Joint-fit and Held-out generator fitting. **Explanatory fidelity.** Joint-fit exceeds the strongest external baseline in 17 of 18 dataset–architecture combinations

Table 1: Explanatory fidelity. Bold marks best means (including ties), and underlining marks second-best means when the best is unique, using unrounded values.

GNN	Dataset	Metric (↑)	GNNExplainer	PGExplainer	GraphMask	GEAR	IDEA	Graph Curvatures	Held-out	Joint-fit
GCN	MUTAG	Fidelity AUC	0.6113	0.6124	0.6057	0.6090	0.6093	0.6167	0.6367	**0.6368**
		Robust Fidelity AUC	0.6075	0.6026	0.5997	0.6026	0.6019	0.6064	**0.6301**	0.6300
	BBBP	Fidelity AUC	0.7667	0.6609	0.6878	0.7348	0.6743	0.6920	**0.8505**	0.8500
		Robust Fidelity AUC	0.7707	0.6656	0.6687	0.7292	0.6720	0.6917	0.8412	**0.8414**
	BACE	Fidelity AUC	0.5625	0.5492	0.5542	0.5601	0.5489	0.5600	0.5722	**0.5728**
		Robust Fidelity AUC	0.5611	0.5483	0.5545	0.5584	0.5486	0.5580	0.5703	**0.5705**
	HIV	Fidelity AUC	0.6622	0.5952	0.6076	0.6604	0.6011	0.6114	0.7035	**0.7072**
		Robust Fidelity AUC	0.6773	0.6216	0.6380	0.6748	0.6285	0.6352	0.7101	**0.7103**
	Benzene	Fidelity AUC	0.6155	0.6599	0.6619	0.6538	0.6582	0.6599	**0.7670**	0.7663
		Robust Fidelity AUC	0.5983	0.6197	0.6018	0.6113	0.6146	0.6198	**0.7400**	0.7399
	PROTEINS	Fidelity AUC	0.6277	0.5755	0.5588	0.5967	0.5690	0.5803	**0.6549**	0.6547
		Robust Fidelity AUC	0.6271	0.5757	0.5999	0.5965	0.5694	0.5805	0.6539	**0.6539**
GIN	MUTAG	Fidelity AUC	0.5275	0.5205	0.5648	0.5290	0.5287	0.5181	0.5958	**0.5984**
		Robust Fidelity AUC	0.5424	0.5272	0.5526	0.5386	0.5359	0.5272	0.5961	**0.6000**
	BBBP	Fidelity AUC	0.7466	0.6688	0.7040	0.7873	0.5941	0.6689	0.8176	**0.8188**
		Robust Fidelity AUC	0.7639	0.6804	0.7189	0.7944	0.6111	0.6861	0.8284	**0.8296**
	BACE	Fidelity AUC	0.5095	0.5076	0.5115	0.5118	0.5081	0.5113	0.5112	**0.5118**
		Robust Fidelity AUC	0.5119	0.5082	0.5121	0.5144	0.5089	0.5131	0.5137	**0.5145**
	HIV	Fidelity AUC	0.6102	0.5800	0.6013	0.6191	0.5644	0.5919	0.6507	**0.6542**
		Robust Fidelity AUC	0.6123	0.5795	0.6004	0.6113	0.5669	0.5895	0.6491	**0.6529**
	Benzene	Fidelity AUC	0.5220	0.6188	0.6290	**0.6485**	0.5366	0.5864	0.5998	0.6223
		Robust Fidelity AUC	0.6263	0.6828	0.6828	**0.7109**	0.6336	0.6656	0.6771	0.6916
	PROTEINS	Fidelity AUC	0.5745	0.5607	0.5811	0.6028	0.5709	0.5439	0.6090	**0.6108**
		Robust Fidelity AUC	0.5727	0.5581	0.5648	0.6018	0.5698	0.5435	0.6070	**0.6091**
GAT	MUTAG	Fidelity AUC	0.6501	0.6314	0.6409	0.6618	0.6478	0.6564	0.6888	**0.6889**
		Robust Fidelity AUC	0.6509	0.6251	0.6375	0.6533	0.6405	0.6461	**0.6815**	0.6813
	BBBP	Fidelity AUC	0.8015	0.7358	0.7397	0.7777	0.7574	0.7425	0.8621	**0.8641**
		Robust Fidelity AUC	0.8026	0.7326	0.7443	0.7781	0.7545	0.7446	**0.8600**	0.8596
	BACE	Fidelity AUC	0.5508	0.5236	0.5267	0.5504	0.5150	0.5311	0.5784	**0.5793**
		Robust Fidelity AUC	0.5449	0.5191	0.5258	0.5484	0.5131	0.5295	0.5748	**0.5755**
	HIV	Fidelity AUC	0.6531	0.6128	0.6345	0.6117	0.6215	0.6123	0.7206	**0.7250**
		Robust Fidelity AUC	0.7063	0.6748	0.6857	0.6754	0.6753	0.6837	0.7570	**0.7604**
	Benzene	Fidelity AUC	0.6584	0.6873	0.6363	0.7063	0.6053	0.6966	**0.7749**	0.7742
		Robust Fidelity AUC	0.6560	0.6673	0.6192	0.6867	0.6018	0.6807	**0.7610**	0.7597
	PROTEINS	Fidelity AUC	0.6486	0.5729	0.6248	0.6242	0.5582	0.5806	0.6687	**0.6694**
		Robust Fidelity AUC	0.6459	0.5736	0.6238	0.6242	0.5587	0.5801	0.6679	**0.6681**

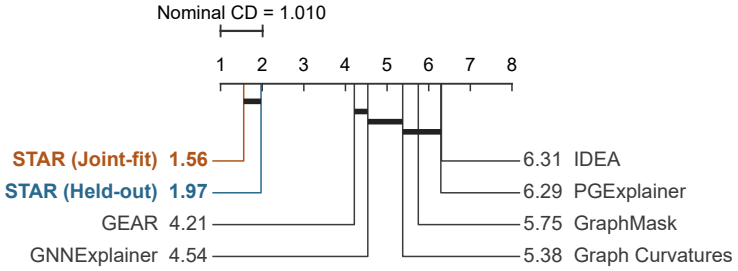

Figure 3: Average ranks across 108 dataset–GNN–seed–AUC tasks. Bars indicate gaps below the nominal Nemenyi CD (1.010, $\alpha = 0.05$), an exploratory reference because tasks are dependent. Lower ranks are better.

on both AUCs, indicating stronger predictive support on original and perturbed inputs. **Transferability.** Held-out retains this advantage in 16 combinations and performs similarly to Joint-fit in most settings. These results support explanation generation for new target pretrained GNNs without refitting the generator, consistent with refinement adapting source-informed guidance to the queried prediction. **Overall comparison.** Joint-fit and Held-out achieve the lowest average ranks across 108 dataset–GNN–seed–AUC tasks (Figure 3). Appendix D.4 specifies the comparison and its dependence assumptions.

Table 2: Component means over three architectures and three seeds. Within each setting, bold marks best (including ties) and underlining marks second best when the best is unique.

		Joint-fit					Held-out				
Dataset	Metric ↑	Generator-only	Local-only	w/o Mask regularization	w/o Prediction alignment	STAR	Generator-only	Local-only	w/o Mask regularization	w/o Prediction alignment	STAR
MUTAG	Fidelity AUC	0.6265	0.6342	0.6372	0.6347	**0.6414**	0.6240	0.6345	0.6366	0.6350	**0.6404**
	Robust Fidelity AUC	0.6201	0.6312	0.6334	0.6316	**0.6371**	0.6177	0.6312	0.6329	0.6319	**0.6359**
BBBP	Fidelity AUC	0.7777	0.8402	0.8429	0.8412	**0.8443**	0.7499	0.8404	**0.8439**	0.8415	0.8434
	Robust Fidelity AUC	0.7790	0.8405	0.8425	0.8409	**0.8435**	0.7529	0.8405	0.8425	0.8413	**0.8432**
Benzene	Fidelity AUC	0.6893	0.7017	0.7077	0.7042	**0.7209**	0.6875	0.7036	0.7080	0.7038	**0.7139**
	Robust Fidelity AUC	0.6826	0.7186	0.7241	0.7191	**0.7304**	0.6847	0.7191	0.7242	0.7204	**0.7260**

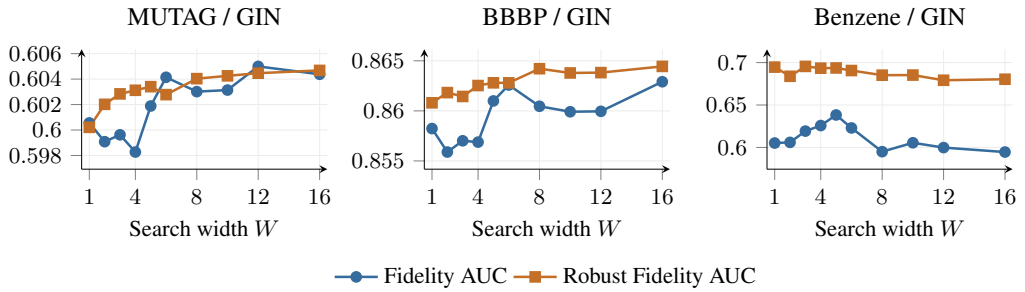

Figure 4: Search-width sensitivity with GIN. Points average up to 16 fixed, correctly predicted test inputs per seed and then three seeds. Vertical scales differ across panels.

4.3 Contribution of the Explanatory Prior and Refinement

Table 2 examines component contributions. *Generator-only* omits refinement. *Local-only* removes explanatory-prior-guided candidates and proposals but retains generated-mask padding. *w/o Mask regularization* removes the Bernoulli-reference penalty. *w/o Prediction alignment* removes both prediction-alignment losses. Complete STAR achieves the best mean AUC in all but one comparison in Table 2, supporting the combined contribution of explanatory-prior generation and target-adaptive refinement. On BBBP under Held-out fitting, removing mask regularization slightly improves Fidelity AUC. Weaker regularization may preserve useful differences between generated edge scores in this setting. Additional results appear in Appendices E.9, E.2, and E.4.

4.4 Search-Width Sensitivity

Search width W controls the number of candidate edge exchanges assessed during refinement. Figure 4 examines its effect on explanation fidelity. MUTAG and BBBP vary by less than 0.008 in Fidelity AUC, indicating limited sensitivity over the evaluated range. Benzene improves from 0.605 at $W = 1$ to 0.638 at $W = 5$, but falls to 0.595 at $W = 16$. Its greater sensitivity may reflect the importance of retaining relevant ring structures, which individual bond exchanges can disrupt. These results favor moderate exploration on Benzene and indicate that search width should account for dataset-specific behavior rather than simply increasing the number of assessed exchanges. Appendix E.12 provides complete results.

5 Conclusion

STAR transfers explanatory information through source-informed explanatory prior generation and target-adaptive explanation refinement. The response-conditioned explanatory prior generator provides guidance without target-specific refitting, while refinement assesses candidate evidence under the target pretrained GNN to address source-to-target mismatches. Experiments on six datasets and three architectures demonstrate explanatory fidelity under joint and held-out generator fitting. Future work further investigates explanation stability and the applicability of STAR in real-world scenarios.

AI USE STATEMENT

GPT is used to correct grammar and polish the language of the manuscript.

REFERENCES

Chirag Agarwal, Owen Queen, Himabindu Lakkaraju, and Marinka Zitnik. Evaluating explainability for graph neural networks. *Scientific Data*, 10(1):144, 2023.

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 *Proceedings of the First Learning on Graphs Conference*, volume 198, pp. 44:1–44:23. PMLR, 2022. URL https://proceedings.mlr.press/v198/amara22a.html.

Steve Azzolin, Antonio Longa, Stefano Teso, and Andrea Passerini. Reconsidering faithfulness in regular, self-explainable and domain invariant gnns. In *International Conference on Learning Representations*, volume 2025, pp. 48644–48677, 2025.

Ngoc Bui, Hieu Trung Nguyen, Viet Anh Nguyen, and Rex Ying. Explaining graph neural networks via structure-aware interaction index. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 4854–4883. PMLR, 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 *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 7712–7730, 2024.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. John Wiley & Sons, 2 edition, 2006. doi: 10.1002/047174882X.

Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine learning research*, 7(Jan):1–30, 2006.

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

Wenya Hu, Jia Wu, and Quan Qian. Cirlexplainer: Causality-inspired explainer for graph neural networks via reinforcement learning. *IEEE transactions on neural networks and learning systems*, 36(6):9970–9984, 2025.

Kexin Huang, Payal Chandak, Qianwen Wang, Shreyas Havaldar, Akhil Vaid, Jure Leskovec, Girish N Nadkarni, Benjamin S Glicksberg, Nils Gehlenborg, and Marinka Zitnik. A foundation model for clinician-centered drug repurposing. *Nature Medicine*, 30(12):3601–3613, 2024a.

Rundong Huang, Farhad Shirani, and Dongsheng Luo. Factorized explainer for graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 12626–12634, 2024b.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

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

Xuyan Li, Jie Wang, and Zheng Yan. Can graph neural networks be adequately explained? a survey. *ACM Computing Surveys*, 57(5):1–36, 2025.

Yazheng Liu, Xi Zhang, Sihong Xie, and Hui Xiong. Robust explanations of graph neural networks via graph curvatures. *Advances in Neural Information Processing Systems*, 38:30680–30717, 2025.

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

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

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 *Proceedings of the 41st International Conference on Machine Learning*, volume 235, 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. *Advances in neural information processing systems*, 33:19620–19631, 2020.

Dongsheng Luo, Tianxiang Zhao, Wei Cheng, Dongkuan Xu, Feng Han, Wenchao Yu, Xiao Liu, Haifeng Chen, and Xiang Zhang. Towards inductive and efficient explanations for graph neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(8):5245–5259, 2024.

Wenzheng Ma, Xiaofeng Liu, Yihu Liu, and Yinglong Ma. Post-hoc explainability of graph neural networks: A comprehensive survey. *Information Sciences*, pp. 123202, 2026.

Siqi Miao, Mia Liu, and Pan Li. Interpretable and generalizable graph learning via stochastic attention mechanism. In *International conference on machine learning*, pp. 15524–15543. PMLR, 2022.

Christopher Morris, Nils M Kriege, Franka Bause, Kristian Kersting, Petra Mutzel, and Marion Neumann. Tudataset: A collection of benchmark datasets for learning with graphs. In *ICML 2020 Workshop on Graph Representation Learning and Beyond*, 2020. URL `https://chrsmrrs.github.io/datasets/`.

Maximilian Muschalik, Fabian Fumagalli, Paolo Frazzetto, Janine Strotherm, Luca Hermes, Alessandro Sperduti, Eyke Hüllermeier, and Barbara E Hammer. Exact computation of any-order shapley interactions for graph neural networks. In *International Conference on Learning Representations*, volume 2025, pp. 36295–36337, 2025.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Yang Qiu, Wei Liu, Jun Wang, and Ruixuan Li. Page: Parametric generative explainer for graph neural network. In *27th European Conference on Artificial Intelligence*, pp. 858–865. SAGE Publications Pvt. Ltd 1 Oliver's Yard, 55 City Road, London, EC1Y 1SP, 2024.

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

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

Minh Vu and My T Thai. Pgm-explainer: Probabilistic graphical model explanations for graph neural networks. *Advances in neural information processing systems*, 33:12225–12235, 2020.

Xiang Wang, Yingxin Wu, An Zhang, Xiangnan He, and Tat-Seng Chua. Towards multi-grained explainability for graph neural networks. *Advances in neural information processing systems*, 34: 18446–18458, 2021.

Zhenqin Wu, Bharath Ramsundar, Evan N Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S Pappu, Karl Leswing, and Vijay Pande. Moleculenet: a benchmark for molecular machine learning. *Chemical science*, 9(2):513–530, 2018.

Yaochen Xie, Sumeet Katariya, Xianfeng Tang, Edward Huang, Nikhil Rao, Karthik Subbian, and Shuiwang Ji. Task-agnostic graph explanations. *Advances in neural information processing systems*, 35:12027–12039, 2022.

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

Zhilin Yang, William Cohen, and Ruslan Salakhudinov. Revisiting semi-supervised learning with graph embeddings. In *International conference on machine learning*, pp. 40–48. PMLR, 2016.

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 *International Conference on Learning Representations*, volume 2026, pp. 13710–13734, 2026.

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

Hao Yuan, Haiyang Yu, Jie Wang, Kang Li, and Shuiwang Ji. On explainability of graph neural networks via subgraph explorations. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 12241–12252. PMLR, 2021. URL https://proceedings.mlr.press/v139/yuan21c.html.

Youmin Zhang, Qun Liu, Guoyin Wang, William K Cheung, and Li Liu. Gear: Learning graph neural network explainer via adjusting gradients. *Knowledge-Based Systems*, 302:112368, 2024.

