LLMs AS PROPOSERS: GENERATIVE CLASS-LEVEL COUNTERFACTUAL EXPLAINER FOR GNNS

Anonymous authors
Paper under double-blind review

ABSTRACT

Counterfactual explanations for graph neural networks (GNNs) require finding the minimal perturbations on the input graphs for changing GNNs' predictions. Nevertheless, the instance-level edits fail to provide a holistic view of the model's global decision boundaries. To address this challenge, we introduce LAPCE (LLM-as-Proposer for Counterfactual Explanation), a post-hoc class-level framework organized into the stages of propose–verify–select, identifying shared, discriminative subgraphs that systematically govern decision boundaries across a whole class. In the proposal stage, an instruction-conditioned chemical large language model (LLM) generates candidate subgraphs for counterfactual explanation, circumventing the combinatorial search in the discrete space. In the verification stage, each candidate subgraph is matched to a molecule, whose residual is further evaluated: The candidate subgraph is valid only if the residual molecule is valid and the GNN's predicted label changes. In the selection stage, the candidates are selected and ordered using their verification results to form a compact class-level explanation across semantic-cost thresholds and explanation budgets, while controlling redundancy. Empirical results show that LAPCE outperforms the strongest baseline across four molecular tasks. [Anonymous Project Page].

1 INTRODUCTION

Graph neural networks (GNNs) predict molecular properties such as biological activity and toxicity, but their outputs do not identify the structures supporting a decision. *Explanations* make GNNs' behavior open to inspection. Counterfactual explanations (Wachter et al., 2018) provide an intervention-based view by identifying structural changes that alter the prediction of a frozen model.

At the instance level, graph counterfactual explanation methods explain individual inputs through edited graphs or subgraphs (Lucic et al., 2022; Ma et al., 2022). Such an explanation does not summarize behavior shared by other molecules. Class-level methods address this broader scope through representative counterfactual graphs (Huang et al., 2023), generated molecules (Wang et al., 2024), subgraph mappings (He et al., 2025), or common recourses (Fournier & Medya, 2025).

Three connected challenges remain. First, useful candidates must be found in a combinatorial structural space. Edit-map exploration (Huang et al., 2023), deletion optimization (Lucic et al., 2022), and graph generation (Ma et al., 2022; Wang et al., 2024) can evaluate many structures that do not yield valid prediction-changing interventions. Second, reusing a candidate across a class requires parent-specific verification: a shared pattern alone does not establish that its application changes another molecule's prediction (Yuan et al., 2020; Wang & Shen, 2023). Third, a library selected for one budget and threshold need not offer an effective trade-off at smaller budgets or tighter cost thresholds. A class-level explanation therefore needs both reusable candidates and an ordering that supports multiple operating points. These challenges motivate focused proposal, independent cross-parent verification, and multi-budget selection in a single framework.

To address these challenges, we introduce LAPCE (LLM-as-Proposer for Counterfactual Explanation), a post-hoc framework organized into propose, verify, and select stages. A pretrained chemical LLM (Zhang et al., 2024) supplies molecular knowledge for candidate counterfactual explanation discovery. Given a molecule's SMILES representation (Weininger, 1988) and its GNN-predicted source class, the model proposes a subgraph for deletion-based testing. Supervised fine-tuning (SFT) adapts fragment output through parent–fragment correspondence supervision; proximal policy

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.

optimization (PPO) then incorporates the frozen GNN's response to deletion, with KL regularization against the SFT policy. The verifier records each candidate's valid prediction-changing effects across molecules in the same class. Class-level selection turns this evidence into one compact ordering, jointly considering explanation prefixes, semantic-cost thresholds, and redundancy, forming an ordered class-level explanation library.

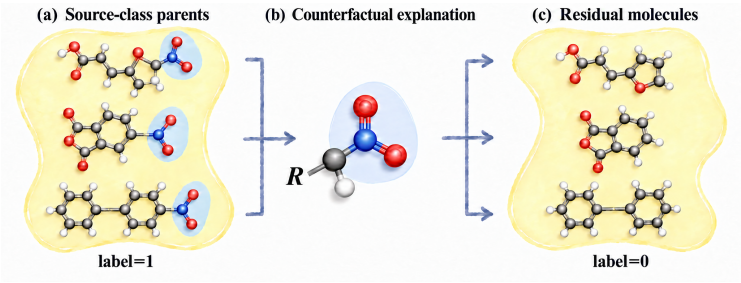


Figure 1: A cross-parent counterfactual explanation verified by the frozen GNN. Blue shading highlights the shared candidate; each displayed occurrence is deleted and verified independently.

Figure 1 illustrates the resulting class-level counterfactual explanation. A shared nitro-group candidate is reused across three structurally different parents, while each deletion is verified independently by the frozen GNN. This separates a reusable structural proposal from its parent-specific counterfactual effects.

Our contributions are threefold:

- **LLM-based candidate counterfactual explanation proposal.** We develop a task-adapted chemical LLM proposer for the combinatorial subgraph search space. Pretrained chemical priors and GNN-feedback adaptation support direct generation of molecule-conditioned counterfactual explanation candidates, without constructing an exhaustive inventory of substructures and edit combinations.
- **A generative class-level counterfactual explanation framework.** We develop a post-hoc framework that connects molecule-conditioned LLM generation to class-level counterfactual explanations of a frozen GNN. An intervention-cost matrix for each class records which parent molecule admits a valid prediction-changing intervention for each candidate counterfactual explanation and the corresponding minimum semantic cost, turning open-ended candidate counterfactual explanation generation into a finite class-level selection problem.
- **Class-level semantic-cost-aware selection.** We define a selection objective that jointly scores the prefixes of an explanation sequence across multiple semantic-cost thresholds, with penalties for structural redundancy, coverage overlap, and candidate size. It organizes verified candidate counterfactual explanations into one ordered class-level explanation whose prefixes serve different budgets without test-time reselection.

2 Related Work

Graph counterfactual explanations. Graph explainers return different objects, from predictive subgraphs to prediction-changing interventions (Yuan et al., 2023). GNNExplainer and PGExplainer identify predictive substructures (Ying et al., 2019; Luo et al., 2020), with related subgraph explainers including GStarX and GFlowExplainer (Zhang et al., 2022; Li et al., 2023); CF-GNNExplainer searches for edge deletions (Lucic et al., 2022), while CLEAR and D4Explainer learn graph-generative mechanisms (Ma et al., 2022; Chen et al., 2023). RCExplainer models decision logic shared across similar graphs for robust counterfactual edge-removal explanations (Bajaj et al., 2021). For molecules, SME explains predictions with chemical substructures (Wu et al., 2023), and Counterfactual Masking replaces regions using context-compatible completions (Janisiów et al., 2026). LAPCE uses generated subgraphs as reusable deletion candidates and evaluates their effects across source-class molecules.

Global and class-level graph explanations. Global recourse summaries characterize groups rather than isolated inputs (Rawal & Lakkaraju, 2020). XGNN and GNNInterpreter generate model-level graph patterns (Yuan et al., 2020; Wang & Shen, 2023). GCFExplainer selects representative counterfactual graphs from edit-map exploration (Huang et al., 2023); GlobalGCE learns subgraph

mappings (He et al., 2025), and COMRECGC constructs common recourses (Fournier & Medya, 2025). RLHEX uses a fragment-based variational generator and a PPO-trained latent-space adapter for global molecular counterfactuals (Wang et al., 2024). LAPCE combines LLM-generated structural candidates with a shared intervention-cost matrix and selection across explanation budgets.

Chemical language models and LLM-assisted graph counterfactual explanation. ChemLLM (Zhang et al., 2024) exposes pretrained chemical knowledge through molecular strings and text. LLM-GCE (He et al., 2024) uses counterfactual text pairs to guide a graph autoencoder and also studies direct whole-molecule prompting. Giorgi et al. (2025) use LLMs to verbalize counterfactual instances produced by existing explainers. LAPCE instead generates parent-conditioned fragment candidates, verifies their deletion effects across source-class parents, and selects a reusable ordered library. RLHEX (Wang et al., 2024) provides a non-LLM, PPO-adapted whole-molecule generative comparison. A numerical comparison with LLM-GCE is not included here; it would require specifying and evaluating an additional cross-parent realization and library-selection adapter.

3 Problem Formulation

Molecular graphs and source class. We study post-hoc, class-level explanations of a frozen molecular GNN. Let $G = (V, E, \mathbf{X}^V, \mathbf{X}^E) \in \mathbb{G}$ be a molecular graph, where V and E are its atoms and bonds, and $\mathbf{X}^V$ and $\mathbf{X}^E$ collect their respective attributes. A fixed classifier $f_\phi : \mathbb{G} \to \Delta^{C-1}$ has prediction $\hat{y}_\phi(G) = \arg\max_{c\in[C]} f_\phi(G)_c$. For source class y, define $D_y = \{G \in D : \hat{y}_\phi(G) = y\}$ and $N_y = |D_y| > 0$. Membership in D_y is defined by the frozen classifier's prediction rather than the ground-truth label.

Verified candidate interventions. Let $\mathcal{C}_y$ be a finite pool of connected attributed subgraph candidates. An occurrence of $s \in \mathcal{C}_y$ defines an executable test: delete its matched atoms and check the frozen GNN's prediction. The fixed atom- and bond-consistent matcher returns distinct matched atom sets $\mathcal{M}(G, s)$; symmetry-equivalent mappings with the same deletion atom set are represented once. For $m \in \mathcal{M}(G, s)$, the deterministic operator $\tau(G; s, m) \in \mathbb{G} \cup \{\perp\}$ applies the deletion to the original parent and returns a sanitized residual, or $\perp$ if invalid. The matching and residual-processing policies are fixed (Appendix A.1). Define

$$q_y(G, s, m) = \begin{cases} \mathbf{1}[\hat{y}_\phi(\tau(G; s, m)) \neq y], & \tau(G; s, m) \neq \perp, \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

A matched deletion is a valid counterfactual intervention exactly when $q_y(G, s, m) = 1$. A candidate is verified for a parent when at least one returned match yields a valid prediction change; if $\mathcal{M}(G, s)$ is empty, it is inapplicable to that parent. Different occurrences and candidates are tested independently against the original molecule. For nonnegative graph dissimilarity $\mathrm{dist} : \mathbb{G} \times \mathbb{G} \to \mathbb{R}_{\geq 0}$, the effective candidate cost is

$$d_y(G, s) = \min_{\substack{m\in\mathcal{M}(G,s) \\ q_y(G,s,m)=1}} \mathrm{dist}\big(G, \tau(G; s, m)\big), \qquad \min \varnothing := +\infty. \tag{2}$$

Class-level explanation. The goal is a compact set of reusable deletions for which as many source-class parents as possible admit a valid flip at low cost. For $S \subseteq \mathcal{C}_y$, let $d_y(G, S) = \min_{s\in S} d_y(G, s)$, with $d_y(G, \varnothing) = +\infty$. At intervention threshold θ, strict close-counterfactual coverage is

$$\mathrm{CCRCov}_{y,\theta}(S) = \frac{1}{N_y} \sum_{G\in D_y} \mathbf{1}[d_y(G, S) \leq \theta]. \tag{3}$$

Coverage counts parents rather than occurrences, so several successful deletions on the same molecule contribute only once. Inapplicable candidates and non-flipping deletions do not remove parents from the denominator; a parent with no successful candidate contributes zero. Given budget K, we seek $S_y \subseteq \mathcal{C}_y$ with $|S_y| \leq K$ that achieves high verified coverage at prespecified intervention-cost thresholds while remaining compact and non-redundant. LAPCE instantiates dist with WNode and constructs one ordering whose prefixes support different explanation budgets. The budget counts candidates; the threshold limits verified intervention cost.

4 METHOD

4.1 FRAMEWORK OVERVIEW

LAPCE combines candidate proposal, independent intervention verification, and class-level selection. SFT and frozen-GNN feedback adapt a chemical LLM to propose connected subgraphs. Verification records their minimum successful intervention costs across parents; selection uses this evidence to construct one ordered explanation library. The explained GNN remains frozen. Figure 2 summarizes the pipeline, and Algorithm 1 summarizes its implementation; the independent match-level verifier is Algorithm 2 in Appendix A.2.

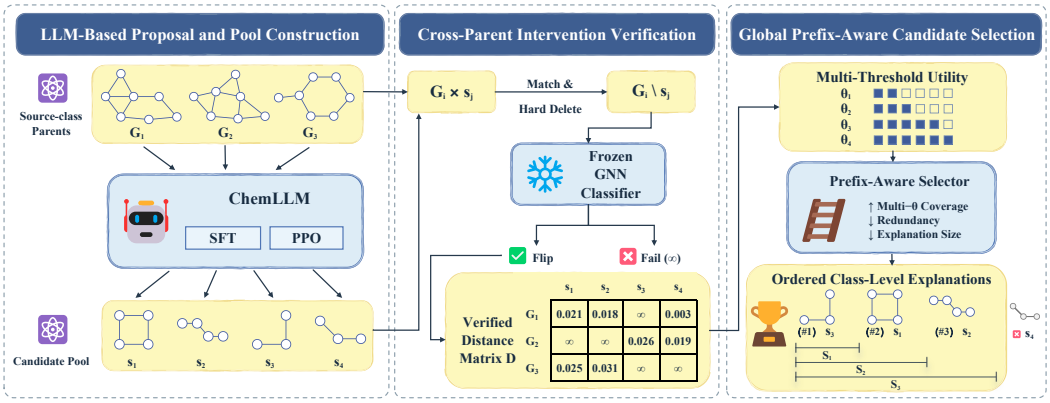


Figure 2: Illustration of LAPCE's propose–verify–select workflow. Independent parent–candidate verification forms a reusable intervention-cost matrix; prefix-aware selection constructs one ordered library. The displayed matrix and ordering illustrate the data flow.

4.2 LLM-BASED CANDIDATE PROPOSAL AND TASK ADAPTATION

Instruction-conditioned proposal. Let $x(G)$ be the canonical SMILES representation of parent G (Weininger et al., 1989) and $y = \hat{y}_\phi(G)$ its source class. A fixed instruction requests one connected fragment SMILES. We denote its sequence distribution by $q_\psi(s \mid x(G), y)$, suppressing the instruction in the notation. Generated strings are parsed into subgraph candidates for matched-deletion testing. Appendix C.1 gives the prompt; Appendix C.2 gives decoding settings, and Appendix C.3 fixes parent-only projection.