Yibowen Zhao, Yonghui Xu, Ning Liu, Yixin Zhang, Wei Guo, Xudong Lu, and Lizhen Cui. Causal denoising framework for generalizable recommendation system using graph neural network. In *2024 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6. IEEE, 2024.

Yibowen Zhao, Yonghui Xu, Di Wang, Yixin Zhang, Qingzhong Li, and Lizhen Cui. A multi-objective explanation framework for graph neural networks. *IEEE Transactions on Knowledge and Data Engineering*, 2025.

A NOTATION

Table A.1: Principal notation for the method and its objective and selection properties.

Symbol	Meaning
$G = (V, E, X)$	Input graph, nodes, stored edges, and node attributes
f_θ, C, c_0	Frozen GNN being explained, number of classes, original predicted class
p_0, p_ξ	Predictive distributions on the original and perturbed inputs
π_0, $\chi_0(q)$	Original target-class confidence and candidate confidence relative to it
D_C	Class-normalized KL divergence, D_{KL}/C
m^G, $\widetilde{m}^G$, $m(z)$	Raw explanatory prior, clipped probability-symmetrized explanatory prior, and local mask assessed after sigmoid, clipping, and symmetrization
Sym_G, $\mathcal{C}$, z^G	Edge-group probability averaging, numerical clipping, and logits of the symmetrized explanatory prior
m_0, m_T, $P(m)$	Uniform initial mask, optimized local mask, original-input prediction-preservation and complementary-confidence score
η_h, η_d, $\mathcal{I}$	Initialization entropy weight, discrete-mask alignment weight, and checked fractions
M, ρ, k_ρ	Number of edge budget units, retained fraction, retained count
a_v, b	Input descriptor for each node and response descriptor for the queried prediction
d_a, d_b	Input and response descriptor dimensions, 15 and 40
e_v, $h_v^{(l)}$	Frozen predictor representation and generator hidden state at layer l
g_ϕ, ϕ	Response-conditioned explanatory prior generator and its parameters, frozen at target time
ξ, G_ξ	Perturbation and resulting graph view
a, ℓ_0, ℓ_ξ, $\mathcal{R}_\xi$	Alignment scale, original-input and perturbed-view losses, and perturbation-conditioned risk
λ, τ_g, α_ξ	Perturbation weight, response-similarity tolerance, bounded mixture weight
β, r, $\mathcal{B}$	KL regularization weight, Bernoulli reference rate, mask regularizer
$\mathcal{L}_{\mathrm{shared}}$, $\mathcal{L}_{\mathrm{val}}$, N_{val}	Source-fitting objective, original-input validation score, validation-pair count
A_ξ, $\overline{m}^G$, $\overline{H}(m)$	Affine endpoint alignment, mean symmetrized explanatory prior used for new-edge padding, and mean binary entropy
T, B, W	Local gradient steps, exchange iterations, proposals per iteration
w, q, v_c	Interior mixture scale, local-adjustment factor, intermediate candidate coefficient
v, $\mathcal{V}$	Scalar interpolation coefficient and coefficient set for explanatory-prior-guided initialization
μ, ν, ω	Complementary-confidence, perturbed-view, and entropy weights
τ_{loc}, ν_s	Local response-similarity tolerance and discrete perturbed-view weight
$g_{\min}$, γ_p, γ_u	Gate floor, generator influence, and midpoint-score heuristic contribution
t_p	Explanatory-prior-weight coefficient used to construct search proposals
n_s, n_v, η_v	Search/selection view limits and candidate-dispersion weight
$\mathcal{Q}$, $\mathcal{Q}'$	Views fixed within one search route, and additional selection views shared across candidates
$\mathcal{B}_s$, $\mathcal{B}_v$	Retained fractions for search and candidate comparison, respectively
$\mathbf{r}$, $S_b^{\mathbf{r}}$, $h_{\mathbf{r}}$	Candidate edge-score vector, extracted subgraph at fraction b, reference scale
Δ_b, $\bar{\Delta}$, s_Δ	Paired candidate–reference difference, mean difference, sample standard deviation
$\mathcal{S}_\mathcal{Q}$	Retained/complementary/perturbed-view subgraph score on fixed views
J, h_j	Number of accepted candidate replacements and reference scale at replacement j
δ, ϵ, κ	Per-budget tolerance, improvement margin, dispersion multiplier

B SUPPLEMENTARY METHOD DETAILS

This section expands the descriptors, perturbations, and candidate refinement introduced in Section 3. Definitions follow the order in which STAR constructs and evaluates an explanation. The final subsections analyze the fitting loss, mask regularization, and candidate-selection rule.

Predictive discrepancy uses the class-normalized divergence

$$D_C(p\|q) = \frac{1}{C}\sum_{c=1}^{C} p_c \log \frac{p_c}{q_c}, \tag{A.1}$$

where C denotes the number of classes and p_c, q_c denote class probabilities. This normalization is used consistently in the fitting and refinement objectives.

B.1 Input and Response Descriptors

The descriptors give the explanatory prior generator a common input format for pretrained GNNs with different feature, representation, and output dimensions. Each node receives a descriptor a_v, and the queried prediction receives a response descriptor b. Node attributes contribute through the pretrained GNN's representations and predictions; they are not concatenated directly with the descriptors. For node classification, the reference representation and prediction belong to the queried node. For graph classification, the reference representation is the graph mean and the prediction is graph-level.

Input descriptor. The seven structural coordinates describe connectivity, graph size, and location relative to the queried node. Eight additional coordinates describe the geometry of the pretrained GNN's node representations. Together, these quantities locate a node in the graph and relate its representation to the prediction being explained.

Let d_v^{out} and d_v^{in} be degrees in the stored graph, and let $s_d = \max\{1, \max_{v\in V}(d_v^{\mathrm{out}} + d_v^{\mathrm{in}})\}$. The first three structural coordinates are d_v^{out}/s_d, d_v^{in}/s_d, and $(d_v^{\mathrm{out}} + d_v^{\mathrm{in}})/s_d$. The remaining coordinates are $\log(1 + |V|)/10$, target membership, reciprocal truncated target distance, and four-hop target reachability. Distance is computed on the undirected neighborhood, regardless of stored orientation. For distance $d \le 4$ its reciprocal coordinate is $1/(1 + d)$, and otherwise zero. For graph-level explanation, all three target-specific coordinates are zero. These seven coordinates do not depend on a raw feature vocabulary.

For frozen final-layer representations e_v, let $\bar{e}$ denote the graph mean, e_* the target representation, and $\bar{e}_{\mathcal{N}(v)}$ the mean incident-neighbor representation. An isolated node uses its own representation as the neighbor mean. In graph classification $e_* = \bar{e}$. Each scalar family x_v is standardized as $\mathrm{clip}((x_v - \bar{x})/\max\{\mathrm{sd}(x), 10^{-6}\}, -5, 5)/5$ when noted below, using population standard deviation. The eight node-geometry coordinates are standardized root-mean-square norm, cosine to e_*, standardized distance to e_*, cosine to $\bar{e}$, standardized distance to $\bar{e}$, cosine to the neighbor mean, standardized distance to that mean, and standardized local dispersion. Local dispersion is neighbor distance divided by the sum of the node and neighbor root-mean-square norms, with denominator floor 10^{-6}. Together with the seven structural coordinates, these give $d_a = 15$; e_v denotes the frozen predictor representation, distinct from the generator states $h_v^{(l)}$.

Response descriptor. This descriptor summarizes the prediction, its response to input perturbations, and graph-level representation geometry. The forty coordinates of b comprise eight sorted top probabilities, eight sorted standardized logits, six predictive summaries, ten perturbation-response summaries, and eight graph-level embedding-geometry summaries. Lists shorter than eight are zero-padded, and longer lists are truncated. Predictive summaries are maximum and runner-up confidence, their difference, normalized entropy, squared probability mass, and a scaled log class count. Perturbation summaries are mean/standard deviation/maximum prediction divergence, mean/standard deviation/minimum confidence in the original class, mean/standard deviation predictive margin, and mean/standard deviation normalized entropy. Embedding summaries measure target and centroid norm relative to the mean node norm, target–centroid cosine and relative distance, relative node-norm dispersion, mean/standard deviation endpoint cosine, and mean relative endpoint distance. These eight values are clipped and transformed with $\tanh$ before concatenation.

Logits are centered, divided by their population standard deviation floored at 10^{-6}, clipped to $[-5, 5]$, and scaled by $1/5$. Entropy is divided by $\log\max\{2, C\}$, and the class-count coordinate is $\log(1 + C)/\log 33$. For a probe distribution $\widehat{p}$, the recorded divergence is $|D_C(\widehat{p}\|p_0)|$. Relative geometry uses the mean node root-mean-square norm or mean distance to the graph centroid, respectively, with denominator floor 10^{-6}. Geometry summaries are clipped to $[-10, 10]$ before $\tanh$; all descriptor standard deviations use population normalization.

These scalar summaries have dimensions $d_a = 15$ and $d_b = 40$ regardless of the pretrained GNN's hidden width or class count. Norms, distances, and cosines are unchanged by a common orthogonal transformation of the representations and their references. Sorted classwise statistics are unchanged by a consistent permutation of class coordinates. These properties provide a consistent interface without assuming that different pretrained GNNs use the same decision evidence. Perturbation probes and their sampling schedules are specified in Appendix B.2.

B.2 Perturbation Design and Sampling

STAR uses perturbed inputs to characterize the pretrained GNN's behavior, optimize explanations, and evaluate predictive support. These uses have different roles and sampling schedules. The operations are computational probes of the frozen GNN; they do not assume chemically valid molecular transformations or causal interventions.

Generator fitting. Response descriptors use predictions on fourteen probes of the original input. The probes cycle through seven families: edge dropping (0.10), edge addition (0.05), edge rewiring (0.05), feature masking (0.10), feature noise (0.05), feature dropout (0.05), and feature shuffling (0.05). A separate perturbation G_ξ supplies the second alignment loss in Equation (9). This perturbation follows the local operators defined below, cycling by optimization step and epoch. When the descriptor is recomputed on G_ξ, its probes also follow these local operators, with the same configured probe count. Descriptor probes thus supply generator inputs, whereas the outer perturbation supplies a fitting objective.

Local optimization and candidate assessment. Local relevance estimation draws perturbed inputs during mask optimization. Subgraph search constructs a view set once for each search route and holds it fixed during that search. Final candidate selection constructs an additional view set shared by all candidates. For discrete search and selection, only views with matching original edges and $g_\xi^{\mathrm{loc}} > g_{\min}$ contribute to the score. This restriction keeps comparisons tied to corresponding evidence and sufficiently similar predictions. The number of search and selection views appears in Table A.2.

The local schedule cycles through the following three operators. *Endpoint exchange* selects two edges (a, b) and (c, d) with four distinct endpoints and proposes (a, d) and (c, b). For an undirected graph, reverse entries are updated together. Edge groups are randomly ordered, and up to $\max\{1, \mathrm{round}(0.10M)\}$ disjoint pairs are considered, subject to availability, where M is the number of undirected relations. A collision-free exchange preserves endpoint degrees; duplicate removal after collisions can change them. Directed graphs exchange destinations and exclude self-loops. For node explanations, target-incident edges are protected when sufficient alternatives exist.

Role-matched feature interpolation perturbs attributes using a structurally similar donor node in the same graph. The donor minimizes the ℓ_1 distance between vectors of incoming, outgoing, and total degree, with query-relative membership added for node explanations. Ties follow node-index order. Each eligible node is selected with probability 0.15, and its features are replaced by the mean of its original and donor features. The queried node remains unchanged. For categorical attributes, the resulting values are continuous probes of the GNN.

Query-boundary feature masking selects entries with probability 0.10 on the two-hop boundary outside the queried node and its immediate neighbors. Other non-target nodes are eligible when the boundary is empty. For graph-level explanation, this schedule entry instead uses role-matched interpolation with selection probability 0.10.

Experimental evaluation. Robust Fidelity uses fourteen views, two from each of the seven families listed above, and regenerates an explanation for every view. Prediction-changing views remain in this evaluation. Supplementary fixed-subgraph tests instead retain the original selected edges wherever they survive in a perturbed input. Their experimental sections specify whether views are filtered by prediction similarity. All explanatory methods within a comparison receive the same evaluation views.

Randomness and frozen models. Descriptor probes use an input-derived seed and restore the surrounding random state after probing. On CPU, local optimization uses seed $1729+|E_{\mathrm{stored}}|+|V|$, and search uses $7919 + 2|E_{\mathrm{stored}}| + |V|$. Selection adds seed offset 104729 and strategy-step offset 97. On CUDA, local optimization, search, and selection draw sequentially from the device generator initialized by the explanatory seed, with the same strategy offset. Evaluation view j uses seed

$100003s + j$, and its re-explanation uses $100003s + j + 1$, where s is the experimental seed and j starts at zero. Different seed streams can still produce identical or unchanged graphs. The pretrained GNN always remains in evaluation mode. The explanatory prior generator uses training dropout only during source fitting and remains frozen in evaluation mode during target explanation.

B.3 Generator Fitting and Validation

To associate edge relevance with predictive support across source pretrained GNNs, generator fitting minimizes Equation (10) over source pretrained GNN–input pairs. Gradients through masked predictions update only the generator, with all source pretrained GNNs fixed. Checkpoint selection uses original-input prediction alignment and mask regularization,

$$\mathcal{L}_{\mathrm{val}} = \frac{1}{N_{\mathrm{val}}} \sum_{i=1}^{N_{\mathrm{val}}} \left[\ell_{0,i} + \beta \mathcal{B}(m_i^G, r_i) \right], \tag{A.2}$$

where N_{val} denotes the number of validation GNN–input pairs and r_i denotes the reference rate assigned to pair i. The validation criterion assesses alignment and regularization on original inputs, while generator fitting additionally includes the perturbation-conditioned loss. The selected generator remains fixed during instance-specific inference. Fitting samples r uniformly from the nine decile fractions, independently of the input and mask. Validation cycles through these fractions in item order, assigning one rate to each pair per pass. Table A.2 reports optimization settings.

B.4 Candidate Construction and Selection

Target-adaptive refinement compares evidence guided by the explanatory prior with evidence obtained through local optimization. This section specifies the sequence from continuous masks to discrete subgraphs and final candidate selection. All candidate evaluations use the same frozen target pretrained GNN. Numerical values are collected in Table A.2.

Prepare the explanatory prior and local masks. An undirected relation can have two stored directions with different generated scores. Before local refinement, STAR averages their probabilities to treat the relation consistently. Let $\mathcal{C}(s) = \mathrm{clip}(s, \varepsilon_{\mathrm{num}}, 1 - \varepsilon_{\mathrm{num}})$ act elementwise, and let Sym_G average probabilities within each undirected edge group. For directed inputs, Sym_G is the identity. The raw explanatory prior m^G, its local form $\widetilde{m}^G$, and local mask $m(z)$ satisfy

$$\begin{aligned} \widetilde{m}^G &= \mathrm{Sym}_G(\mathcal{C}(m^G)), \qquad z^G = \mathrm{logit}(\mathcal{C}(\widetilde{m}^G)), \\ m(z) &= \mathrm{Sym}_G(\mathcal{C}(\mathrm{sigmoid}(z))). \end{aligned} \tag{A.3}$$

Thus, opposite-direction probabilities are averaged before taking explanatory-prior logits. Source fitting uses the raw generated mask; local assessment uses the symmetrized mask.

Local optimization starts from $z^{(0)} = 0$ and minimizes Equation (11). Relative complementary confidence uses the numerical form $\chi_0(q) = q_{c_0} / \max\{p_{0,c_0}, \varepsilon_{\mathrm{num}}\}$. The prediction-similarity weight is

$$g_\xi^{\mathrm{loc}} = \mathrm{clip}(\exp[-D_C(p_0 \| p_\xi)/\tau_{\mathrm{loc}}], g_{\min}, 1)\,. \tag{A.4}$$

This weight uses original-to-perturbed divergence, whereas generator fitting uses perturbed-to-original divergence in Equation (9). In both cases, target responses are constant during differentiation. The local entropy term is $\overline{H}(m) = -|E|^{-1} \sum_e [m_e \log m_e + (1 - m_e) \log(1 - m_e)]$.

Match evidence across perturbed inputs. Continuous masks retain their values on edges with matching endpoints. New edges receive the fixed mean explanatory-prior value. For nonempty E, let $\overline{m}^G = |E|^{-1} \sum_{e \in E} \widetilde{m}_e^G$; then

$$(A_\xi m)_{uv} = \begin{cases} m_{uv}, & (u,v) \in E_\xi \cap E, \\ \mathrm{stopgrad}(\overline{m}^G), & (u,v) \in E_\xi \setminus E. \end{cases} \tag{A.5}$$

Here, E_ξ is the perturbed edge set, and $\mathrm{stopgrad}$ holds the padding value constant. The map is affine for a fixed explanatory prior. Gradients pass through matched mask entries, and the continuous perturbation loss is zero when no original edges match. This padding retains generator dependence in the ordinary Local-only variant; Strict Local-only instead uses constant padding and removes the generator entirely (Appendix E.4).