Supervised adaptation. On non-test parent–fragment pairs, token-level cross-entropy trains the proposer to produce structurally supported output:

$$\mathcal{L}_{\mathrm{SFT}}(\psi) = -\mathbb{E}_{(G,y,s)} \sum_{t=1}^{L_s} \log q_\psi(s_t \mid s_{<t}, x(G), y). \tag{4}$$

Here s_t is token t of a fragment sequence of length L_s. Targets are directly matched connected fragments admitting a sanitized deletion residual. This structural supervision precedes the GNN-feedback adaptation below; pair construction and training settings are specified in Appendix C.1.

Frozen-GNN feedback. PPO (Schulman et al., 2017) updates the proposer using

$$\begin{aligned} R_{\mathrm{prop}}(G,u) = \alpha_{\mathrm{val}} r_{\mathrm{val}} + \alpha_{\mathrm{sub}} r_{\mathrm{sub}} + \alpha_{\mathrm{cf}} r_{\mathrm{cf}} - \alpha_{\mathrm{size}} r_{\mathrm{size}} \\ - \alpha_{\mathrm{proj}} r_{\mathrm{proj}} - \beta_{\mathrm{KL}} \widehat{k}_{\mathrm{ref}}(u; G, y). \end{aligned} \tag{5}$$

The feedback rewards connected output, direct matching, and prediction change. Size and projection penalties discourage near-parent fragments and fallback reliance. The frozen SFT reference supplies the sampled token-mean log-ratio $\widehat{k}_{\mathrm{ref}}$ for response u. Appendix C.2 specifies the joint match-level feedback and PPO optimization; the explained GNN remains fixed.

Class-level candidate pool. For each parent in the non-test proposal split D_y^{prop}, we sample R responses, with $R = R_0 = 8$ in the reference configuration. Canonicalization and deduplication retain source-parent metadata:

$$\mathcal{C}_y = \text{Dedup}\left(\bigcup_{G \in D_y^{\text{prop}}} \bigcup_{r=1}^{R} \{\text{Can}(s_{G,r})\} \right). \tag{6}$$

The fixed schedule appears in Appendix C.2. Canonical identities define the pool; verification establishes the parent-specific counterfactual effects.

4.3 Deterministic Verification and WNode Effect Matrix

Independent intervention verification. Each distinct returned atom set is deleted independently from the original parent. If the residual disconnects, τ retains its largest heavy-atom component, breaking ties by canonical SMILES; empty or single-heavy-atom residuals are invalid. Sanitization and a strict flip then determine the successful matches, whose minimum WNode defines $d_y(G, s)$ in Eq. (2). This evaluates the complete processed residual; candidate size measures the proposed fragment, not all atoms discarded by τ. Appendix A.1 gives the fixed policy.

Contextual molecular distance. WNode measures semantic intervention cost by optimal transport over contextual atom representations (Peyré & Cuturi, 2019; Togninalli et al., 2019). For a frozen encoder h_ω, let $z_u = h_\omega(G)_u$ and $z'_v = h_\omega(G')_v$. With node-mass vectors a, b and transport polytope $\Pi(a, b)$,

$$D_{\text{WNode}}(G, G') = \min_{T \in \Pi(a,b)} \sum_{u \in V(G)} \sum_{v \in V(G')} T_{uv} \left(1 - \frac{z_u^\top z'_v}{\|z_u\|_2 \|z'_v\|_2} \right). \tag{7}$$

We use a frozen MolCLR-GIN encoder (Wang et al., 2022), uniform node masses, and exact balanced EMD. WNode quantifies representation change; molecular validity is determined separately by the fixed intervention policy. Appendix A.1 documents the representation, solver, and feasibility checks.

Cross-parent intervention matrix. For calibration parents $D_y^{\text{cal}} = \{G_i\}_{i=1}^{N_y^{\text{cal}}}$ and candidates $\mathcal{C}_y = \{s_j\}_{j=1}^{P_y}$, verification produces

$$D^{\text{cal}} = [d_y(G_i, s_j)]_{i,j} \in (\mathbb{R}_{\geq 0} \cup \{+\infty\})^{N_y^{\text{cal}} \times P_y}. \tag{8}$$

Each finite entry certifies at least one sanitized prediction-changing intervention and records its least cost. This fixed evidence matrix allows candidate selection to reuse verified effects without further GNN queries on calibration parents. Completion status is tracked separately; $+\infty$ denotes a completed pair with no valid prediction flip.

4.4 Class-Level Selection from Candidate Verification Results

Multi-threshold prefix utility. For calibration-fixed thresholds $\Theta = \{\theta_1, \ldots, \theta_M\}$, selection aggregates coverage over intervention-cost budgets:

$$F_\Theta(S) = \sum_{j=1}^{M} w_j \, \text{CCRCov}_{y,\theta_j}(S), \qquad w_j \geq 0. \tag{9}$$

An explicit cost term uses the negative capped mean over the same calibration cohort:

$$Q_{\text{cost}}(S) = -\frac{1}{N_y^{\text{cal}}} \sum_{i=1}^{N_y^{\text{cal}}} \min\{d_y(G_i, S), d_{\text{cap}}\}. \tag{10}$$

The fixed cap gives unsuccessful parents a common finite contribution. For ordering $\pi_y = (s_1, \ldots, s_K)$ and prefixes $S_k = \{s_1, \ldots, s_k\}$, we optimize

$$\begin{aligned} \pi_y^\star \in \arg \max_{\pi_y \in \text{Perm}_K(\mathcal{C}_y)} \sum_{k=1}^{K} \rho_k \big[& F_\Theta(S_k) + \lambda_{\text{cost}} Q_{\text{cost}}(S_k) \\ & - \lambda_{\text{str}} \mathcal{R}_{\text{str}}(S_k) - \lambda_{\text{cov}} \mathcal{R}_{\text{cov}}(S_k) - \lambda_{\text{size}} \mathcal{R}_{\text{size}}(S_k) \big]. \end{aligned} \tag{11}$$

Here Perm_K contains ordered selections of distinct candidates and $\rho_k \geq 0$ weights the budgets. Earlier coverage contributes to more prefixes. Prefix growth enlarges the explanation library; its candidates are still applied independently to the original parent. The short-pool and stopping conventions are given in Appendix B.1.

Why prefix scoring rewards early coverage. Let r_{ij} be the first prefix covering calibration parent i at threshold θ_j. Rearranging the coverage part of Eq. (11) gives

$$\sum_{k=1}^{K} \rho_k F_\Theta(S_k) = \frac{1}{N_y^{\mathrm{cal}}} \sum_i \sum_j w_j \sum_{k=r_{ij}}^{K} \rho_k,$$

with zero contribution when the parent is never covered. Earlier coverage accumulates more prefix weight; refinement can change both membership and order to improve this complete-sequence criterion.

Redundancy and candidate-size control. At threshold θ_ℓ, a candidate covers the calibration set

$$A_{y,\theta_\ell}^{\mathrm{cal}}(s) = \{G_i \in D_y^{\mathrm{cal}} : d_y(G_i, s) \leq \theta_\ell\}. \tag{12}$$

Structural redundancy averages Tanimoto similarity of binary radius-2, 2048-bit Morgan fingerprints (Appendix B.4):

$$\mathcal{R}_{\mathrm{str}}(S) = \frac{2}{|S|(|S|-1)} \sum_{\substack{s,t \in S \\ s<t}} \mathrm{sim}_{\mathrm{chem}}(s,t). \tag{13}$$

Repeated parent coverage is penalized by

$$\mathcal{R}_{\mathrm{cov}}(S) = \frac{2}{|S|(|S|-1)} \sum_{\substack{s,t \in S \\ s<t}} \sum_{j=1}^{M} \bar{w}_j \frac{|A_{y,\theta_j}^{\mathrm{cal}}(s) \cap A_{y,\theta_j}^{\mathrm{cal}}(t)|}{|A_{y,\theta_j}^{\mathrm{cal}}(s) \cup A_{y,\theta_j}^{\mathrm{cal}}(t)|}. \tag{14}$$

Here $\bar{w}_j = w_j / \sum_\ell w_\ell$ with a positive total weight. Empty unions contribute zero, and pairwise penalties vanish for $|S| < 2$. Candidate size is the mean heavy-atom count divided by the maximum count in the fixed full candidate pool. Appendix B.2 specifies the thresholds, coefficients, refinement settings, and the distinct coverage, overlap, and prefix weight scales.

Inputs and execution. Algorithm 1 uses non-test SFT pairs $\mathcal{P}_y$, PPO parents D_y^{adapt}, and the proposal split. Each sampled response is parsed, projected onto its parent only when direct matching fails, and merged by canonical identity while retaining origin IDs. Verification is Algorithm 2; the GNN, matching, residual policy, and WNode are fixed.

Algorithm 1: LAPCE: propose–verify–select

Require: Pretrained q_{ψ_0}, source y, budgets K, R
Fit q_{SFT} from q_{ψ_0} on non-test pairs $\mathcal{P}_y$
Freeze the reference $q_{\mathrm{ref}} \leftarrow q_{\mathrm{SFT}}$
Adapt q_ψ by PPO on D_y^{adapt}
Initialize candidate pool $\mathcal{C}_y \leftarrow \varnothing$
for each parent $G \in D_y^{\mathrm{prop}}$ **do**
    Sample R fragment responses conditioned on G
    Merge accepted candidates and their origin IDs
end for
Verify $\mathcal{C}_y$ on calibration parents
Fix selector settings γ using calibration data
Select ordering π_y using Eq. (11)
Freeze π_y and its selected candidate identities
Verify those candidates on held-out parents
Evaluate coverage and cost of frozen prefixes
return π_y and its held-out metrics

Optimization and held-out evaluation. Greedy initialization is refined by swaps, relocations, and replacements under Appendix B.2's settings. For fixed verified costs, F_Θ is monotone submodular (Nemhauser et al., 1978); the complete regularized sequence objective is optimized heuristically. Calibration fixes $\gamma = (\Theta, w, \rho, \lambda, d_{\mathrm{cap}}, \ldots)$ and the ordering before held-out verification. Each finite matrix entry certifies a parent-specific intervention. All reported budgets use prefixes of this one frozen library, without test-time reselection.

5 EXPERIMENTS

5.1 EVALUATION PROTOCOL AND COMPARATIVE QUALITY

We evaluate HIV, Mutagenicity, BACE, and three-class TasteMolNet (Wu et al., 2018; Kazius et al., 2005; Subramanian et al., 2016; Shi et al., 2026) (Table 1). Within each task, methods share the frozen GINE (Hu et al., 2020) and explanation-test cohort. Sweet is the TasteMolNet source class; either other predicted class constitutes a strict flip. Classification-test and explanation-cohort memberships are defined separately in Appendix E.

Table 1: Dataset views (Mut.: Mutagenicity; Taste: TasteMolNet). Source-class molecules (%) is the fraction in the explanation source class: HIV-active, mutagenic, BACE-active, and Sweet, respectively.

	HIV	Mut.	BACE	Taste
Classes	2	2	2	3
Molecules	41,120	4,247	1,513	13,421
Atoms	1,048,955	70,701	51,577	359,767
Bonds	1,129,451	75,230	55,768	376,371
Atom types	54	9	8	19
Source class (%)	3.51	55.62	45.67	41.43

We compare GCFExplainer (Huang et al., 2023), GlobalGCE (He et al., 2025), COMRECGC (Fournier & Medya, 2025), CM-CReM+ (our class-level realization) (Janisiów et al., 2026; Polishchuk, 2020), and RLHEX (Wang et al., 2024) using the parent–element realizations in Appendix D. A single calibration-selected ordering is frozen for both budget and threshold sweeps. Cov@K is strict CCRCov (Eq. (3)); conditional cost is the median best finite strict-flip WNode, without an additional threshold filter. The reference is $K = 20$ and $\theta = d_{\text{cap}} = 0.1$. Native-element budgets do not equate editing or query budgets. Bold and underline identify the best and second-best distinct values; ratios use source precision (Appendix A.4).

Consistent coverage–cost gains. At $K = 20$ and $\theta = 0.1$, LAPCE achieves the highest coverage and lowest conditional median WNode on all four tasks (Table 2). RLHEX is the strongest coverage baseline on each task at this operating point. Relative to the metric-specific best baseline, task-macro improvements are 3.01% in coverage and 3.19% in conditional cost. Absolute coverage gains are 2.29, 2.68, 1.07, and 1.31 percentage points on HIV, Mutagenicity, BACE, and TasteMolNet, respectively.

Table 2: Frozen-GINE comparison at $K = 20, \theta = 0.1$. Coverage is a percentage; cost is conditional median WNode. Preferred direction: ↑ higher is better; ↓ lower is better.

	HIV		Mutagenicity		BACE		TasteMolNet	
Method	Cov↑	Cost↓	Cov↑	Cost↓	Cov↑	Cost↓	Cov↑	Cost↓
GCFExplainer	71.81	0.0759	70.02	0.0678	37.82	0.0642	43.65	0.0431
GlobalGCE	55.84	0.0695	67.29	0.0736	33.33	0.0673	36.72	0.0460
COMRECGC	66.84	0.0724	75.58	0.0785	36.97	0.0628	41.28	0.0447
CM-CReM+	59.36	0.0668	68.96	0.0655	34.62	<u>0.0621</u>	35.84	0.0406
RLHEX	<u>74.10</u>	<u>0.0665</u>	<u>79.37</u>	<u>0.0645</u>	<u>38.03</u>	0.0632	<u>47.81</u>	<u>0.0405</u>
LAPCE	**76.39**	**0.0645**	**82.05**	**0.0620**	**39.10**	**0.0617**	**49.12**	**0.0384**

Appendix F.1 reports paired coverage margins against the strongest of the same five baselines, including RLHEX; all four 95% intervals are positive. Appendix F.2 separately compares conditional costs with fixed CM-CReM+; BACE's interval includes zero.

Budget profiles. Across the eight evaluated budgets $K \in \{1, 3, 6, 9, 12, 15, 18, 20\}$, LAPCE has higher coverage than RLHEX in 31 of 32 task–budget pairs and lower conditional cost in 31 of 32. The exceptions are Mutagenicity coverage at $K = 1$ and HIV cost at $K = 3$. At the six evaluated budgets $K \geq 6$, it leads all five baselines on both metrics for every task.