Discrete candidates use $S_\xi = S \cap E_\xi$: deleted edges disappear, and new edges are not selected. Only contributing views enter the average in Equation (13); an empty view set contributes zero perturbation loss. This distinction separates transferring continuous relevance scores from evaluating a fixed selected subgraph.

Construct and assess initial candidates. Let $m_T = m(z^{(T)})$ be the optimized mask and $m_0 = m(0)$ the uniform mask. The local candidate pool contains $\{m_T, m_0\}$. The explanatory-prior-guided pool contains generated and optimized masks together with their logit interpolations,

$$m_\upsilon = \mathrm{Sym}_G\Big(\mathcal{C}\Big(\mathrm{sigmoid}((1-\upsilon)z^G + \upsilon z^{(T)})\Big)\Big), \qquad \upsilon \in \mathcal{V}, \tag{A.6}$$

with

$$\mathcal{V} = \{0, \min(v_c, wq), wq, 1\}. \tag{A.7}$$

The scale w and cap v_c control intermediate mixtures. The adjustment $q \in [0,1]$ reduces their contribution when the uniform mask already explains the prediction well or local optimization worsens the predictive assessment. Both endpoints remain candidates even when $q = 0$.

To compare masks before discrete search, define $P(m) = D_C(p_0 \| f_\theta(G; m)) + \mu\chi_0(f_\theta(G; 1-m))$. Each pool selects a mask minimizing

$$P(m) + \eta_h \overline{H}(m) + \frac{\eta_d}{|\mathcal{I}|} \sum_{\rho \in \mathcal{I}} D_C(p_0 \| f_\theta(G; \mathrm{TopK}_\rho(m))), \tag{A.8}$$

where $\eta_h, \eta_d \geq 0$ weight entropy and thresholded-mask alignment, and $\mathcal{I}$ contains the checked retained fractions. The operator TopK_ρ applies a binary message mask. This assessment considers retention, removal, and the effect of thresholding before the subsequent search evaluates physical edge deletion.

Search discrete subgraphs. Search begins with the selected mask from each pool. For normalized initialization score s_e and explanatory-prior score g_e, proposed edge additions are ranked by

$$\begin{aligned} u_e &= 1 - 2|s_e - 0.5|, \\ v_e &= (1 - \gamma_p t_p) s_e + \gamma_p t_p g_e, \\ a_e &= (1 - \gamma_u(1 - t_p)) v_e + \gamma_u (1 - t_p) u_e. \end{aligned} \tag{A.9}$$

Here, $t_p \in [0,1]$ controls explanatory-prior guidance, and $\gamma_p, \gamma_u \in [0,1]$ control the contributions of generated scores and intermediate-relevance edges. The local route sets $t_p = 0$. The explanatory-prior-guided route adjusts t_p according to $P(\widetilde{m}^G)$, assigning greater weight when the generated mask preserves the prediction and its complement has lower target-class confidence. The quantity u_e promotes exploration of intermediate scores; it is not a calibrated uncertainty estimate.

At each retained fraction, the search considers removing low-ranked selected edges and adding edges from the bounded proposal pool. Up to W paired exchanges are assessed in each of B iterations. An exchange is accepted only if it decreases the composite score while preserving the retained-edge count. Fractions are processed in ascending order, and edges finalized at smaller fractions remain protected. For a graph with M undirected relations or directed edges, the retained count is $k_\rho = \max\{1, \min\{M, \mathrm{round}(\rho M)\}\}$. Opposite directions of an undirected relation are retained together. The search properties are derived in Appendix B.7.

Select the final explanation. The refined local ordering is the initial reference. The generated-mask and refined explanatory-prior-guided orderings are assessed using one additional view set $\mathcal{Q}'$ shared across all candidates. For an ordering $\mathbf{r}$, let $S_b^{\mathbf{r}}$ be its extracted subgraph at retained fraction $b \in \mathcal{B}_v$. Its comparison score $R_{\mathbf{r}}$ is the mean $\mathcal{S}_{\mathcal{Q}'}(S_b^{\mathbf{r}})$ across these fractions plus η_v times their population standard deviation.

Candidates are examined in increasing $R_{\mathbf{r}}$. Against the current reference $\mathbf{r}_0$, define $\Delta_b = \mathcal{S}_{\mathcal{Q}'}(S_b^{\mathbf{r}}) - \mathcal{S}_{\mathcal{Q}'}(S_b^{\mathbf{r}_0})$ and $h_{\mathbf{r}_0} = \max\{1, |R_{\mathbf{r}_0}|\}$. Replacement requires

$$\bar{\Delta} + \kappa s_\Delta / \sqrt{|\mathcal{B}_v|} \leq -\epsilon h_{\mathbf{r}_0}, \qquad \max_{b \in \mathcal{B}_v} \Delta_b \leq \delta h_{\mathbf{r}_0}, \tag{A.10}$$

where $\bar{\Delta}$ and s_Δ denote the mean and sample standard deviation of paired score differences. For one checked fraction, $s_\Delta = 0$. The first condition requires a mean improvement after penalizing

variation across fractions; the second limits deterioration at any individual fraction. The coefficients are deterministic selection settings, not calibrated confidence levels. An accepted candidate becomes the next reference. The resulting edge scores determine the final explanatory subgraph through grouped top-k extraction.

Clipping, coefficient thresholds, finite-score fallbacks, and tie conventions are documented together in the accompanying `implementation_notes.md`. These conventions specify the numerical implementation of the equations above without changing the two candidate sources or their target-GNN assessment.

B.5 Properties of the Perturbation Loss

For nonnegative losses and $\alpha = \alpha_\xi \in [0, 1]$, Equation (9) has the exact form

$$\mathcal{R}_\xi = \begin{cases} \ell_0, & \ell_\xi \le \ell_0, \\ (1-\alpha)\ell_0 + \alpha\ell_\xi, & \ell_\xi > \ell_0. \end{cases} \tag{A.11}$$

Equivalently,

$$\mathcal{R}_\xi = \max_{u \in [0,\alpha]} \{(1-u)\ell_0 + u\ell_\xi\}. \tag{A.12}$$

The maximand is affine in u: its maximum is at zero when $\ell_\xi \le \ell_0$ and at α otherwise. Consequently, $\ell_0 \le \mathcal{R}_\xi \le \max(\ell_0, \ell_\xi)$ and $\alpha\ell_\xi \le \mathcal{R}_\xi$. At $\alpha = 0$, the loss is simply ℓ_0. With detached α and $\ell_\xi > \ell_0$, its gradient is the convex combination $(1-\alpha)\nabla_\phi\ell_0 + \alpha\nabla_\phi\ell_\xi$; the lower bound on the loss does not assert a unit coefficient on the original-input gradient.

For $d_\xi = D_C(p_\xi \| p_0) \ge 0$, the implemented weight satisfies $0 \le \alpha_\xi \le 1 - e^{-\lambda} < 1$ and decreases with d_ξ when $\lambda > 0$. Thus similar full-input responses increase the penalty only when perturbed-view alignment is worse. These are properties of the continuously masked losses with a mask recomputed on each input, not guarantees about selected edge identities, discrete-subgraph fidelity, or transfer to another predictor.

B.6 Effect of the Mask Regularizer

The Bernoulli-reference regularizer in Equation (10) is

$$\mathcal{B}(m, r) = \frac{1}{|E|} \sum_e \left[m_e \log \frac{m_e}{r} + (1 - m_e) \log \frac{1 - m_e}{1 - r} \right]. \tag{A.13}$$

Here, m_e and r are clipped to $[\varepsilon_{\text{num}}, 1 - \varepsilon_{\text{num}}]$ before logarithms. This expression is the mean Bernoulli KL divergence defined in Section 3.2. The term regularizes generated relevance probabilities and is not used as an information-theoretic claim about the final explanation. During source fitting, r is sampled uniformly from $\{0.1, \ldots, 0.9\}$ independently of the current generator input and mask. For a fixed scalar mask value $s \in (0, 1)$ and binary entropy $h(s) = -s \log s - (1 - s) \log(1 - s)$, expanding Bernoulli KL and using $\mathbb{E}_r \log r = \mathbb{E}_r \log(1 - r)$ gives

$$\mathbb{E}_r D_{\text{KL}}(\text{Ber}(s) \| \text{Ber}(r)) = c_{\text{ref}} - h(s), \qquad c_{\text{ref}} = -\frac{1}{9} \sum_{k=1}^{9} \log(k/10), \tag{A.14}$$

$$\frac{d}{ds}[c_{\text{ref}} - h(s)] = \log \frac{s}{1 - s}, \qquad \frac{d^2}{ds^2}[c_{\text{ref}} - h(s)] = \frac{1}{s(1 - s)} > 0.$$

Hence the regularizer-only minimum occurs at $s = 0.5$. Averaging over edges gives $\mathbb{E}_r \mathcal{B}(m, r) = c_{\text{ref}} - \overline{H}(m)$, with the numerical clipping convention applied to the evaluated probabilities. This is an expectation at a fixed mask, not a statement about every stochastic update or the optimum of the complete objective. It does not apply itemwise to the cyclic validation assignment in Appendix B.3. The expected term discourages extreme generated probabilities rather than imposing sparsity. Retained-edge budgets control explanation size, while the positive local entropy term in Equation (11) favors sharper instance-specific masks.

B.7 Score Control during Subgraph Refinement

Within-route exchanges. Fix a route's scoring views $\mathcal{Q}$, full-input reference predictions, and retained count k, and assume finite scores. The implementation accepts a proposed set S_{t+1} only when

$$\mathcal{S}_{\mathcal{Q}}(S_{t+1}) < \mathcal{S}_{\mathcal{Q}}(S_t) - 10^{-5}\max\{1, |\mathcal{S}_{\mathcal{Q}}(S_t)|\}. \tag{A.15}$$

An unsuccessful proposal leaves S_t unchanged. The score is therefore nonincreasing within the fixed-budget search. Each exchange preserves k and draws removals only from the segment beyond the previously finalized prefix. Later budgets therefore preserve earlier finalized sets in the internal ordering; repeated rounded counts are skipped. The comparison is with the current budget's starting set, which already includes earlier prefix refinements, not necessarily with the original seed at that budget.

Candidate-selection bounds. Fix one shared selection-view set $\mathcal{Q}'$ and a nonempty set of checked fractions $\mathcal{B}_v$, and assume finite candidate scores. Let $J \le 2$ be the number of accepted replacements, $\mathbf{r}_0$ the refined local score vector, and $\mathbf{r}_j$ the selected vector after replacement j. Write $h_j = h_{\mathbf{r}_{j-1}}$ and $\Delta_{j,b} = \mathcal{S}_{\mathcal{Q}'}(S_b^{\mathbf{r}_j}) - \mathcal{S}_{\mathcal{Q}'}(S_b^{\mathbf{r}_{j-1}})$ for the subgraphs actually extracted for scoring. Equation (A.10) implies

$$\overline{\Delta}_j \le -\epsilon h_j - \frac{\kappa s_{\Delta_j}}{\sqrt{|\mathcal{B}_v|}} \le -\epsilon h_j, \qquad \Delta_{j,b} \le \delta h_j \quad (b \in \mathcal{B}_v). \tag{A.16}$$

Summing over accepted replacements telescopes, giving the mean-score bound

$$\frac{1}{|\mathcal{B}_v|}\sum_{b\in\mathcal{B}_v} \mathcal{S}_{\mathcal{Q}'}(S_b^{\text{out}}) \le \frac{1}{|\mathcal{B}_v|}\sum_{b\in\mathcal{B}_v} \mathcal{S}_{\mathcal{Q}'}(S_b^{\text{local}}) - \epsilon\sum_{j=1}^{J} h_j, \tag{A.17}$$

and, at every checked fraction b, the per-budget bound

$$\mathcal{S}_{\mathcal{Q}'}(S_b^{\text{out}}) \le \mathcal{S}_{\mathcal{Q}'}(S_b^{\text{local}}) + \delta\sum_{j=1}^{J} h_j. \tag{A.18}$$

If $J = 0$, the output is the local candidate and the sums are zero. There are at most two replacements because only the generated-mask and explanatory-prior-guided alternatives are tested. These deterministic bounds concern the composite score on the checked fractions. Individual score components and unchecked fractions remain subject to empirical evaluation.

Extraction and ties. Search returns rank-encoded scores after accepting an exchange, but retains the original seed scores when no exchange improves the score. The generated-mask candidate also uses its original scores. Extraction applies grouped top-k selection separately at each budget. Tied scores, including a uniform mask, need not produce prefixes of one common ordering. Distinct scores or a shared fixed tie order ensure nested extraction; the latter is a sufficient condition, not an asserted implementation convention. Equations (A.17) and (A.18) concern the scoring sets; applying them to later extraction requires reproduction of those sets.

C Pseudocode

Algorithms A.1 and A.2 summarize source fitting and single-input explanation. Only the response-conditioned explanatory prior generator parameters are fitted on source inputs. At target time, its explanatory prior initializes candidates and guides search. The generator and target GNN remain frozen while instance-specific masks and edge subsets are refined. The inference algorithm assumes a nonempty edge set and available message-mask hooks. Numerical fallbacks are listed in the accompanying code documentation.

The algorithms use the perturbation schedules in Appendix B.2 and the mask and candidate definitions in Appendix B.4. The returned scores are converted to an explanatory subgraph by grouped top-k selection.

Algorithm A.1 Optimization of the response-conditioned explanatory prior generator

Require: Source GNNs with frozen parameters, source training and validation inputs
Initialize the generator parameters ϕ.
repeat
  for each shuffled source GNN–input pair (f_θ, G) **do**
    Draw perturbation ξ from the scheduled local strategy and independently sample reference rate r.
    Construct descriptors for G with configured perturbation probes and for G_ξ with local-policy probes.
    Generate edge-relevance masks m^G and m^G_ξ using g_ϕ.
    Evaluate prediction-preservation risk and mask regularization in Equation (10).
    Update only ϕ through the masked, frozen GNN.
  end for
  Evaluate Equation (A.2) and retain the best checkpoint.
  Reduce the learning rate on a validation plateau.
until validation early stopping or the epoch limit
return The validation-selected generator g_ϕ.

Algorithm A.2 Target-adaptive explanation refinement

Require: Frozen target GNN f_θ and generator g_ϕ, input graph G, search fractions $\mathcal{B}_s$
Construct descriptors and generate explanatory prior m^G using g_ϕ.
if $T = 0$ or $w = 0$ **then**
  return m^G.
end if
Clip and symmetrize explanatory prior probabilities before taking logits, as in Equation (A.3).
Initialize edge logits at zero and optimize Equation (11) for T steps.
Set the interpolation factor from the initial and post-update predictive/complementary proxies.
Select a local initialization from optimized/neutral masks and an explanatory-prior-guided initialization from generated/local endpoints and interpolations.
for each of the two initializations **do**
  Construct its bounded candidate acquisition order.
  Construct this route's scoring views $\mathcal{Q}$ once; keep them fixed throughout its search.
  for each ascending retained fraction in $\mathcal{B}_s$ **do**
    Protect the previously finalized prefix of the internal search ordering.
    Skip fractions whose rounded retained count is already finalized.
    Test up to W exchanges in each of B iterations using Equation (13).
    Accept only improving exchanges and finalize the current retained prefix.
  end for
  Set route scores to rank-encoded scores after accepted exchanges; retain seed scores if no exchange improves the score.
end for
Set checked fractions $\mathcal{B}_v$ to the search fractions and intervening deciles (Appendix B.4).
Construct one additional selection-view set $\mathcal{Q}'$ and reuse it for all candidates and checked fractions.
Score the generated-mask, refined local, and refined explanatory-prior-guided candidates on $\mathcal{Q}'$ at $\mathcal{B}_v$.
Starting from local, test the other candidates in increasing population-penalized score, the generated-mask candidate first on a tie.
Apply Equation (A.10) and update the reference after each accepted replacement.
return Selected edge scores.

D Experimental Design

This section defines the evaluation measures, fixed experimental settings, and aggregation procedures. Method equations and perturbation operations are specified in Appendix B; supplementary experiments use the common comparison conventions in Appendix E.1.

D.1 METRICS

Fidelity and Robust Fidelity evaluate how well the selected explanatory subgraph preserves the queried prediction, following the sufficiency-oriented evaluation of Amara et al. (2022); Agarwal et al. (2023) and the perturbation-based assessment of Liu et al. (2025). The confidence-based definitions used in this study are

$$\begin{aligned} \mathrm{Fidelity}_{\rho}(G) &= f_\theta(G[S_\rho(G)])_{c(G)}\,, \\ \mathrm{RobustFidelity}_{\rho}(G) &= \frac{1}{|\mathcal{V}_G|}\sum_{\xi\in\mathcal{V}_G} f_\theta(G_\xi[S_\rho(G_\xi)])_{c(G_\xi)}\,, \end{aligned} \tag{A.19}$$

where $S_\rho(H)$ is recomputed on input H at retained fraction ρ, $c(H) = \arg\max_c f_\theta(H)_c$, and $\mathcal{V}_G$ contains successful evaluations among fourteen prescribed views, including prediction-changing views. Fixed-subgraph support and edge-set consistency are evaluated separately in Appendix E.6. Per-target normalized *AUC* integrates $\rho = 0.1, \ldots, 0.9$ by the trapezoidal rule and divides by 0.8; *Best* takes the maximum. Results average targets and then seeds. Failed views are recorded for coverage reporting. Annotation-based scores are excluded from the main comparisons and ranks because edge rationales are not available benchmark-wide.