Threshold profiles. At $K = 20$, LAPCE exceeds RLHEX at all 28 positive evaluated task–threshold checkpoints (Figure 4). It leads all five baselines at every positive sampled threshold on

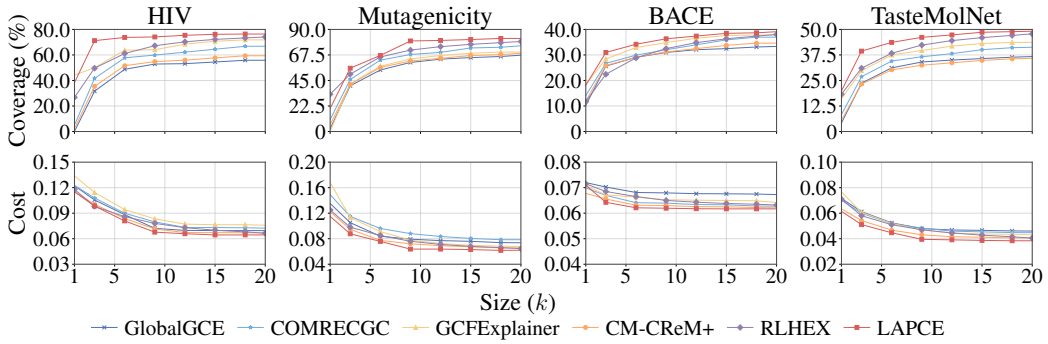


Figure 3: Coverage and conditional median WNode versus explanation budget K ($\theta = 0.1$). Lines connect evaluated budgets; cost axes are task-specific.

HIV, Mutagenicity, and TasteMolNet, and at $\theta \geq 0.04$ on BACE. Mutagenicity reaches 85.09% finite reachability at $\theta = 0.12$.

5.2 Proposal Adaptation and Backbone Compatibility

Table 3(a) varies the BACE proposer while fixing GINE and verification–selection; BRICS uses retrosynthetic cleavage rules (Degen et al., 2008). Panel (b) reports results for five frozen GNNs. Both share the full reference of 39.10% coverage and 0.0617 cost. Diagnostic denominators are defined in Appendix C.4.

Table 3: BACE results at $K = 20$ and $\theta = 0.1$; 2B/7B denote ChemLLM. Match and Flip are raw-attempt percentages; Flip includes allowed projection. Reuse is candidate-macro. The dagger denotes the bond-aware GatedGCN configuration in Appendix E.2.

(a) Candidate generation and utility						(b) Frozen GNN		
Source	Cov↑	Cost↓	Match↑	Flip↑	Reuse↑	Backbone	Cov↑	Cost↓
BRICS	19.87	0.0735	**100.0**	5.8	4.3	GINE	**39.10**	**0.0617**
2B	17.74	0.0751	49.6	4.7	3.1	GIN	<u>34.83</u>	0.0718
7B	20.51	0.0744	62.8	7.1	4.6	GatedGCN†	33.55	<u>0.0683</u>
7B+SFT	<u>27.56</u>	<u>0.0716</u>	79.6	<u>11.2</u>	<u>7.2</u>	GATv2	28.42	0.0751
7B+SFT+PPO	**39.10**	**0.0617**	<u>81.3</u>	**17.6**	**11.3**	GCN	23.93	0.0775

SFT improves coverage by 7.05 percentage points and lowers cost by approximately 3.76%; PPO adds 11.54 points and lowers cost by 13.83%. Against BRICS, the full pipeline gains 19.23 points and reduces cost by 16.05%. Structural supervision and frozen-GNN feedback make complementary contributions.

Executability and counterfactual utility are distinct. BRICS directly matches 100% of attempts, but its strict-flip yield and reuse are 5.8% and 4.3%, versus 17.6% and 11.3% for SFT+PPO. SFT raises matching from 62.8% to 79.6%; PPO then raises strict flipping from 11.2% to 17.6% and reuse from 7.2% to 11.3%. The adapted proposer improves intervention utility, not just syntactic validity.

Across the five frozen backbones, coverage spans 23.93–39.10%, with GINE leading both metrics. This establishes compatibility beyond one architecture (Appendix E). With the libraries fixed, rejecting disconnected residuals retains 96.72% and 98.09% of originally covered parents on BACE and Mutagenicity, respectively (Appendix A.5).

5.3 Selector Objectives and Budget Trade-offs

With candidates, frozen GNNs, verified matrices, and cohorts fixed, S0/S1 use single-/multi-threshold greedy selection; S3/S5 add terminal/prefix refinement. S6 is full LAPCE. From the same initial-

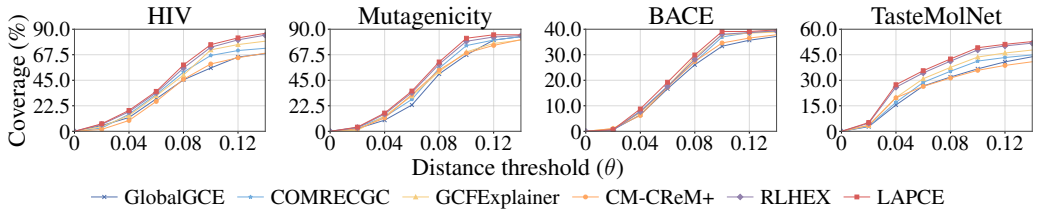


Figure 4: Coverage versus WNode threshold for frozen top-20 libraries. Lines connect evaluated thresholds; $\theta = 0.1$ reproduces Table 2.

ization, S7–S10 remove coverage overlap, structural similarity, candidate size, and explicit cost, respectively. S11 uses terminal-only budget weight; S12 uses reporting-threshold coverage at unchanged scale. Table 4 summarizes outcomes; Tables 5–7 define all thirteen variants and diagnostics.

Table 4: Selector outcomes (S6=LAPCE). MeanCov averages coverage over $K = 1, \ldots, 20$. A_θ is normalized threshold area and C_{cap} capped mean cost; $A_\theta = 1 - C_{\text{cap}}/0.1$.

	BACE				Mutagenicity			
Variant	MeanCov↑	A_θ ↑	Cov@20↑	C_{cap} ↓	MeanCov↑	A_θ ↑	Cov@20↑	C_{cap} ↓
S0	24.84	0.104	30.13	0.0896	60.69	0.265	71.84	0.0735
S1	30.03	0.152	36.32	0.0848	68.01	0.335	78.82	0.0665
S3	31.06	0.160	37.82	0.0840	69.23	0.350	80.13	0.0650
S5	33.43	**0.166**	37.82	**0.0834**	71.99	0.358	80.43	0.0642
S6	**34.34**	0.165	**39.10**	0.0835	**73.33**	**0.360**	**82.05**	**0.0640**
S10	33.80	0.159	38.46	0.0841	72.60	0.352	81.75	0.0648
S11	31.79	0.164	38.46	0.0836	70.60	0.359	81.80	0.0641
S12	32.20	0.156	38.03	0.0844	70.80	0.347	80.79	0.0653

Restoring explicit cost (S10 to S6) lowers capped cost by 0.0006 on BACE and 0.0008 on Mutagenicity while increasing Cov@20 by 0.64 and 0.30 points. Explicit cost preference complements multi-threshold coverage.

Against terminal-only S11, S6 increases coverage by 3.21 and 3.89 points at $K = 5$, 2.35 and 2.63 at $K = 10$, and 0.64 and 0.25 at $K = 20$ on BACE and Mutagenicity, respectively. The larger early gains show the value of prefix scoring while retaining terminal coverage.

Against single-threshold S12, S6 improves A_θ by 0.009 and 0.013 and Cov@20 by 1.07 and 1.26 points. The S7–S9 comparisons show that respective penalties reduce coverage overlap, structural similarity, and candidate size. Appendices F.3 and C.5 report BACE sensitivity and the direct-supported-pool control.

6 Conclusion

LAPCE addresses class-level molecular counterfactual explanation through task-adapted fragment proposal, independent parent-wise verification, and prefix-aware selection. A chemical LLM focuses candidate discovery, while the frozen-GNN verifier independently validates matched deletions and records minimum successful WNode costs; one ordered library serves multiple budgets without test-time reselection. Across four tasks, LAPCE leads five baselines in coverage and conditional cost at $K = 20$ and $\theta = 0.1$. Paired coverage-margin intervals against the strongest of these baselines are positive on all four tasks, conditional on the frozen models and libraries. This inference concerns parent resampling, not retraining variability or every operating point. The separate fixed-CM-CReM+ cost comparison does not establish a nonzero BACE difference, despite its lower LAPCE point estimate. Ablations examine proposal adaptation, explicit cost, threshold aggregation, and early prefix coverage. Future work will target faster verification and richer chemistry-aware intervention constraints without weakening the requirement for independently verified prediction changes.

AI USE STATEMENT

Generative AI tools were used solely for language refinement. All technical content, analyses, experiments, and conclusions were independently developed and verified by the authors.

ETHICS STATEMENT

This work studies post-hoc explanations of frozen molecular GNNs using publicly available molecular datasets. It does not involve human subjects or personally identifiable information. The generated counterfactual molecules and substructures are intended for model interpretation rather than claims of biological activity, safety, or synthetic feasibility.

REFERENCES

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

Kenza Amara, Zhitao Ying, Zitao Zhang, Zhichao Han, Yang Zhao, Yinan Shan, Ulrik Brandes, Sebastian Schemm, and Ce Zhang. GraphFramEx: Towards systematic evaluation of explainability methods for graph neural networks. In Bastian Rieck and Razvan Pascanu (eds.), *Learning on Graphs Conference, LoG 2022, 9-12 December 2022, Virtual Event*, volume 198 of *Proceedings of Machine Learning Research*, pp. 44:1–44:23. PMLR, 2022. URL https://proceedings.mlr.press/v198/amara22a.html.

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

Xavier Bresson and Thomas Laurent. Residual gated graph convnets. *arXiv preprint arXiv:1711.07553*, 2017. URL https://arxiv.org/abs/1711.07553.

Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks? In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=F72ximsx7C1.

Jialin Chen, Shirley Wu, Abhijit Gupta, and Rex Ying. D4Explainer: In-distribution explanations of graph neural network via discrete denoising diffusion. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, volume 36, pp. 78964–78986, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/f978c8f3b5f399cae464e85f72e28503-Abstract-Conference.html.

Jörg Degen, Christof Wegscheid-Gerlach, Andrea Zaliani, and Matthias Rarey. On the art of compiling and using 'Drug-Like' chemical fragment spaces. *ChemMedChem*, 3(10):1503–1507, 2008. doi: 10.1002/cmdc.200800178.

Rémi Flamary, Nicolas Courty, Alexandre Gramfort, Mokhtar Z. Alaya, Aurélie Boisbunon, Stanislas Chambon, Laetitia Chapel, Adrien Corenflos, Kilian Fatras, Nemo Fournier, Léo Gautheron, Nathalie T. H. Gayraud, Hicham Janati, Alain Rakotomamonjy, Ievgen Redko, Antoine Rolet, Antony Schutz, Vivien Seguy, Danica J. Sutherland, Romain Tavenard, Alexander Tong, and Titouan Vayer. POT: Python optimal transport. *Journal of Machine Learning Research*, 22(78): 1–8, 2021. URL https://www.jmlr.org/papers/v22/20-451.html.

Gregoire Fournier and Sourav Medya. COMRECGC: global graph counterfactual explainer through common recourse. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*, pp. 17522–17541. PMLR / OpenReview.net, 2025. URL https://proceedings.mlr.press/v267/fournier25a.html.

Flavio Giorgi, Cesare Campagnano, Fabrizio Silvestri, and Gabriele Tolomei. Natural language counterfactual explanations for graphs using large language models. In Yingzhen Li, Stephan Mandt, Shipra Agrawal, and Mohammad Emtiyaz Khan (eds.), *International Conference on Artificial Intelligence and Statistics, AISTATS 2025, Mai Khao, Thailand, 3-5 May 2025*, volume 258 of *Proceedings of Machine Learning Research*, pp. 3565–3573. PMLR, 2025. URL https://proceedings.mlr.press/v258/giorgi25a.html.

Yinhan He, Zaiyi Zheng, Patrick Soga, Yaochen Zhu, Yushun Dong, and Jundong Li. Explaining graph neural networks with large language models: A counterfactual perspective on molecule graphs. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, pp. 7079–7096. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.FINDINGS-EMNLP.415. URL https://doi.org/10.18653/v1/2024.findings-emnlp.415.

Yinhan He, Wendy Zheng, Yaochen Zhu, Jing Ma, Saumitra Mishra, Natraj Raman, Ninghao Liu, and Jundong Li. Global graph counterfactual explanation: A subgraph mapping approach. *Transactions on Machine Learning Research*, 2025. URL https://openreview.net/forum?id=KQzJYI6eo0.

Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay S. Pande, and Jure Leskovec. Strategies for pre-training graph neural networks. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=HJlWWJSFDH.

Zexi Huang, Mert Kosan, Sourav Medya, Sayan Ranu, and Ambuj K. Singh. Global counterfactual explainer for graph neural networks. In Tat-Seng Chua, Hady W. Lauw, Luo Si, Evimaria Terzi, and Panayiotis Tsaparas (eds.), *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining, WSDM 2023, Singapore, 27 February 2023 - 3 March 2023*, pp. 141–149. ACM, 2023. doi: 10.1145/3539597.3570376. URL https://doi.org/10.1145/3539597.3570376.

Łukasz Janisiów, Marek Kochańczyk, Bartosz Michał Zieliński, and Tomasz Danel. Enhancing chemical explainability through counterfactual masking. In *Fortieth AAAI Conference on Artificial Intelligence*, pp. 22155–22163. AAAI Press, 2026. doi: 10.1609/aaai.v40i26.39371. URL https://doi.org/10.1609/aaai.v40i26.39371.

Jeroen Kazius, Ross McGuire, and Roberta Bursi. Derivation and validation of toxicophores for mutagenicity prediction. *Journal of Medicinal Chemistry*, 48(1):312–320, 2005. doi: 10.1021/jm040835a. URL https://doi.org/10.1021/jm040835a.

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

Greg Landrum, Paolo Tosco, Brian Kelley, Ricardo Rodriguez, David Cosgrove, Riccardo Vianello, sriniker, Peter Gedeck, Gareth Jones, Eisuke Kawashima, NadineSchneider, Dan Nealschneider, Andrew Dalke, tadhurst cdd, Matt Swain, Brian Cole, Samo Turk, Aleksandr Savelev, Alain Vaucher, Maciej Wójcikowski, Ichiru Take, Hussein Faara, Vincent F. Scalfani, Rachel Walker, Daniel Probst, Kazuya Ujihara, Niels Maeder, Jeremy Monat, Juuso Lehtivarjo, and guillaume godin. rdkit/rdkit: 2025_03_5 (q1 2025) release. Zenodo, 2025. URL https://doi.org/10.5281/zenodo.16439048.

Wenqian Li, Yinchuan Li, Zhigang Li, Jianye Hao, and Yan Pang. DAG matters! GFlowNets enhanced explainer for graph neural networks. In *The Eleventh International Conference on*

Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023. OpenReview.net, 2023. URL https://openreview.net/forum?id=jgmuRzM-sb6.