Perturbation families, sampling strengths, and random-seed schedules are specified together in Appendix B.2. The same retained-edge grouping and rounding conventions apply to every explanatory method.

Supplementary controls also distinguish confidence preservation from the effect of removing selected evidence (Amara et al., 2022). The absolute sufficiency gap is $|f_\theta(G)_{c(G)} - f_\theta(G[S_\rho(G)])_{c(G)}|$, where smaller values indicate closer preservation of the original confidence. Comprehensiveness is $f_\theta(G)_{c(G)} - f_\theta(G[E \setminus S_\rho(G)])_{c(G)}$, where larger values indicate a greater confidence decrease after removing the selected edges. Their AUCs use the same fractions and normalization.

D.2 IMPLEMENTATION DETAILS

Table A.2 collects the architecture, optimization, and inference settings. Target GNNs are trained on the training split and selected by validation ROC AUC on BBBP, BACE, and HIV, or validation balanced accuracy on the remaining datasets. Their parameters remain frozen throughout explanatory-prior fitting, baseline fitting, and explanation evaluation.

The explanatory prior generator processes shuffled source GNN–input pairs individually and is optimized with AdamW (Loshchilov & Hutter, 2019). Joint-fit uses all source datasets; Held-out excludes the evaluated dataset and its pretrained GNNs as specified in Appendix D.3. Equation (A.2) selects the generator checkpoint using source validation pairs. The selected generator remains frozen during target explanation; only the instance-specific mask and selected edges change.

D.2.1 FROZEN TARGET QUALITY

All 54 frozen target models remain in the main comparison. Five predict a single class on the full test set: BACE–GCN seeds 0 and 2, BACE–GIN seeds 1 and 2, and HIV–GIN seed 2. Their balanced accuracy is 0.5; fidelity on correctly predicted inputs does not establish useful full-test discrimination. Appendix E.8 reports the common exclusion analysis. The explanation population contains 4,750 target occurrences: ten configurations exhaust fewer than 100 correct predictions, and the remaining 44 supply 100 each, with no incorrectly predicted fallback targets. Exact uncapped eligible totals and full-test class histograms are unavailable. Saved accuracy and balanced accuracy cover all models; ROC AUC is missing for seed 0 on MUTAG, Benzene, and PROTEINS.

D.2.2 BASELINE ADAPTERS

GNNExplainer uses 12 optimization iterations in the main comparison and 50 in the Cora extension and the additional paired control in Appendix E.4. These iteration counts do not equalize total model queries with STAR. PGExplainer and IDEA use 50 fitting epochs, learning rate 0.003, and at most 128 training graphs. IDEA uses batches of 32, a codebook of size 32, and distribution-alignment

Table A.2: Experimental configuration. All settings remain fixed during test explanation.

Component	Setting	Value
Target GNN	Message-passing layers; hidden width	3; 128
	Dropout; GAT heads; graph pooling	0.5; 4; mean
	Adam learning rate; weight decay	10^{-3}; 5×10^{-4}
	Epoch limit; early-stopping patience	200; 50
	Batch size	32 molecular graphs; 64 protein graphs
	LR plateau patience; factor; minimum	20; 0.5; 10^{-6}
	Class-weighting imbalance threshold	1.5
Generator	GIN layers; hidden width; GIN epsilon	5; 384; 0 (fixed)
	Descriptor dimensions; dropout	$d_a = 15$, $d_b = 40$; 0.1
	Source targets per GNN–dataset pair	Up to 128 training and 64 validation
	AdamW learning rate; weight decay	10^{-4}; 10^{-4}
	Epoch limit; early-stopping patience	120; 10
	Minimum improvement; gradient clip	5×10^{-5}; norm 5
	LR plateau patience; factor; minimum	5; 0.5; 10^{-6}
	Objective weights; prediction tolerance	$a = 2$, $\beta = 0.2$, $\lambda = 0.25$; $\tau_g = 0.05$
	Reference rates	$r \in \{0.1, 0.2, \ldots, 0.9\}$
Refinement	Adam updates; learning rate	$T = 12$; 0.1
	Local objective weights	$\mu = 0.35$, $\nu = 0.5$, $\omega = 0.01$
	Prediction tolerance; minimum weight	$\tau_{\text{loc}} = 0.25$; $g_{\min} = 0.05$
	Initialization weights; checked fractions	$\eta_h = 0.002$, $\eta_d = 0.25$; $\mathcal{I} = \{0.2, 0.5, 0.8\}$
	Interpolation scale; intermediate cap	$w = 0.75$; $v_c = 0.35$
	Proposal weights	$\gamma_p = 0.65$, $\gamma_u = 0.25$
	Search iterations; width; perturbation weight	$B = 2$; $W = 4$; $\nu_s = 0.12$
	Maximum search and selection views	$n_s = 2$; $n_v = 3$
	Search and selection fractions	$\mathcal{B}_s = \mathcal{B}_v = \{0.1, 0.2, \ldots, 0.9\}$
	Selection dispersion and acceptance	$\eta_v = 0.25$, $\kappa = 1.645$, $\epsilon = 10^{-4}$, $\delta = 0.01$
	Numerical probability floor	$\varepsilon_{\text{num}} = 10^{-6}$

weight 1, with its adapter retaining complete training batches. GraphMask uses 50 epochs per input and learning rate 0.01.

GEAR fits its graph-level mask module for 50 epochs on the target training split using Adam at 0.003 and deterministic masks. Its multiobjective adapter accumulates predictive-agreement, representation-alignment, counterfactual, size, and entropy gradients across training graphs before coordination. Prediction and counterfactual objectives define the dominant direction at 45 degrees; size and entropy coefficients are 0.05 and 1. All reported GEAR scores use this multiobjective adapter.

Graph Curvatures uses the same PGExplainer configuration, Ricci parameter 0.5, and at most eight validation graphs to select a curvature weight from $\{0, 0.05, 0.1, 0.2, 0.5, 1\}$. Selection at retained fraction 0.2 combines retained confidence, confidence reduction after removing the selected subgraph, and sufficiency gap with coefficients $-1, -0.25, 0.25$, respectively, choosing the smallest mean score. The selected weight remains fixed during test explanation. All target GNN parameters remain frozen in these procedures.

D.3 Data Splits and Fitting Scope

Seeds 0, 1, and 2 define label-stratified random 70/10/20% train/validation/test splits. These are random molecular splits, so related scaffolds can occur in different partitions (Wu et al., 2018). Class-wise permutations determine the partitions, using floored proportions with minimum-count safeguards for small classes. Test targets are correctly predicted graphs selected in shuffled, class-wise round-robin order, up to 100 per dataset–GNN–seed configuration. Available correct predictions limit the attainable class balance. All compared explanatory methods use the same checkpoints, target instances, and retained fractions.

Joint-fit fits one explanatory prior generator per seed across all six datasets and three architectures. Held-out excludes the evaluated dataset and all pretrained GNNs trained on that dataset from generator fitting and checkpoint validation. The architecture families remain represented in the source set, and both settings use the same white-box target inference. Parameterized baselines fit on target-dataset training instances and the target GNN's responses. No explanatory method uses annotated edge rationales.

Inference settings are selected once on pooled benchmark-validation configurations. Selection maximizes the smaller of the Joint-fit and Held-out mean AUCs, with their joint mean breaking ties. The resulting settings are fixed for testing. Held-out therefore excludes target data from generator fitting and checkpoint selection, but not from inference-hyperparameter validation.

D.4 Aggregation and Critical Difference Analysis

Each main-table value averages target-specific AUCs within a seed and then gives the three seeds equal weight. Figure 3 ranks the same eight explanatory methods separately for each dataset–GNN–seed–metric task, using only Fidelity AUC and Robust Fidelity AUC. The $6 \times 3 \times 3 \times 2 = 108$ tasks receive equal weight. For task t and method j,

$$\overline{r}_j = \frac{1}{N} \sum_{t=1}^{N} \mathrm{rank}_j(-y_{t\cdot}), \qquad N = 108, \tag{A.20}$$

where y_{tj} denotes the unrounded seed-specific AUC underlying the main table. Larger AUCs receive lower ranks, with average ranks for exact ties. Best scores, Edge-F1, and individual retained fractions are excluded. Joint-fit and Held-out obtain mean ranks 1.556 and 1.972, respectively.

Using the conventional Nemenyi formula (Demšar, 2006) with $k = 8$ methods gives the nominal threshold

$$\mathrm{CD}_{\mathrm{nom}} = q_{0.05} \sqrt{\frac{k(k+1)}{6N}} = 1.010, \tag{A.21}$$

where $q_{0.05} = 3.031$ is the infinite-degrees-of-freedom studentized-range critical value divided by $\sqrt{2}$. Bars connect maximal groups whose rank gaps do not exceed this threshold. Tasks share datasets and checkpoints, and the two AUCs use related predictions. The independent-block assumption is therefore not established, so this task-level CD is an exploratory reference, not a calibrated significance test. The average ranks describe performance across the evaluated tasks.

As a dataset-block sensitivity analysis, averaging seeds, architectures, and the two AUCs within each dataset gives six blocks. The Friedman statistic is $\chi_F^2 = 36.444$ (seven degrees of freedom, asymptotic $p = 5.98 \times 10^{-6}$), and the Nemenyi CD is 4.286. Joint-fit and Held-out retain the lowest ranks, 1.00 and 2.00. Joint-fit differs significantly from PGExplainer, Graph Curvatures, and IDEA, while Held-out differs from PGExplainer and IDEA under this dataset-block analysis. Other pairwise gaps do not exceed the threshold. Statistics use SciPy 1.16.3.

D.5 Explanation Cost

On MUTAG, BBBP, BACE, and HIV (36 backbone–seed configurations; 2,950 paired target occurrences), first-explanation forward calls average 813.50 for Joint-fit, 814.75 for Held-out, and 348.31 for Strict Local-only. Joint-fit uses approximately 2.34 times as many calls, including response probes, a second search route, and candidate selection. All three record 36 backward-hook invocations, counting differentiated forward branches rather than optimizer steps. Generator-only uses 15 forwards and GEAR 3, both without backward calls. Means weight targets within configurations, then backbones, seeds, and datasets equally; training/preparation, repeat explanations, and external evaluation are excluded. Concurrent jobs preclude a fair wall-clock comparison. The initialization controls retain both routes but are procedure-matched, not exactly query-matched (Appendix E.5).

E Supplementary Experiments

These experiments examine the roles of target-GNN conditioning and candidate refinement, then report transfer, target-quality sensitivity, and detailed benchmark results.

E.1 Common Experimental Settings

Unless stated otherwise, the experiments use seeds 0, 1, and 2, the main-study pretrained GNNs and test targets, and retained fractions $0.1, \ldots, 0.9$. Joint-fit and Held-out retain the source scopes defined in Appendix D.3. Paired controls use the same target GNN, input, retained fractions, and evaluation perturbations; each subsection identifies the factor that changes. Generator parameters remain fixed during explanation unless generator fitting itself is the factor under study.

AUCs are computed per target using Appendix D.1 and averaged within each dataset–architecture–seed configuration. Reported means and sample standard deviations use the three seed-specific results. Comparisons that combine architectures average them within each seed; cross-dataset summaries give datasets equal weight. For paired differences, subtraction precedes aggregation. Counts of positive comparisons describe the direction of observed effects, not independent significance tests. Each experiment reports its dataset and architecture coverage, with any departures from these conventions stated locally.

Scope of the comparisons. The joint and held-out results assess complete-method fidelity under the specified generator-fitting protocol. Random molecular splits and computational perturbations do not establish unseen-scaffold generalization or chemical validity. Matched transfer comparisons with ReFine and TAGE, and exactly query-matched generator controls, remain separate evaluations; explanation cost is documented in Appendix D.5.

E.2 Target-Model Conditioning on the Same Input

This experiment tests whether the explanatory prior depends usefully on information from the GNN being explained, beyond the input's structure. For each test graph, a frozen GIN remains the evaluated predictor and supplies all local-refinement responses. The generator receives either its matched GIN descriptors or descriptors computed from a frozen GCN or GAT on the same graph. Donor GNNs are trained on the same dataset and seed-specific split. Only the generator's predictive-response and representation-geometric descriptors are substituted; the input, structural descriptors, generator parameters, refinement procedure, target class, and retained fractions remain fixed. The comparison is repeated before and after refinement, under Joint-fit and Held-out generator fitting.

The experiment covers all six datasets under both fitting scopes, giving 36 GNN–dataset–seed–scope configurations and 3,168 test-instance occurrences. Each occurrence is evaluated with matched GIN descriptors and both donor alternatives, before and after refinement. The table reports paired Fidelity AUC differences under the common aggregation protocol.

Matched conditioning improves generator-only predictive support. Matched descriptors produce higher generator-only Fidelity AUC than both donor alternatives on all six datasets under both fitting scopes, giving 24 positive dataset–scope–donor comparisons (Table A.3). After refinement, 22 of these 24 differences remain positive. The two reversals occur on HIV with GCN descriptors, and the refined differences are generally smaller than the generator-only differences. Thus, conditioning affects the quality of the initial edge ordering, while target-specific refinement often reduces sensitivity to that initialization. Because the evaluated predictor is always GIN, this experiment isolates descriptor matching and does not measure transfer to an unseen architecture family.

Prediction preservation and removal impact. Matched conditioning also lowers the generator-only absolute sufficiency gap in 22 of 24 comparisons and raises comprehensiveness in all 24. After refinement, the corresponding counts are 23 and 19. Table A.4 summarizes these measures with equal dataset weights. This corroborates the generator-only Fidelity result through closer preservation of original confidence and a greater decrease in confidence when the selected edges are removed. The measurements concern dependence of the fixed GIN's predictions on selected evidence, not agreement with annotated chemical rationales.

Separating predictive responses from representation geometry. A seven-condition factorial control on MUTAG, BBBP, and Benzene distinguishes the two model-dependent descriptor blocks in Appendix B.1. For each donor, it substitutes either the 32-dimensional predictive-response block, the representation-geometric block comprising eight global and eight node-level statistics, or both. The matched condition supplies the seventh arm. Structural information is unchanged. All seven arms are evaluated together in fresh paired runs with the same frozen checkpoints and test targets; results are not pooled with the preceding three-condition runs.

Table A.3: Target-model conditioning on the same input. Entries are paired Fidelity AUC differences, matched GIN conditioning minus GCN- or GAT-donor conditioning, multiplied by 100 (percentage points). Means and sample standard deviations are computed over three seed-specific paired target averages. Positive values favor matched conditioning. The evaluated predictor remains GIN in every condition; donor architectures supply descriptors only.

		Generator-only		After refinement	
Dataset	Fitting scope	GCN donor	GAT donor	GCN donor	GAT donor
MUTAG	Joint-fit	$+1.137 \pm 1.618$	$+0.577 \pm 0.521$	$+1.292 \pm 2.188$	$+0.867 \pm 1.327$
MUTAG	Held-out	$+0.430 \pm 0.523$	$+1.107 \pm 1.312$	$+0.781 \pm 1.154$	$+1.022 \pm 1.109$
PROTEINS	Joint-fit	$+1.900 \pm 0.975$	$+2.201 \pm 1.783$	$+0.202 \pm 0.061$	$+0.388 \pm 0.162$
PROTEINS	Held-out	$+1.773 \pm 1.334$	$+1.746 \pm 1.507$	$+0.267 \pm 0.218$	$+0.400 \pm 0.275$
BBBP	Joint-fit	$+3.280 \pm 1.056$	$+2.917 \pm 0.663$	$+0.034 \pm 0.048$	$+0.292 \pm 0.028$
BBBP	Held-out	$+3.168 \pm 1.931$	$+2.869 \pm 2.219$	$+0.111 \pm 0.319$	$+0.232 \pm 0.210$
BACE	Joint-fit	$+0.370 \pm 0.621$	$+0.203 \pm 0.357$	$+0.092 \pm 0.160$	$+0.063 \pm 0.109$
BACE	Held-out	$+0.227 \pm 0.352$	$+0.178 \pm 0.316$	$+0.029 \pm 0.050$	$+0.022 \pm 0.039$
HIV	Joint-fit	$+1.316 \pm 2.834$	$+1.277 \pm 1.177$	-0.135 ± 0.218	$+0.138 \pm 0.126$
HIV	Held-out	$+2.758 \pm 3.102$	$+2.638 \pm 2.434$	-0.102 ± 0.530	$+0.189 \pm 0.238$
Benzene	Joint-fit	$+5.140 \pm 4.482$	$+3.852 \pm 2.069$	$+1.110 \pm 0.868$	$+1.025 \pm 1.656$
Benzene	Held-out	$+7.747 \pm 5.205$	$+4.661 \pm 3.553$	$+1.127 \pm 1.890$	$+0.405 \pm 0.465$

Table A.4: Prediction preservation and removal impact in the same-input model-conditioning control. Differences are matched minus donor normalized AUC, multiplied by 100. Datasets are weighted equally within each seed before computing the three-seed mean and sample standard deviation. Favorable counts use the six dataset-specific seed means. Lower absolute sufficiency gap and higher comprehensiveness favor matched conditioning.

Stage	Fitting scope	Donor	Sufficiency gap ↓	Favorable	Comprehensiveness ↑	Favorable
Generator-only	Joint-fit	GCN	-2.175 ± 1.310	6/6	$+1.792 \pm 1.133$	6/6
Generator-only	Joint-fit	GAT	-1.587 ± 0.513	6/6	$+1.258 \pm 0.382$	6/6
Generator-only	Held-out	GCN	-2.323 ± 1.470	5/6	$+2.001 \pm 1.122$	6/6
Generator-only	Held-out	GAT	-1.534 ± 0.791	5/6	$+1.929 \pm 0.251$	6/6
Refined	Joint-fit	GCN	-0.469 ± 0.331	6/6	$+0.209 \pm 0.408$	5/6
Refined	Joint-fit	GAT	-0.488 ± 0.224	6/6	$+0.171 \pm 0.058$	5/6
Refined	Held-out	GCN	-0.352 ± 0.442	5/6	$+0.084 \pm 0.207$	5/6
Refined	Held-out	GAT	-0.282 ± 0.046	6/6	$+0.106 \pm 0.131$	4/6

Matched representation geometry yields higher Fidelity AUC in all 12 dataset–scope–donor comparisons both before and after refinement. Predictive-response substitution alone has mixed effects, with positive matched-minus-donor differences in 5 of 12 generator-only comparisons and 8 of 12 refined comparisons. The result identifies representation geometry as the more consistent source of model-specific conditioning in these tests. It does not establish that predictive responses alone improve every explanation. Tables A.5 and A.6 report the complete block-substitution results. These controls assess retained predictive support; replacement by another model's descriptors can also alter their joint distribution.

Table A.5: Generator-only factorial conditioning control. Differences use the same convention as Table A.3. Response substitutes predictive-response descriptors, Geometry substitutes representation-geometric descriptors, and Joint substitutes both. All seven conditions, including matched conditioning, are evaluated together in a separate paired run.

		GCN donor			GAT donor		
Dataset	Fitting scope	Response	Geometry	Joint	Response	Geometry	Joint
MUTAG	Joint-fit	-0.03 ± 0.02	$+0.39 \pm 0.42$	$+1.14 \pm 1.62$	-0.12 ± 0.23	$+0.66 \pm 0.57$	$+0.58 \pm 0.52$
MUTAG	Held-out	-0.83 ± 1.57	$+2.34 \pm 3.40$	$+0.43 \pm 0.52$	-0.85 ± 1.48	$+2.81 \pm 2.65$	$+1.11 \pm 1.31$
BBBP	Joint-fit	$+0.25 \pm 0.25$	$+4.19 \pm 1.37$	$+3.28 \pm 1.06$	-0.05 ± 0.56	$+2.99 \pm 1.13$	$+2.92 \pm 0.66$
BBBP	Held-out	-0.35 ± 0.50	$+3.14 \pm 1.83$	$+3.17 \pm 1.93$	-0.63 ± 0.92	$+2.75 \pm 1.51$	$+2.87 \pm 2.22$
Benzene	Joint-fit	$+0.92 \pm 0.87$	$+5.57 \pm 3.83$	$+5.14 \pm 4.48$	$+2.20 \pm 1.50$	$+2.68 \pm 2.57$	$+3.85 \pm 2.07$
Benzene	Held-out	$+0.50 \pm 0.96$	$+8.54 \pm 5.03$	$+7.75 \pm 5.20$	$+2.88 \pm 3.46$	$+3.29 \pm 2.30$	$+4.66 \pm 3.55$

Table A.6: After refinement factorial conditioning control. Differences use the same convention as Table A.3. Response substitutes predictive-response descriptors, Geometry substitutes representation-geometric descriptors, and Joint substitutes both. All seven conditions, including matched conditioning, are evaluated together in a separate paired run.

		GCN donor			GAT donor		
Dataset	Fitting scope	Response	Geometry	Joint	Response	Geometry	Joint
MUTAG	Joint-fit	$+0.22 \pm 0.40$	$+0.06 \pm 0.09$	$+1.29 \pm 2.18$	-0.05 ± 0.07	$+0.19 \pm 0.19$	$+0.86 \pm 1.32$
MUTAG	Held-out	-0.24 ± 0.54	$+0.92 \pm 1.37$	$+0.78 \pm 1.15$	-0.16 ± 0.27	$+1.17 \pm 1.36$	$+1.02 \pm 1.11$
BBBP	Joint-fit	$+0.06 \pm 0.12$	$+0.15 \pm 0.18$	$+0.05 \pm 0.07$	$+0.17 \pm 0.14$	$+0.26 \pm 0.11$	$+0.30 \pm 0.03$
BBBP	Held-out	-0.02 ± 0.09	$+0.15 \pm 0.27$	$+0.11 \pm 0.32$	$+0.14 \pm 0.12$	$+0.13 \pm 0.22$	$+0.23 \pm 0.21$
Benzene	Joint-fit	$+0.70 \pm 1.38$	$+1.52 \pm 1.54$	$+0.99 \pm 1.03$	$+0.70 \pm 0.89$	$+0.55 \pm 0.72$	$+1.02 \pm 1.65$
Benzene	Held-out	$+0.80 \pm 0.65$	$+0.94 \pm 1.74$	$+1.13 \pm 1.89$	$+0.31 \pm 0.95$	$+0.99 \pm 0.51$	$+0.40 \pm 0.46$

E.3 Candidate Selection with a Fixed Candidate Pool

This control isolates the contribution of intervention contexts to final candidate selection. Each target's three final route candidates are constructed once and scored by two selectors. STAR uses the original and intervention contexts, whereas the original-input-only selector omits intervention contexts at this final comparison. Candidate construction, search, fallback rules, and all other selection settings are identical. The experiment covers MUTAG, BBBP, and Benzene with GIN under both fitting scopes, giving 1,372 paired test-instance occurrences.

Selected subgraphs are evaluated on the original input and transported to 14 additional perturbation draws per target, using the same draws for both selectors. No explanation is regenerated on these draws. Fixed-view support measures retained target-class confidence, and fixed-view gap measures its absolute difference from the corresponding full-view confidence. The draws use a separate random-seed stream under the local perturbation policy in Appendix B.2, without rejecting duplicates or unchanged graphs. Of 19,208 draws, 5,505 coincide with construction views, 3,231 with selection views, and 4,209 are unchanged inputs; these categories overlap. Independent sampling therefore does not imply distinct graphs. These fixed-subgraph measurements differ from Robust Fidelity, which re-explains perturbed inputs.

Table A.7 shows small, mixed changes. MUTAG is effectively unchanged; Benzene improves under Held-out fitting but declines under Joint-fit fitting, and BBBP changes are small. The control characterizes the final selector's incremental effect on an already constructed pool, rather than the contribution of interventions throughout candidate construction and search.

E.4 Paired Generator-Free and Baseline Controls

These controls compare the complete method with generator-free local inference and longer baseline optimization. Table A.8 covers six datasets, three architectures, and three seeds, using the same frozen checkpoints and 4,750 test-target occurrences across 54 configurations. Aggregation follows Appendix E.1.

Table A.7: Fixed-candidate selection control. Entries are STAR selection minus original-input-only selection, in percentage points of normalized AUC, with paired three-seed standard deviations. Positive Fidelity and fixed-view support differences, and negative fixed-view gap differences, favor STAR selection. Route changes count targets across all three seeds. Fixed-view measurements transport the selected subgraph to new perturbation draws without re-explaining; they are not Robust Fidelity.

Dataset	Fitting scope	Fidelity	Fixed-view support	Fixed-view gap	Route changes
MUTAG	Joint-fit	$+0.000 \pm 0.000$	$+0.000 \pm 0.000$	$+0.000 \pm 0.000$	0/86
MUTAG	Held-out	$+0.000 \pm 0.000$	$+0.000 \pm 0.000$	$+0.000 \pm 0.000$	0/86
BBBP	Joint-fit	$+0.017 \pm 0.053$	$+0.018 \pm 0.061$	-0.042 ± 0.074	4/300
BBBP	Held-out	$+0.001 \pm 0.001$	-0.012 ± 0.016	$+0.006 \pm 0.012$	6/300
Benzene	Joint-fit	-0.156 ± 0.262	-0.092 ± 0.156	$+0.086 \pm 0.157$	4/300
Benzene	Held-out	$+0.171 \pm 0.283$	$+0.154 \pm 0.213$	-0.129 ± 0.177	5/300

Controls. Strict Local-only removes the explanatory prior generator, its response probes, explanatory-prior-guided candidates, and generator-guided proposals. New edges receive constant relevance 0.5, replacing the generated-mask padding retained by ordinary Local-only. Its Joint-fit and Held-out settings coincide because it uses no fitted generator. GNNExplainer receives 50 optimization iterations. GEAR uses the multiobjective adapter in Appendix D.2.2. STAR retains its main-study checkpoints and scores.

Joint-fit exceeds Strict Local-only in 16 of 18 seed-averaged Fidelity AUC comparisons and all 18 Robust Fidelity AUC comparisons; Held-out exceeds it in 15 and 18, respectively. Against GNNExplainer (50 iterations), Joint-fit is higher in all 18 comparisons per metric and Held-out in 17. These complete-procedure comparisons do not isolate learned initialization from the additional search route and queries.

The absolute sufficiency gap and comprehensiveness defined in Appendix D.1 provide complementary assessments. Their AUCs over the same 54 configurations are 0.0945 and 0.2508 for Joint-fit, 0.1038 and 0.2391 for Strict Local-only, 0.1393 and 0.1386 for GEAR, and 0.1360 and 0.1725 for GNNExplainer (50 iterations).

E.5 Generator Initialization and Perturbation-Objective Controls

The initialization control covers MUTAG with GCN, GIN, and GAT and BBBP with GCN, under both fitting scopes, giving 24 complete paired configurations. Learned, uniform, and random initializations all undergo the same refinement with both search routes. Candidate-dependent execution makes them procedure-matched, not exactly query-matched.

The largest mean benefit over uniform initialization occurs on MUTAG–GIN: Fidelity AUC and Robust Fidelity AUC increase by 0.0202 and 0.0161 under Joint-fit and 0.0175 and 0.0122 under Held-out. Fidelity differences on MUTAG–GCN and MUTAG–GAT are slightly negative in both settings, whereas BBBP–GCN improves. Comparisons with random initialization also vary across targets.

Table A.8: Paired controls on all six datasets, averaged equally over three architectures and three seeds. The GNNExplainer control uses 50 optimization iterations. Strict Local-only excludes the generator, and GEAR uses multiobjective training. Second lines give sample standard deviations across three seed-specific backbone averages. Bold indicates the best result, including ties. Underlining indicates the second-best result only when the best is unique. Highlighting uses unrounded values.

Dataset	Metric (↑)	STAR Joint-fit	STAR Held-out	GNNExplainer (50)	Strict Local-only	GEAR
MUTAG	Fidelity AUC	**0.6414** ± 0.0639	<u>0.6404</u> ± 0.0638	0.6077 ± 0.0580	0.6345 ± 0.0597	0.5999 ± 0.0446
MUTAG	Robust Fidelity AUC	**0.6371** ± 0.0619	<u>0.6359</u> ± 0.0613	0.6099 ± 0.0539	0.6314 ± 0.0580	0.5982 ± 0.0421
BBBP	Fidelity AUC	**0.8443** ± 0.0061	<u>0.8434</u> ± 0.0081	0.8223 ± 0.0089	0.8397 ± 0.0068	0.7666 ± 0.0148
BBBP	Robust Fidelity AUC	**0.8435** ± 0.0078	<u>0.8432</u> ± 0.0068	0.8293 ± 0.0076	0.8400 ± 0.0066	0.7672 ± 0.0132
BACE	Fidelity AUC	**0.5547** ± 0.0372	<u>0.5540</u> ± 0.0374	0.5483 ± 0.0319	0.5525 ± 0.0367	0.5408 ± 0.0260
BACE	Robust Fidelity AUC	**0.5535** ± 0.0343	<u>0.5529</u> ± 0.0346	0.5474 ± 0.0290	0.5515 ± 0.0341	0.5404 ± 0.0233
HIV	Fidelity AUC	**0.6955** ± 0.0268	<u>0.6916</u> ± 0.0248	0.6687 ± 0.0141	0.6703 ± 0.0249	0.6304 ± 0.0271
HIV	Robust Fidelity AUC	**0.7079** ± 0.0254	<u>0.7054</u> ± 0.0229	0.6910 ± 0.0166	0.6877 ± 0.0238	0.6539 ± 0.0190
Benzene	Fidelity AUC	**0.7209** ± 0.0093	<u>0.7139</u> ± 0.0139	0.6240 ± 0.0146	0.7009 ± 0.0137	0.6695 ± 0.0670
Benzene	Robust Fidelity AUC	**0.7304** ± 0.0105	<u>0.7260</u> ± 0.0125	0.6494 ± 0.0101	0.7184 ± 0.0116	0.6696 ± 0.0546
PROTEINS	Fidelity AUC	**0.6450** ± 0.0209	<u>0.6442</u> ± 0.0211	0.6303 ± 0.0208	0.6416 ± 0.0222	0.6079 ± 0.0177
PROTEINS	Robust Fidelity AUC	**0.6437** ± 0.0212	<u>0.6429</u> ± 0.0209	0.6279 ± 0.0245	0.6407 ± 0.0221	0.6075 ± 0.0176

The objective control compares standard and perturbation-weight-zero generators on MUTAG, BBBP, and Benzene with GIN, three seeds, and both source-fitting scopes. Generator-only inference isolates fitting effects before local refinement, with source configurations, target GNNs, and evaluation inputs matched. Table A.9 reports the paired AUC differences.

The perturbation term does not consistently improve Generator-only fidelity in this control. Its empirical contribution is therefore distinct from the gains of the complete refinement procedure.

E.6 Prediction Preservation and Explanation-Set Consistency

Robust Fidelity re-explains each perturbed input. Repeated runs, transported subgraphs, and conditioned overlap separately test whether the original explanation retains support and whether selected edges remain consistent.

Paired population. MUTAG, BBBP, BACE, and HIV have complete diagnostics for all six methods in Table A.10: 36 backbone–seed configurations and 2,950 target occurrences per method. Completion, not performance, defines this subset; incomplete Benzene and PROTEINS diagnostics are excluded for every method. Explanations are re-evaluated with the same fitted models, without filling missing diagnostics from historical scores. All records have fourteen valid perturbations, nine retained fractions, and zero evaluation failures; the six-dataset controls separately use archived main-study STAR scores.

Repeated explanations. Each input is explained with seeds s, $s + 7919$, and $s + 15838$, keeping target and generator parameters fixed in evaluation mode and local logits initially zero. Descriptor and local-perturbation randomness follow Appendix B.2. At each fraction, repeat overlap averages Jaccard similarity between the first explanation and the other two. This measures execution variability, not variation across fitted models, and does not separate sampling from numerical effects. Perturbed overlap compares original edge identities with re-explained selections. Undirected edges are grouped consistently; empty–empty overlap is one.

Transported subgraph. Let S_ρ denote the selected original edges and $\mathcal{T}_\xi(S_\rho)$ their surviving identities in G_ξ. Deleted edges disappear and newly inserted edges receive zero relevance. The original predicted class $c(G)$ remains the reference. Fixed-subgraph support is

$$Q_\rho(G) = \frac{1}{|\mathcal{V}_G|} \sum_{\xi \in \mathcal{V}_G} f_\theta(G_\xi[\mathcal{T}_\xi(S_\rho)])_{c(G)} \,. \tag{A.22}$$

Unlike Equation (A.19), this diagnostic uses neither a newly generated explanation nor the perturbed graph's predicted class. Conditioned fixed-subgraph support and conditioned overlap average only views satisfying $D_C(p_\xi \| p_0) \leq 0.05$. All 2,950 targets have finite conditioned scores for every method, each using 5–14 views with identical accepted view sets across methods. Overall, 38,127 of 41,300 views qualify. Of these, 1,246 change the predicted class, indicating that small divergence does not imply an unchanged class. The denominator is the accepted count for each target. These conditional scores are computed within each target before the common AUC and result aggregation. Accepted views are not pooled across targets.