Ana Lucic, Maartje A. Ter Hoeve, Gabriele Tolomei, Maarten De Rijke, and Fabrizio Silvestri. CF-GNNExplainer: Counterfactual explanations for graph neural networks. In *International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pp. 4499–4511. PMLR, 2022. URL https://proceedings.mlr.press/v151/lucic22a.html.

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

Jing Ma, Ruocheng Guo, Saumitra Mishra, Aidong Zhang, and Jundong Li. CLEAR: generative counterfactual explanations on graphs. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, volume 35, pp. 25895–25907, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/a69d7f3a1340d55c720e572742439eaf-Abstract-Conference.html.

George L. Nemhauser, Laurence A. Wolsey, and Marshall L. Fisher. An analysis of approximations for maximizing submodular set functions - I. *Math. Program.*, 14(1):265–294, 1978. doi: 10.1007/BF01588971. URL https://doi.org/10.1007/BF01588971.

Gabriel Peyré and Marco Cuturi. Computational optimal transport: With applications to data science. *Foundations and Trends in Machine Learning*, 11(5–6):355–607, 2019. doi: 10.1561/2200000073. URL https://doi.org/10.1561/2200000073.

Pavel G. Polishchuk. CReM: chemically reasonable mutations framework for structure generation. *J. Cheminformatics*, 12(1):28, 2020. doi: 10.1186/S13321-020-00431-W. URL https://doi.org/10.1186/s13321-020-00431-w.

Kaivalya Rawal and Himabindu Lakkaraju. Beyond individualized recourse: Interpretable and interactive summaries of actionable recourses. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/8ee7730e97c67473a424ccfeff49ab20-Abstract.html.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. URL https://arxiv.org/abs/1707.06347.

Peiqin Shi, Mengfei Yang, Rui Chang, and Hongli Yao. TasteMolNet: A machine learning-driven platform for sweet, bitter, and tasteless compounds prediction in food chemistry. *Journal of Food Composition and Analysis*, 150:108888, 2026. doi: 10.1016/j.jfca.2026.108888. URL https://www.sciencedirect.com/science/article/pii/S0889157526000311.

Govindan Subramanian, Bharath Ramsundar, Vijay Pande, and Rajiah Aldrin Denny. Computational modeling of β-secretase 1 (BACE-1) inhibitors using ligand based approaches. *Journal of Chemical Information and Modeling*, 56(10):1936–1949, 2016. doi: 10.1021/acs.jcim.6b00290. URL https://doi.org/10.1021/acs.jcim.6b00290.

Matteo Togninalli, M. Elisabetta Ghisu, Felipe Llinares-López, Bastian Rieck, and Karsten M. Borgwardt. Wasserstein weisfeiler-lehman graph kernels. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in*

Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada, pp. 6436–6446, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/73fed7fd472e502d8908794430511f4d-Abstract.html.

Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the GDPR. *Harvard Journal of Law & Technology*, 31(2): 841–887, 2018.

Danqing Wang, Antonis Antoniades, Kha-Dinh Luong, Edwin Zhang, Mert Kosan, Jiachen Li, Ambuj K. Singh, William Yang Wang, and Lei Li. Global human-guided counterfactual explanations for molecular properties via reinforcement learning. In Ricardo Baeza-Yates and Francesco Bonchi (eds.), *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2024, Barcelona, Spain, August 25-29, 2024*, pp. 2991–3000. ACM, 2024. doi: 10.1145/3637528.3672045. URL https://doi.org/10.1145/3637528.3672045.

Xiaoqi Wang and Han-Wei Shen. GNNInterpreter: A probabilistic generative model-level explanation for graph neural networks. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=rqq6Dh8t4d.

Yuyang Wang, Jianren Wang, Zhonglin Cao, and Amir Barati Farimani. Molecular contrastive learning of representations via graph neural networks. *Nat. Mach. Intell.*, 4(3):279–287, 2022. doi: 10.1038/S42256-022-00447-X. URL https://doi.org/10.1038/s42256-022-00447-x.

David Weininger. SMILES, a chemical language and information system. 1. introduction to methodology and encoding rules. *J. Chem. Inf. Comput. Sci.*, 28(1):31–36, 1988. doi: 10.1021/CI00057A005. URL https://doi.org/10.1021/ci00057a005.

David Weininger, Arthur Weininger, and Joseph L. Weininger. SMILES. 2. algorithm for generation of unique SMILES notation. *J. Chem. Inf. Comput. Sci.*, 29(2):97–101, 1989. doi: 10.1021/CI00062A008. URL https://doi.org/10.1021/ci00062a008.

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. doi: 10.1039/C7SC02664A. URL https://doi.org/10.1039/C7SC02664A.

Zhenxing Wu, Jike Wang, Hongyan Du, Dejun Jiang, Yu Kang, Dan Li, Peichen Pan, Yafeng Deng, Dongsheng Cao, Chang-Yu Hsieh, and Tingjun Hou. Chemistry-intuitive explanation of graph neural networks for molecular property prediction with substructure masking. *Nature Communications*, 14:2585, 2023. doi: 10.1038/s41467-023-38192-3.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=ryGs6iA5Km.

Zhitao Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. GNNExplainer: Generating explanations for graph neural networks. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 9240–9251, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/d80b7040b773199015de6d3b4293c8ff-Abstract.html.

Hao Yuan, Jiliang Tang, Xia Hu, and Shuiwang Ji. XGNN: towards model-level explanations of graph neural networks. In Rajesh Gupta, Yan Liu, Jiliang Tang, and B. Aditya Prakash (eds.), *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, pp. 430–438. ACM, 2020. doi: 10.1145/3394486.3403085. URL https://doi.org/10.1145/3394486.3403085.

Hao Yuan, Haiyang Yu, Shurui Gui, and Shuiwang Ji. Explainability in graph neural networks: A taxonomic survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(5):5782–5799, 2023. doi: 10.1109/TPAMI.2022.3204236. URL https://doi.org/10.1109/TPAMI.2022.3204236.

Di Zhang, Wei Liu, Qian Tan, Jingdan Chen, Hang Yan, Yuliang Yan, Jiatong Li, Weiran Huang, Xiangyu Yue, Wanli Ouyang, Dongzhan Zhou, Shufei Zhang, Mao Su, Han-Sen Zhong, and Yuqiang Li. ChemLLM: A chemical large language model. *arXiv preprint arXiv:2402.06852*, 2024. URL https://arxiv.org/abs/2402.06852.

Shichang Zhang, Yozen Liu, Neil Shah, and Yizhou Sun. GStarX: Explaining graph neural networks with structure-aware cooperative games. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, volume 35, pp. 19810–19823, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/7d53575463291ea6b5a23cf6e571f59b-Abstract-Conference.html.

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

A VERIFIED INTERVENTIONS AND REPORTING SEMANTICS

This appendix specifies LAPCE's parent-wise molecular interventions and the statistics computed from their verified costs.

A.1 MOLECULAR INTERVENTION AND DISTANCE IMPLEMENTATION

Matching. RDKit (Landrum et al., 2025) matching uses `GetSubstructMatches` with `uniquify=True`, `useChirality=False`, and `maxMatches=100000`. Atoms match by element, formal charge, and aromaticity; bonds match by type and aromaticity. Returned index tuples are collapsed by deletion atom set, so symmetry-equivalent mappings induce only one tested operation. The reported runs contain no cap-hit pairs; the maximum observed returned set has 64 distinct matches. This operational set is $\mathcal{M}(G, s)$ in Section 3.