Joint-fit and Held-out have higher average prediction preservation than their respective Generator-only controls, both after re-explanation and after transporting the original subgraph. Their mean differences between original and fixed-subgraph support are 0.0224 and 0.0221, versus 0.0187 and

Table A.9: Generator-only effect of the perturbation-conditioned training term. Entries are standard minus perturbation-weight-zero AUC, reported as mean ± sample standard deviation of three paired seed differences. Positive values favor the standard objective. The compared generators use Joint-fit or Held-out source scopes, but neither receives local refinement here.

Source scope	Dataset	Fidelity AUC difference	Robust Fidelity AUC difference
Joint-fit	MUTAG	$+0.0036 \pm 0.0016$	$+0.0022 \pm 0.0038$
Joint-fit	BBBP	-0.0120 ± 0.0116	-0.0115 ± 0.0070
Joint-fit	Benzene	-0.0273 ± 0.0535	-0.0214 ± 0.0385
Held-out	MUTAG	$+0.0001 \pm 0.0067$	-0.0008 ± 0.0080
Held-out	BBBP	-0.0408 ± 0.0510	-0.0416 ± 0.0539
Held-out	Benzene	-0.0086 ± 0.0186	-0.0071 ± 0.0103

0.0178 for the corresponding Generator-only controls. Higher absolute support therefore coexists with greater degradation. Refinement also reduces repeated-run and perturbed edge overlap relative to Generator-only. GEAR has higher overlap despite lower support. Relative to Strict Local-only, Joint-fit and Held-out improve average support with smaller, nonuniform differences in overlap. The Generator-only objective control in Appendix E.5 concerns a separate three-dataset GIN population; the training term's effect after full refinement remains unisolated here.

Table A.10: Predictive support and explanation-set agreement on the four datasets with complete diagnostic coverage (MUTAG, BBBP, BACE, HIV), using the same 36 backbone–seed configurations for every method. Entries give normalized AUC means and sample standard deviations across three seed-specific dataset–backbone averages. Fixed-subgraph support evaluates the original selected subgraph without re-explanation. Conditioned columns use the prediction-divergence threshold. Higher values indicate greater support or agreement, respectively.

Method	Fidelity	Robust Fidelity	Fixed support	Conditioned fixed support	Repeat overlap	Perturbed overlap	Conditioned overlap
STAR Joint-fit	0.6839 ± 0.0261	0.6855 ± 0.0234	0.6615 ± 0.0224	0.6759 ± 0.0242	0.7441 ± 0.0086	0.5819 ± 0.0035	0.5938 ± 0.0067
STAR Held-out	0.6823 ± 0.0267	0.6843 ± 0.0237	0.6603 ± 0.0232	0.6746 ± 0.0248	0.7348 ± 0.0161	0.5806 ± 0.0041	0.5920 ± 0.0060
Joint-fit Generator-only	0.6525 ± 0.0196	0.6549 ± 0.0169	0.6338 ± 0.0172	0.6472 ± 0.0187	0.9981 ± 0.0002	0.7209 ± 0.0388	0.7384 ± 0.0418
Held-out Generator-only	0.6449 ± 0.0268	0.6478 ± 0.0234	0.6271 ± 0.0232	0.6404 ± 0.0250	0.9980 ± 0.0006	0.7205 ± 0.0239	0.7374 ± 0.0241
Strict Local-only	0.6743 ± 0.0267	0.6776 ± 0.0243	0.6535 ± 0.0233	0.6670 ± 0.0250	0.7200 ± 0.0234	0.5742 ± 0.0103	0.5856 ± 0.0114
GEAR	0.6344 ± 0.0196	0.6399 ± 0.0148	0.6189 ± 0.0174	0.6306 ± 0.0182	0.9999 ± 0.0002	0.7658 ± 0.0081	0.7835 ± 0.0091

Table A.11: Cross-domain explanation on Cora. Each entry is the mean and sample standard deviation across three seeds, with 100 paired test nodes per backbone and seed. Bold indicates the best mean, including ties. Underlining indicates the second-best mean only when the best is unique.

Backbone / seeds	Metric (↑)	GNNExplainer	PGExplainer	GraphMask	Graph Curvatures (PGExplainer)	STAR Generator-only	STAR
GCN / 0,1,2	Fidelity AUC	<u>0.8666</u> ± 0.0265	0.8544 ± 0.0283	0.7646 ± 0.0405	0.8544 ± 0.0283	0.7808 ± 0.0519	**0.8970** ± 0.0252
GCN / 0,1,2	Robust Fidelity AUC	<u>0.8555</u> ± 0.0286	0.8440 ± 0.0309	0.7580 ± 0.0367	0.8440 ± 0.0309	0.7729 ± 0.0472	**0.8899** ± 0.0266
GIN / 0,1,2	Fidelity AUC	<u>0.8789</u> ± 0.0431	0.5230 ± 0.0531	0.5032 ± 0.0748	0.5230 ± 0.0531	0.5183 ± 0.2135	**0.8892** ± 0.0461
GIN / 0,1,2	Robust Fidelity AUC	<u>0.8695</u> ± 0.0451	0.5155 ± 0.0524	0.5019 ± 0.0442	0.5155 ± 0.0524	0.5121 ± 0.2060	**0.8811** ± 0.0478
GAT / 0,1,2	Fidelity AUC	<u>0.7931</u> ± 0.1658	0.7923 ± 0.1654	0.6903 ± 0.1286	0.7923 ± 0.1654	0.6588 ± 0.0803	**0.8243** ± 0.1718
GAT / 0,1,2	Robust Fidelity AUC	<u>0.7803</u> ± 0.1679	0.7793 ± 0.1676	0.6826 ± 0.1287	0.7793 ± 0.1676	0.6496 ± 0.0808	**0.8171** ± 0.1744

Table A.12: Checkpoint-quality sensitivity. The original 54 settings and the 49 settings after excluding single-class predictors use the same paired method populations. Remaining seeds are averaged within each dataset–backbone pair before equal weighting of the 18 pairs. Bold marks the best mean and underlining the second-best when the best is unique.

	Original 54		Retained 49, balanced	
Method	Fidelity AUC ↑	Robust Fidelity AUC ↑	Fidelity AUC ↑	Robust Fidelity AUC ↑
GNNExplainer	0.6277	0.6377	0.6315	0.6417
PGExplainer	0.6041	0.6090	0.6053	0.6104
GraphMask	0.6150	0.6184	0.6178	0.6214
GEAR	0.6358	0.6395	0.6398	0.6435
IDEA	0.5927	0.6003	0.5935	0.6013
Graph Curvatures	0.6089	0.6156	0.6120	0.6188
STAR (Held-out)	<u>0.6812</u>	<u>0.6844</u>	<u>0.6874</u>	<u>0.6907</u>
STAR (Joint-fit)	**0.6836**	**0.6860**	**0.6899**	**0.6925**

Table A.13: Joint-fit component analysis, averaged over GCN, GIN, and GAT and then over three seeds. Values show means and sample standard deviations across seed-specific backbone averages. Bold marks the best mean within each setting, including ties. Underlining marks the second-best mean when the best is unique.

Dataset	Metric ↑	Generator-only	Local-only	w/o Mask regularization	w/o Prediction alignment	STAR
MUTAG	Fidelity AUC	0.6265 ± 0.0612	0.6342 ± 0.0596	<u>0.6372</u> ± 0.0607	0.6347 ± 0.0597	**0.6414** ± 0.0639
	Robust Fidelity AUC	0.6201 ± 0.0561	0.6312 ± 0.0579	<u>0.6334</u> ± 0.0590	0.6316 ± 0.0580	**0.6371** ± 0.0619
BBBP	Fidelity AUC	0.7777 ± 0.0109	0.8402 ± 0.0059	<u>0.8429</u> ± 0.0066	0.8412 ± 0.0069	**0.8443** ± 0.0061
	Robust Fidelity AUC	0.7790 ± 0.0077	0.8405 ± 0.0068	<u>0.8425</u> ± 0.0066	0.8409 ± 0.0067	**0.8435** ± 0.0078
Benzene	Fidelity AUC	0.6893 ± 0.0282	0.7017 ± 0.0125	<u>0.7077</u> ± 0.0121	0.7042 ± 0.0161	**0.7209** ± 0.0093
	Robust Fidelity AUC	0.6826 ± 0.0298	0.7186 ± 0.0120	<u>0.7241</u> ± 0.0116	0.7191 ± 0.0128	**0.7304** ± 0.0105

Table A.14: Held-out component analysis, averaged over GCN, GIN, and GAT and then over three seeds. Values show means and sample standard deviations across seed-specific backbone averages. Bold marks the best mean within each setting, including ties. Underlining marks the second-best mean when the best is unique.

Dataset	Metric ↑	Generator-only	Local-only	w/o Mask regularization	w/o Prediction alignment	STAR
MUTAG	Fidelity AUC	0.6240 ± 0.0614	0.6345 ± 0.0598	<u>0.6366</u> ± 0.0608	0.6350 ± 0.0603	**0.6404** ± 0.0638
	Robust Fidelity AUC	0.6177 ± 0.0567	0.6312 ± 0.0579	<u>0.6329</u> ± 0.0588	0.6319 ± 0.0584	**0.6359** ± 0.0613
BBBP	Fidelity AUC	0.7499 ± 0.0236	0.8404 ± 0.0066	**0.8439** ± 0.0075	0.8415 ± 0.0070	<u>0.8434</u> ± 0.0081
	Robust Fidelity AUC	0.7529 ± 0.0259	0.8405 ± 0.0067	<u>0.8425</u> ± 0.0065	0.8413 ± 0.0062	**0.8432** ± 0.0068
Benzene	Fidelity AUC	0.6875 ± 0.0125	0.7036 ± 0.0134	<u>0.7080</u> ± 0.0133	0.7038 ± 0.0149	**0.7139** ± 0.0139
	Robust Fidelity AUC	0.6847 ± 0.0036	0.7191 ± 0.0119	<u>0.7242</u> ± 0.0116	0.7204 ± 0.0122	**0.7260** ± 0.0125

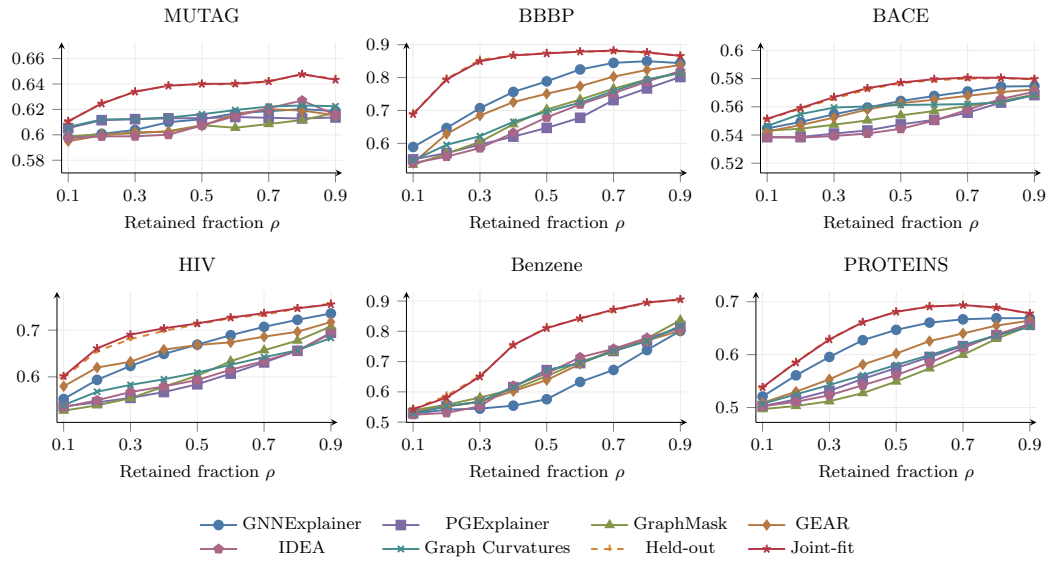

Figure A.1: Fidelity across retained-edge fractions on GCN.

E.7 Cross-Domain Generalization to Cora

Each seed's Joint-fit generator, fitted on molecular and protein graph-classification data, is applied unchanged to Cora node explanations (Yang et al., 2016). Cora is absent from generator fitting and validation; target GNNs are trained on its node-classification training split and then frozen, and parameterized baselines may fit on that split. Table A.11 uses 100 common, correctly predicted test nodes per backbone–seed configuration, under the common comparison protocol. Generator-only removes refinement while retaining the same generator and test nodes.

Table A.15: Complete GCN results with means and sample standard deviations across three seeds. Bold indicates the best mean, including ties. Underlining indicates the second-best mean only when the best is unique. Highlighting uses unrounded values.

Dataset	Metric (↑)	GNNExplainer	PGExplainer	GraphMask	GEAR	IDEA	Graph Curvatures	Held-out	Joint-fit
MUTAG	Fidelity AUC	0.6113 ± 0.0568	0.6124 ± 0.0595	0.6057 ± 0.0526	0.6090 ± 0.0511	0.6093 ± 0.0521	0.6167 ± 0.0557	0.6367 ± 0.0634	**0.6368** ± 0.0636
	Robust Fidelity AUC	0.6075 ± 0.0509	0.6026 ± 0.0510	0.5997 ± 0.0464	0.6026 ± 0.0448	0.6019 ± 0.0460	0.6064 ± 0.0475	**0.6301** ± 0.0591	0.6300 ± 0.0590
	Fidelity Best	0.6410 ± 0.0667	0.6380 ± 0.0639	0.6355 ± 0.0650	0.6412 ± 0.0652	0.6436 ± 0.0662	0.6353 ± 0.0672	0.6526 ± 0.0703	**0.6528** ± 0.0707
	Robust Fidelity Best	0.6301 ± 0.0587	0.6245 ± 0.0544	0.6250 ± 0.0567	0.6292 ± 0.0554	0.6331 ± 0.0585	0.6241 ± 0.0564	0.6421 ± 0.0635	**0.6422** ± 0.0637
BBBP	Fidelity AUC	0.7667 ± 0.0123	0.6609 ± 0.0034	0.6878 ± 0.0399	0.7348 ± 0.0179	0.6743 ± 0.0496	0.6920 ± 0.0280	**0.8505** ± 0.0069	0.8500 ± 0.0051
	Robust Fidelity AUC	0.7707 ± 0.0040	0.6656 ± 0.0035	0.6687 ± 0.0231	0.7292 ± 0.0091	0.6720 ± 0.0448	0.6917 ± 0.0215	0.8412 ± 0.0078	**0.8414** ± 0.0077
	Fidelity Best	0.9041 ± 0.0105	0.8592 ± 0.0108	0.8739 ± 0.0109	0.8928 ± 0.0072	0.8838 ± 0.0076	0.8684 ± 0.0190	**0.9139** ± 0.0078	0.9135 ± 0.0047
	Robust Fidelity Best	0.8649 ± 0.0093	0.8204 ± 0.0079	0.8173 ± 0.0029	0.8523 ± 0.0082	0.8389 ± 0.0041	0.8298 ± 0.0185	0.8857 ± 0.0087	**0.8862** ± 0.0082
BACE	Fidelity AUC	0.5625 ± 0.0517	0.5492 ± 0.0361	0.5542 ± 0.0412	0.5601 ± 0.0483	0.5489 ± 0.0359	0.5600 ± 0.0480	0.5722 ± 0.0657	**0.5728** ± 0.0663
	Robust Fidelity AUC	0.5611 ± 0.0509	0.5483 ± 0.0356	0.5545 ± 0.0423	0.5584 ± 0.0472	0.5486 ± 0.0358	0.5580 ± 0.0464	0.5703 ± 0.0641	**0.5705** ± 0.0641
	Fidelity Best	0.5820 ± 0.0800	0.5745 ± 0.0686	0.5732 ± 0.0666	0.5776 ± 0.0733	0.5779 ± 0.0739	0.5732 ± 0.0663	0.5872 ± 0.0883	**0.5874** ± 0.0885
	Robust Fidelity Best	0.5785 ± 0.0760	0.5710 ± 0.0647	0.5727 ± 0.0672	0.5744 ± 0.0697	0.5742 ± 0.0694	0.5709 ± 0.0641	0.5832 ± 0.0832	**0.5832** ± 0.0831
HIV	Fidelity AUC	0.6622 ± 0.0185	0.5952 ± 0.0385	0.6076 ± 0.0186	0.6604 ± 0.0884	0.6011 ± 0.0508	0.6114 ± 0.0171	0.7035 ± 0.0171	**0.7072** ± 0.0063
	Robust Fidelity AUC	0.6773 ± 0.0264	0.6216 ± 0.0600	0.6380 ± 0.0300	0.6748 ± 0.0871	0.6285 ± 0.0663	0.6352 ± 0.0442	0.7101 ± 0.0371	**0.7103** ± 0.0317
	Fidelity Best	0.8023 ± 0.0258	0.7551 ± 0.0514	0.7702 ± 0.0373	0.7904 ± 0.0694	0.7514 ± 0.0531	0.7449 ± 0.0636	**0.8042** ± 0.0235	0.8037 ± 0.0189
	Robust Fidelity Best	0.7843 ± 0.0320	0.7534 ± 0.0585	0.7643 ± 0.0375	0.7751 ± 0.0655	0.7540 ± 0.0606	0.7468 ± 0.0652	**0.7906** ± 0.0350	0.7886 ± 0.0342
Benzene	Fidelity AUC	0.6155 ± 0.0189	0.6599 ± 0.0123	0.6619 ± 0.0086	0.6538 ± 0.0140	0.6582 ± 0.0168	0.6599 ± 0.0123	**0.7670** ± 0.0204	0.7663 ± 0.0158
	Robust Fidelity AUC	0.5983 ± 0.0220	0.6197 ± 0.0133	0.6018 ± 0.0146	0.6113 ± 0.0166	0.6146 ± 0.0082	0.6198 ± 0.0133	**0.7400** ± 0.0204	0.7399 ± 0.0182
	Fidelity Best	0.9427 ± 0.0067	0.9102 ± 0.0295	0.9400 ± 0.0335	0.9143 ± 0.0477	0.9344 ± 0.0226	0.9102 ± 0.0295	0.9644 ± 0.0112	**0.9649** ± 0.0095
	Robust Fidelity Best	0.8626 ± 0.0096	0.8500 ± 0.0304	0.8494 ± 0.0202	0.8501 ± 0.0503	0.8653 ± 0.0240	0.8500 ± 0.0304	**0.9051** ± 0.0014	0.9041 ± 0.0013
PROTEINS	Fidelity AUC	0.6277 ± 0.0237	0.5755 ± 0.0130	0.5588 ± 0.0259	0.5967 ± 0.0179	0.5690 ± 0.0233	0.5803 ± 0.0089	**0.6549** ± 0.0229	0.6547 ± 0.0223
	Robust Fidelity AUC	0.6271 ± 0.0243	0.5757 ± 0.0121	0.5999 ± 0.0216	0.5965 ± 0.0175	0.5694 ± 0.0201	0.5805 ± 0.0079	0.6539 ± 0.0212	**0.6539** ± 0.0214
	Fidelity Best	0.7069 ± 0.0345	0.6886 ± 0.0216	0.6908 ± 0.0377	0.7021 ± 0.0383	0.6909 ± 0.0285	0.6822 ± 0.0266	0.7120 ± 0.0294	**0.7126** ± 0.0296
	Robust Fidelity Best	0.6974 ± 0.0352	0.6813 ± 0.0220	0.6797 ± 0.0319	0.6923 ± 0.0359	0.6834 ± 0.0276	0.6757 ± 0.0258	0.7053 ± 0.0278	**0.7055** ± 0.0278
Avg. rank across metrics		3.67	6.58	6.25	4.67	5.71	6.12	1.62	**1.38**

STAR exceeds the displayed external baselines in all nine backbone–seed configurations and Generator-only for every backbone. Query-relative descriptors become active (Appendix B.1), and GAT shows substantial seed variation. These results support the complete procedure. A paired Strict Local-only comparison is needed to isolate the source-fitted generator's contribution.

PGExplainer and Graph Curvatures produce nearly identical paired metrics on all 900 Cora targets, with maximum absolute differences of 6.1×10^{-8} and 1.7×10^{-8} on the two AUCs.

E.8 Sensitivity to Frozen Target Quality

Excluding the five single-class predictors in Appendix D.2.1 leaves 49 of 54 checkpoint settings and 4,250 target occurrences per method. The class-count criterion applies to all eight methods, independently of explanation performance; the original main table is unchanged. GCN, GIN, and GAT retain 16, 15, and 18 settings. BACE–GCN and BACE–GIN each retain one seed, and HIV–GIN retains two.

Table A.16: Complete GIN results with means and sample standard deviations across three seeds. Bold indicates the best mean, including ties. Underlining indicates the second-best mean only when the best is unique. Highlighting uses unrounded values.

Dataset	Metric (↑)	GNNExplainer	PGExplainer	GraphMask	GEAR	IDEA	Graph Curvatures	Held-out	Joint-fit
MUTAG	Fidelity AUC	0.5275 ± 0.0307	0.5205 ± 0.0224	0.5648 ± 0.0486	0.5290 ± 0.0178	0.5287 ± 0.0507	0.5181 ± 0.0140	0.5958 ± 0.0705	**0.5984** ± 0.0738
	Robust Fidelity AUC	0.5424 ± 0.0326	0.5272 ± 0.0295	0.5526 ± 0.0392	0.5386 ± 0.0220	0.5359 ± 0.0345	0.5272 ± 0.0255	0.5961 ± 0.0717	**0.6000** ± 0.0764
	Fidelity Best	0.6207 ± 0.0896	0.6308 ± 0.1398	0.6666 ± 0.1405	0.6243 ± 0.0889	0.5986 ± 0.0863	0.6185 ± 0.1094	0.6858 ± 0.1396	**0.6864** ± 0.1398
	Robust Fidelity Best	0.6171 ± 0.0899	0.6108 ± 0.1136	0.6315 ± 0.1018	0.6172 ± 0.0854	0.5929 ± 0.0703	0.6058 ± 0.0965	0.6692 ± 0.1258	**0.6704** ± 0.1273
BBBP	Fidelity AUC	0.7466 ± 0.0223	0.6688 ± 0.0386	0.7040 ± 0.0376	0.7873 ± 0.0212	0.5941 ± 0.0081	0.6689 ± 0.0147	0.8176 ± 0.0242	**0.8188** ± 0.0221
	Robust Fidelity AUC	0.7639 ± 0.0362	0.6804 ± 0.0555	0.7189 ± 0.0553	0.7944 ± 0.0303	0.6111 ± 0.0400	0.6861 ± 0.0419	0.8284 ± 0.0373	**0.8296** ± 0.0362
	Fidelity Best	0.8943 ± 0.0316	0.8596 ± 0.0474	0.8731 ± 0.0451	0.8999 ± 0.0274	0.8058 ± 0.0356	0.8476 ± 0.0380	**0.9099** ± 0.0252	0.9086 ± 0.0253
	Robust Fidelity Best	0.8740 ± 0.0420	0.8361 ± 0.0606	0.8418 ± 0.0567	0.8795 ± 0.0406	0.7748 ± 0.0468	0.8259 ± 0.0527	0.8977 ± 0.0342	**0.8977** ± 0.0340
BACE	Fidelity AUC	0.5095 ± 0.0042	0.5076 ± 0.0028	0.5115 ± 0.0071	0.5118 ± 0.0076	0.5081 ± 0.0034	0.5113 ± 0.0066	0.5112 ± 0.0062	**0.5118** ± 0.0072
	Robust Fidelity AUC	0.5119 ± 0.0077	0.5082 ± 0.0029	0.5121 ± 0.0082	0.5144 ± 0.0118	0.5089 ± 0.0036	0.5131 ± 0.0095	0.5137 ± 0.0102	**0.5145** ± 0.0115
	Fidelity Best	0.5129 ± 0.0068	0.5121 ± 0.0059	0.5134 ± 0.0079	0.5138 ± 0.0082	0.5103 ± 0.0043	0.5136 ± 0.0079	**0.5142** ± 0.0086	0.5142 ± 0.0086
	Robust Fidelity Best	0.5154 ± 0.0107	0.5139 ± 0.0088	0.5152 ± 0.0107	0.5162 ± 0.0120	0.5115 ± 0.0051	0.5155 ± 0.0112	**0.5168** ± 0.0127	0.5167 ± 0.0126
HIV	Fidelity AUC	0.6102 ± 0.0298	0.5800 ± 0.0072	0.6013 ± 0.0222	0.6191 ± 0.0561	0.5644 ± 0.0239	0.5919 ± 0.0278	0.6507 ± 0.0847	**0.6542** ± 0.0865
	Robust Fidelity AUC	0.6123 ± 0.0391	0.5795 ± 0.0230	0.6004 ± 0.0487	0.6113 ± 0.0737	0.5669 ± 0.0089	0.5895 ± 0.0139	0.6491 ± 0.0943	**0.6529** ± 0.0960
	Fidelity Best	0.7433 ± 0.1444	0.7365 ± 0.1390	0.7326 ± 0.1355	0.7405 ± 0.1422	0.7057 ± 0.1211	0.7353 ± 0.1378	**0.7535** ± 0.1537	0.7528 ± 0.1532
	Robust Fidelity Best	0.7091 ± 0.1152	0.7055 ± 0.1116	0.6998 ± 0.1092	0.7106 ± 0.1197	0.6800 ± 0.0943	0.7054 ± 0.1115	**0.7278** ± 0.1336	0.7274 ± 0.1333
Benzene	Fidelity AUC	0.5220 ± 0.0116	0.6188 ± 0.0770	0.6290 ± 0.0178	**0.6485** ± 0.1194	0.5366 ± 0.0306	0.5864 ± 0.0210	0.5998 ± 0.0250	0.6223 ± 0.0143
	Robust Fidelity AUC	0.6263 ± 0.0194	0.6828 ± 0.0729	0.6828 ± 0.0099	**0.7109** ± 0.0960	0.6336 ± 0.0172	0.6656 ± 0.0426	0.6771 ± 0.0053	0.6916 ± 0.0136
	Fidelity Best	0.7008 ± 0.0037	0.8549 ± 0.0751	0.8469 ± 0.0704	0.8387 ± 0.1840	0.6925 ± 0.0807	0.8254 ± 0.0218	0.9050 ± 0.0106	**0.9124** ± 0.0173
	Robust Fidelity Best	0.7658 ± 0.0194	0.8382 ± 0.0634	0.8465 ± 0.0161	0.8414 ± 0.1514	0.7388 ± 0.0465	0.8206 ± 0.0296	0.8885 ± 0.0056	**0.8962** ± 0.0025
PROTEINS	Fidelity AUC	0.5745 ± 0.0254	0.5607 ± 0.0130	0.5811 ± 0.0177	0.6028 ± 0.0307	0.5709 ± 0.0356	0.5439 ± 0.0096	0.6090 ± 0.0299	**0.6108** ± 0.0296
	Robust Fidelity AUC	0.5727 ± 0.0263	0.5581 ± 0.0137	0.5648 ± 0.0233	0.6018 ± 0.0302	0.5698 ± 0.0337	0.5435 ± 0.0097	0.6070 ± 0.0293	**0.6091** ± 0.0295
	Fidelity Best	0.6601 ± 0.0349	0.6553 ± 0.0304	0.6616 ± 0.0311	0.6742 ± 0.0381	0.6560 ± 0.0454	0.6347 ± 0.0264	0.6915 ± 0.0362	**0.6919** ± 0.0362
	Robust Fidelity Best	0.6531 ± 0.0356	0.6486 ± 0.0308	0.6471 ± 0.0346	0.6681 ± 0.0376	0.6509 ± 0.0435	0.6305 ± 0.0262	0.6864 ± 0.0362	**0.6866** ± 0.0361
Avg. rank across metrics		5.08	6.08	4.54	3.21	7.21	6.33	2.21	**1.33**

Table A.12 averages remaining seeds within dataset–backbone pairs, then weights all 18 pairs equally, preserving each dataset's weight of one sixth. Joint-fit and Held-out remain numerically ahead overall, exceeding external baselines in 16 of 18 pairs on both AUCs. Joint-fit's count falls from 17 because of BACE–GIN. Singleton groups have no estimable seed standard deviation.

E.9 Component Variability and Additional Findings

Tables A.13 and A.14 report component results on MUTAG, BBBP, and Benzene across GCN, GIN, and GAT. Variants follow Section 4.3. Loss variants change only the designated objective weight during generator fitting; architecture, source scope, seed, and target GNNs remain fixed. STAR uses its main-study scores. Per-backbone and seed-specific results are retained in the supplementary result files.

The backbone-averaged pattern does not hold for every individual backbone. On Benzene–GIN, Generator-only has a higher Fidelity AUC than STAR under Joint-fit (0.7213 versus 0.6223) and

Table A.17: Complete GAT results with means and sample standard deviations across three seeds. Bold indicates the best mean, including ties. Underlining indicates the second-best mean only when the best is unique. Highlighting uses unrounded values.

Dataset	Metric (↑)	GNNExplainer	PGExplainer	GraphMask	GEAR	IDEA	Graph Curvatures	Held-out	Joint-fit
MUTAG	Fidelity AUC	0.6501 ± 0.0992	0.6314 ± 0.0804	0.6409 ± 0.0913	0.6618 ± 0.1056	0.6478 ± 0.0949	0.6564 ± 0.1019	0.6888 ± 0.1235	**0.6889** ± 0.1234
	Robust Fidelity AUC	0.6509 ± 0.0987	0.6251 ± 0.0772	0.6375 ± 0.0886	0.6533 ± 0.0998	0.6405 ± 0.0899	0.6461 ± 0.0946	**0.6815** ± 0.1184	0.6813 ± 0.1183
	Fidelity Best	0.7034 ± 0.1349	0.6959 ± 0.1304	0.6979 ± 0.1317	0.7079 ± 0.1397	0.7069 ± 0.1388	0.6870 ± 0.1245	0.7128 ± 0.1408	**0.7132** ± 0.1412
	Robust Fidelity Best	0.6943 ± 0.1302	0.6849 ± 0.1230	0.6895 ± 0.1276	0.6955 ± 0.1306	0.6963 ± 0.1319	0.6785 ± 0.1192	0.7019 ± 0.1334	**0.7019** ± 0.1335
BBBP	Fidelity AUC	0.8015 ± 0.0197	0.7358 ± 0.0223	0.7397 ± 0.0107	0.7777 ± 0.0217	0.7574 ± 0.0442	0.7425 ± 0.0090	0.8621 ± 0.0350	**0.8641** ± 0.0282
	Robust Fidelity AUC	0.8026 ± 0.0126	0.7326 ± 0.0275	0.7443 ± 0.0104	0.7781 ± 0.0159	0.7545 ± 0.0399	0.7446 ± 0.0029	**0.8600** ± 0.0267	0.8596 ± 0.0240
	Fidelity Best	0.9247 ± 0.0313	0.8854 ± 0.0323	0.8915 ± 0.0260	0.9115 ± 0.0310	0.9028 ± 0.0374	0.9047 ± 0.0262	0.9349 ± 0.0279	**0.9358** ± 0.0280
	Robust Fidelity Best	0.8860 ± 0.0260	0.8534 ± 0.0328	0.8515 ± 0.0289	0.8777 ± 0.0246	0.8659 ± 0.0285	0.8653 ± 0.0211	**0.9094** ± 0.0246	0.9093 ± 0.0243
BACE	Fidelity AUC	0.5508 ± 0.0454	0.5236 ± 0.0385	0.5267 ± 0.0525	0.5504 ± 0.0490	0.5150 ± 0.0511	0.5311 ± 0.0505	0.5784 ± 0.0605	**0.5793** ± 0.0608
	Robust Fidelity AUC	0.5449 ± 0.0451	0.5191 ± 0.0398	0.5258 ± 0.0487	0.5484 ± 0.0451	0.5131 ± 0.0483	0.5295 ± 0.0483	0.5748 ± 0.0582	**0.5755** ± 0.0583
	Fidelity Best	0.6170 ± 0.0972	0.5930 ± 0.0730	0.6023 ± 0.0825	0.6130 ± 0.0935	0.5909 ± 0.0709	0.6052 ± 0.0843	0.6284 ± 0.1084	**0.6289** ± 0.1086
	Robust Fidelity Best	0.6008 ± 0.0806	0.5805 ± 0.0619	0.5891 ± 0.0698	0.5995 ± 0.0797	0.5802 ± 0.0623	0.5921 ± 0.0722	0.6177 ± 0.0968	**0.6177** ± 0.0967
HIV	Fidelity AUC	0.6531 ± 0.0358	0.6128 ± 0.0769	0.6345 ± 0.0255	0.6117 ± 0.0532	0.6215 ± 0.0456	0.6123 ± 0.0688	0.7206 ± 0.0200	**0.7250** ± 0.0159
	Robust Fidelity AUC	0.7063 ± 0.0263	0.6748 ± 0.0362	0.6857 ± 0.0379	0.6754 ± 0.0521	0.6753 ± 0.0442	0.6837 ± 0.0236	0.7570 ± 0.0247	**0.7604** ± 0.0284
	Fidelity Best	0.8134 ± 0.0170	0.7618 ± 0.0470	0.7994 ± 0.0367	0.7867 ± 0.0161	0.7783 ± 0.0030	0.7619 ± 0.0355	0.8419 ± 0.0246	**0.8428** ± 0.0243
	Robust Fidelity Best	0.7998 ± 0.0579	0.7712 ± 0.0427	0.7823 ± 0.0648	0.7840 ± 0.0497	0.7730 ± 0.0496	0.7751 ± 0.0333	**0.8292** ± 0.0485	0.8289 ± 0.0495
Benzene	Fidelity AUC	0.6584 ± 0.0295	0.6873 ± 0.0783	0.6363 ± 0.0409	0.7063 ± 0.0943	0.6053 ± 0.0149	0.6966 ± 0.0403	**0.7749** ± 0.0378	0.7742 ± 0.0341
	Robust Fidelity AUC	0.6560 ± 0.0214	0.6673 ± 0.0651	0.6192 ± 0.0274	0.6867 ± 0.0858	0.6018 ± 0.0167	0.6807 ± 0.0297	**0.7610** ± 0.0306	0.7597 ± 0.0308
	Fidelity Best	0.8942 ± 0.0115	0.8774 ± 0.0355	0.8792 ± 0.0102	0.8901 ± 0.0564	0.8278 ± 0.0039	0.8966 ± 0.0083	**0.9202** ± 0.0115	0.9170 ± 0.0091
	Robust Fidelity Best	0.8373 ± 0.0102	0.8374 ± 0.0225	0.8078 ± 0.0145	0.8474 ± 0.0433	0.7784 ± 0.0114	0.8530 ± 0.0155	0.8768 ± 0.0133	**0.8771** ± 0.0137
PROTEINS	Fidelity AUC	0.6486 ± 0.0148	0.5729 ± 0.0290	0.6248 ± 0.0113	0.6242 ± 0.0118	0.5582 ± 0.0101	0.5806 ± 0.0168	0.6687 ± 0.0222	**0.6694** ± 0.0218
	Robust Fidelity AUC	0.6459 ± 0.0220	0.5736 ± 0.0301	0.6238 ± 0.0152	0.6242 ± 0.0134	0.5587 ± 0.0134	0.5801 ± 0.0190	0.6679 ± 0.0219	**0.6681** ± 0.0219
	Fidelity Best	**0.7537** ± 0.0212	0.7141 ± 0.0373	0.7358 ± 0.0338	0.7475 ± 0.0284	0.7093 ± 0.0287	0.7170 ± 0.0319	0.7495 ± 0.0279	0.7515 ± 0.0279
	Robust Fidelity Best	0.7390 ± 0.0323	0.7030 ± 0.0367	0.7210 ± 0.0315	0.7350 ± 0.0282	0.6990 ± 0.0298	0.7055 ± 0.0326	0.7396 ± 0.0284	**0.7399** ± 0.0282
Avg. rank across metrics		3.67	7.04	5.92	4.12	6.71	5.46	1.75	**1.33**

Held-out (0.7054 versus 0.5998), with the same ordering for Robust Fidelity AUC. Removing the perturbation weight raises Joint-fit's backbone-averaged Benzene Fidelity AUC from 0.7209 to 0.7245 and Robust Fidelity AUC from 0.7304 to 0.7332, while reducing both averages on BBBP.

The selector's retained, complementary, and perturbed terms can favor different candidates. Its score-control properties apply to the composite criterion, not each reported fidelity measure (Equation (13); Appendix B.7).

E.10 Complete Scores and Seed Variability

Tables A.15–A.17 report complete AUC and Best results with seed variability. The standard deviations contextualize small Joint-fit–Held-out differences and describe repeated fits on these benchmarks, not uncertainty over new datasets.

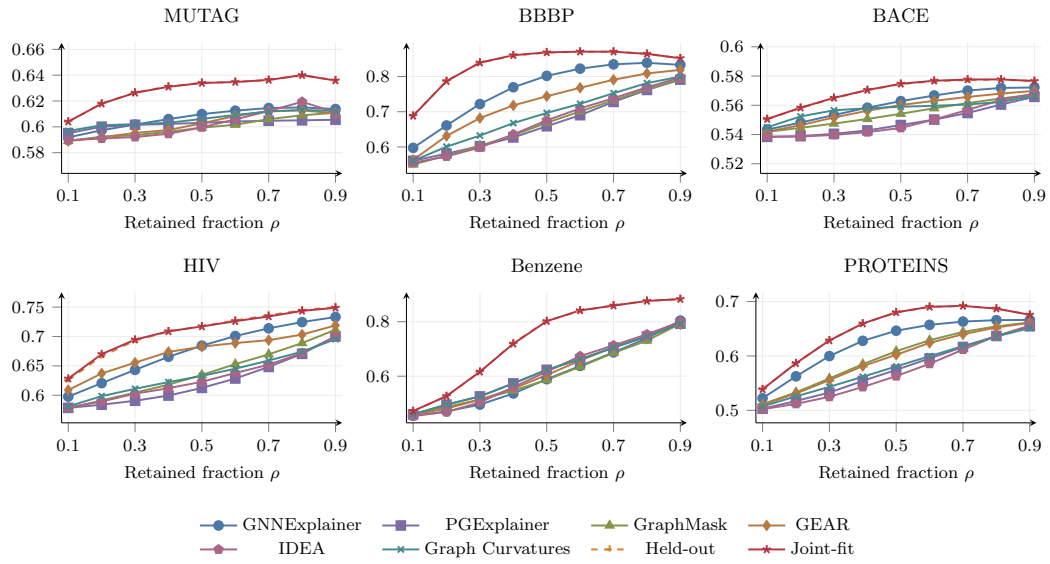

Figure A.2: Robust Fidelity across retained-edge fractions on GCN.

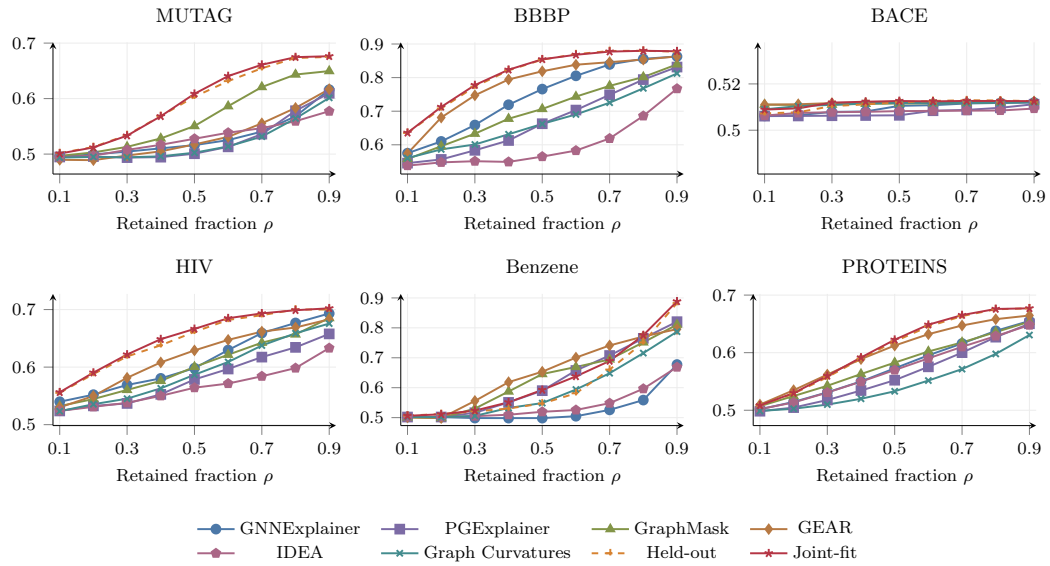

Figure A.3: Fidelity across retained-edge fractions on GIN.

E.11 Fidelity across Retained Fractions

Figures A.1–A.6 plot fidelity over nine retained-edge fractions ρ, using the main-table aggregation. Each seed-specific curve integrates to its main-table AUC. Joint-fit and Held-out ordering can change with explanation size. Colors and markers identify methods consistently, and vertical scales differ by dataset.

E.12 Parameter Sensitivity

Figure A.7 examines local update count T (Equation (11)), candidate interpolation scale w (Equation (A.7)), and search width W (Equation (13)). Each varies over ten settings with other parameters and the generator checkpoint fixed. Sweeps use up to 16 correctly predicted test targets per configuration under the common aggregation protocol. They characterize sensitivity, without test-based parameter selection.

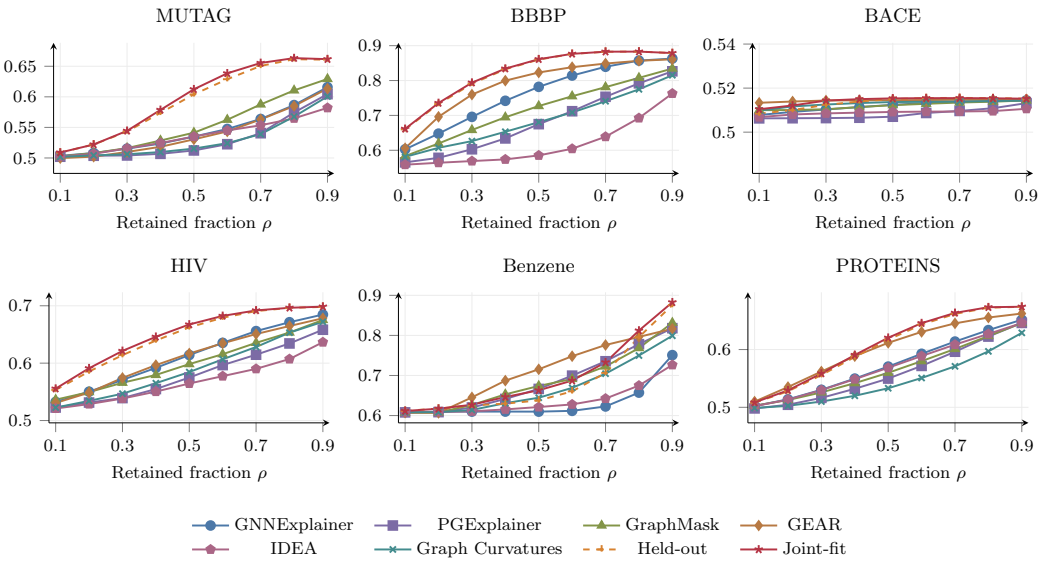

Figure A.4: Robust Fidelity across retained-edge fractions on GIN.

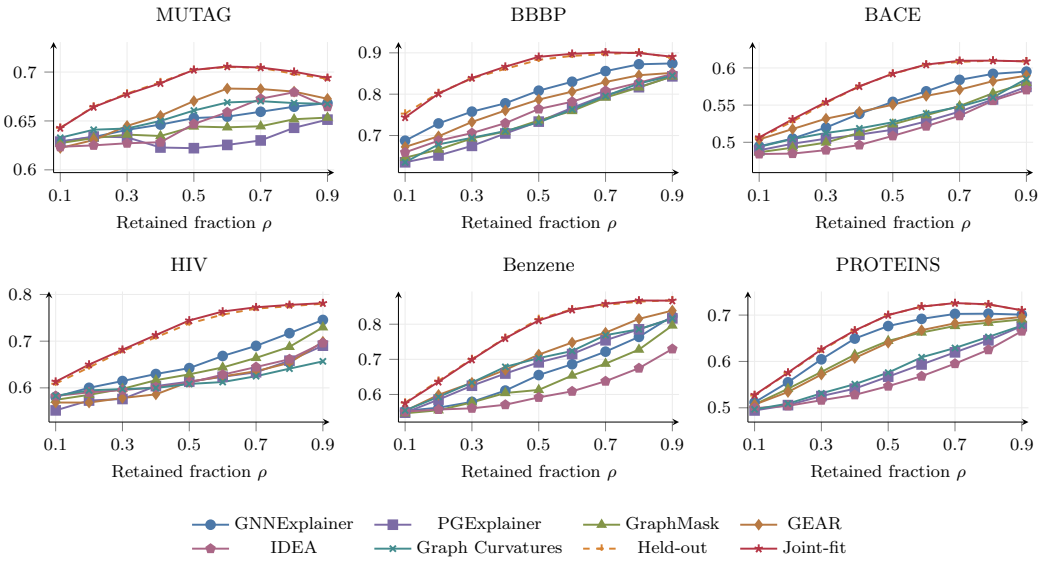

Figure A.5: Fidelity across retained-edge fractions on GAT.

Refinement has nonmonotone effects. Positive-step Fidelity AUC ranges from 0.6846 to 0.6945, below the zero-step generated-mask branch's 0.7198. This trade-off is consistent with the distinction between the composite local objective and retained-confidence AUC. The effect of w depends on candidate selection, not a fixed output mixture. Increasing W expands the exchanges assessed, with AUC ranging from 0.6862 to 0.7004 but no monotone improvement. Only completed GIN sweeps are reported, using a smaller target subset than the main tables.

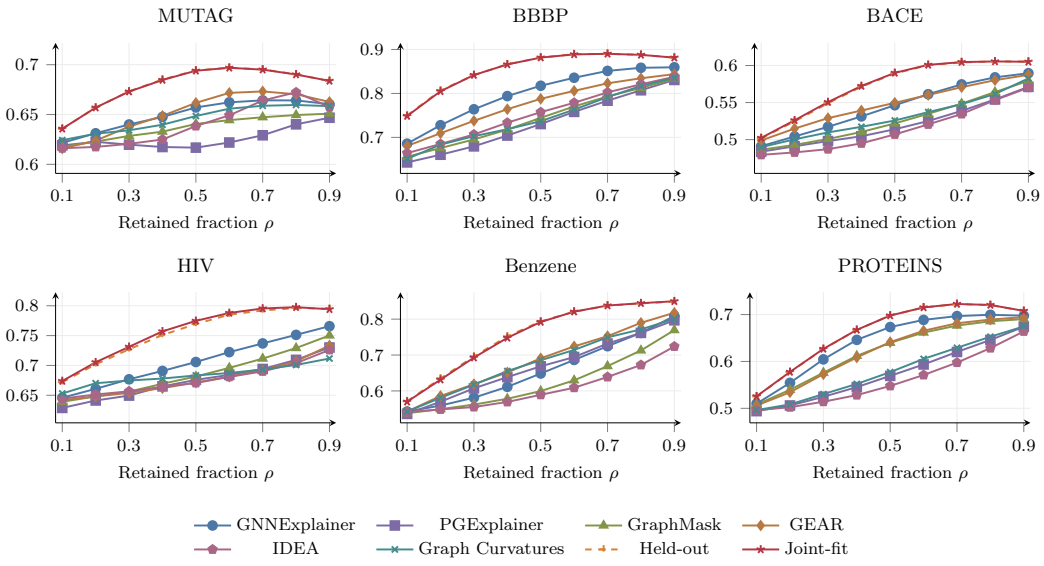

Figure A.6: Robust Fidelity across retained-edge fractions on GAT.

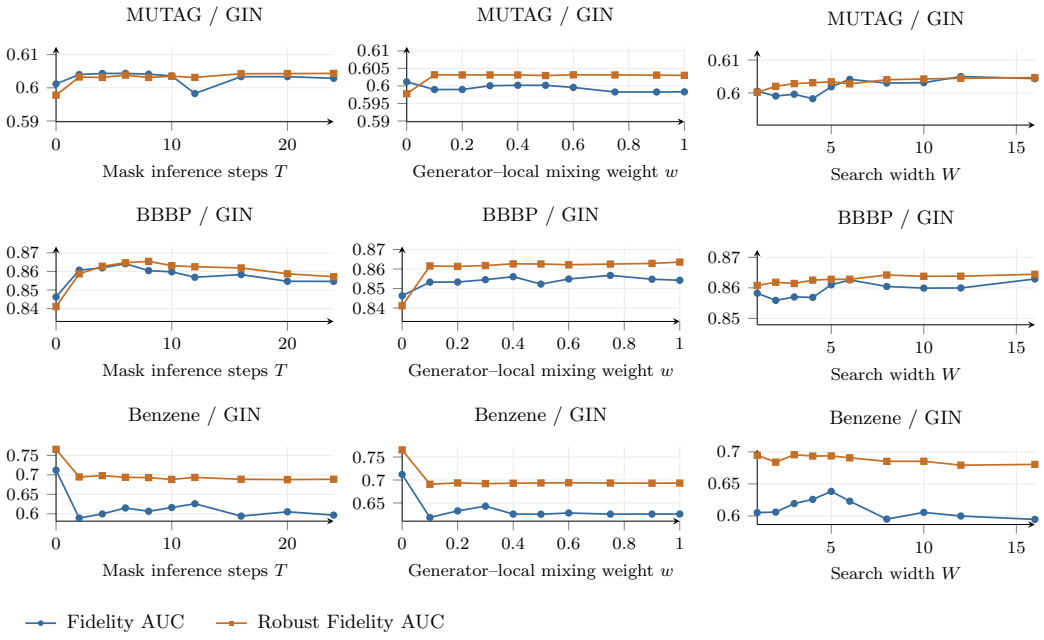

Figure A.7: GIN sensitivity across ten values per parameter. Rows indicate datasets and columns indicate parameters, with panel-specific vertical ranges. The mixing weight w denotes the candidate interpolation scale. Setting either T or w to zero returns the generated mask directly.