Deletion and residual processing. Each occurrence is deleted from the original parent, including its incident bonds. If multiple components remain, retain the one with the most heavy atoms, breaking ties by lexicographically smallest canonical SMILES. Empty and single-heavy-atom residuals are invalid. Apply full `Chem.SanitizeMol`, including valence and aromaticity checks, then `AssignStereochemistry(cleanIt=True, force=True)`. Failure returns $\perp$; there is no prediction-guided repair.

Component retention is part of τ. For an accepted occurrence, let $V_H(G)$ be the parent's heavy-atom set, M_H the matched heavy atoms, and $V_H(G')$ the retained heavy atoms mapped back to parent IDs. Then the operation's atom accounting is

$$n_{\text{matched}} = |M_H|, \qquad n_{\text{extra}} = |V_H(G) \setminus (M_H \cup V_H(G'))|,$$
$$n_{\text{removed}} = n_{\text{matched}} + n_{\text{extra}} = |V_H(G)| - |V_H(G')|.$$

Thus $n_{\text{extra}} \geq 0$. Candidate size counts the matched fragment, while total removal also includes components discarded by the fixed residual policy. WNode evaluates the complete processed residual. We report both deletion scales below and isolate component retention with the fixed-library control in Appendix A.5.

Occurrence-level removal accounting. For each parent with a finite top-20 best cost, one attaining match contributes to the summary. Matched, total removed, and extra heavy-atom counts refer to that same occurrence. The tables report all four evaluation tasks.

Task	Witnesses	Reach (%)	Mean matched	Mean total	Mean extra
HIV	1,178	89.99	6.1	6.6	0.5
Mutagenicity	1,683	85.09	5.3	5.7	0.4
BACE	196	41.88	6.6	7.1	0.5
TasteMolNet	1,453	53.03	5.8	6.2	0.4

Task	Med. matched	Med. total	Med. extra	Extra>0 (%)	P90 extra
HIV	6	6	0	14.7	1
Mutagenicity	5	5	0	11.9	1
BACE	6	6	0	15.3	2
TasteMolNet	5	5	0	12.8	1

The median extra removal is zero on every task. Per-occurrence counts satisfy $n_{\text{removed}} = n_{\text{matched}} + n_{\text{extra}}$; the displayed means are rounded separately. Of HIV's 1,178 finite witnesses, 1,131 have best cost at most 0.14 and 47 exceed it, consistent with the last sampled threshold in Figure 4.

WNode. The encoder is the frozen MolCLR-GIN pre-pooling output after its fifth message-passing layer, with 512-dimensional node states (Wang et al., 2022); the encoder configuration uses pre-training on 10 million unlabeled PubChem molecules. Node states use FP32 and L2 normalization; uniform node masses sum to one separately for each graph. The cosine cost matrix, marginals, and transport plan use FP64. POT (Flamary et al., 2021) `ot.emd`, with `numItermax=100000`

and `check_marginals=True`, solves balanced, unregularized transport. There is no entropic regularization or explicit solver-tolerance argument. After computing $D_{\mathrm{WNode}} = \sum_{u,v} T_{uv} C_{uv}$, check

$$\epsilon_{\mathrm{OT}} = \max\{\|T\mathbf{1} - a\|_\infty, \|T^\top \mathbf{1} - b\|_\infty\} \le 10^{-9}.$$

This is a post-solve feasibility check. Parent embeddings are cached; residual embeddings and duplicate-residual results are reused by canonical identity within the same encoder and preprocessing context. Numerical or resource failures are computation errors, not verified unsuccessful interventions.

A.2 Independent Match-Level Verification

Algorithm 2 implements Eqs. (1)–(2). A matrix is finalized only after the required computations complete. Its $+\infty$ entries then represent completed parent–candidate pairs without a sanitized strict flip. Completion status is stored separately from intervention costs. Selection reuses this matrix, and held-out evaluation starts after the ordering in Algorithm 1 is frozen.

Algorithm 2 Independent match-level verification in LAPCE

Require: Source-class parents $\mathcal{U}_y = \{G_i\}_{i=1}^N$, candidates $\mathcal{C} = \{s_j\}_{j=1}^P$, frozen GNN f_ϕ, fixed dist, source class y, fixed matching policy, and fixed deletion policy τ.
Ensure: Cost matrix $\mathbf{D} \in (\mathbb{R}_{\ge 0} \cup \{+\infty\})^{N \times P}$.

```
Initialize all D_ij ← +∞
for i = 1, ..., N do
    for j = 1, ..., P do
        for m ∈ M(G_i, s_j) do                ▷ Distinct returned deletion atom sets
            G' ← τ(G_i; s_j, m)               ▷ Start from the original parent each time
            if G' ≠ ⊥ then
                if ŷ_φ(G') ≠ y then
                    D_ij ← min{D_ij, dist(G_i, G')}
                end if
            end if
        end for
    end for
end for
return D
```

A.3 Coverage, Conditional Cost, and Budget Summaries

For a fixed cohort of N parents, let $d_i(S_k)$ be the minimum verified cost among candidates in frozen prefix S_k, and define $F_\theta(S_k) = N^{-1} \sum_i \mathbf{1}[d_i(S_k) \le \theta]$. Write $\mathcal{F}(S_k) = \{i : d_i(S_k) < \infty\}$. Coverage divides by all N parents. Conditional cost is the median on the nonempty set $\mathcal{F}(S_k)$, without an additional threshold filter. Coverage and capped cost remain defined for an empty finite set. For this same cohort,

$$F_\theta(S_k) > \tfrac{1}{2} \quad \Longrightarrow \quad \mathrm{median}_{i \in \mathcal{F}(S_k)}\, d_i(S_k) \le \theta.$$

Different methods can have different finite-success subsets, so coverage and conditional cost are interpreted jointly.

The selector study reports the equal-weight budget average

$$\mathrm{MeanCov}_{1:20} = \frac{1}{20} \sum_{k=1}^{20} F_{0.1}(S_k), \qquad \mathrm{Reach}\ @k = \frac{1}{N} \sum_i \mathbf{1}[d_i(S_k) < \infty],$$

and the capped mean

$$C_{\mathrm{cap}}@k = \frac{1}{N} \sum_i \min\{d_i(S_k), 0.1\}.$$

The normalized threshold area for $[a, b]$, $b > a$, is

$$A_\theta@20 = \frac{1}{b-a} \int_a^b F_\theta(S_{20})\, d\theta = \frac{1}{N(b-a)} \sum_{i : d_i(S_{20}) < \infty} [b - \max\{a, d_i(S_{20})\}]_+.$$

All compared selectors use $[a, b] = [0, 0.1]$ and the same cohort; therefore $A_\theta@20 = 1 - C_{\text{cap}}@20/0.1$. These are two parameterizations of one best-cost summary, not independent evidence. MeanCov uses all twenty integer budgets and the area uses best costs, not connecting lines in downsampled plots. Their evaluation scales differ from the nonuniform prefix weights and calibration-quantile thresholds in Eq. (11). $K_{\text{eff}} = |\pi_y|$ records the returned library size. Selection time excludes verification. The selector tables report point estimates, not component-wise bootstrap intervals.

A.4 Cross-Task Aggregation and Shared References

At $K = 20$, $\theta = 0.1$, let C_{db}, M_{db} denote coverage and conditional median cost on task d. The main comparison uses four tasks $\mathcal{T}$ and a five-baseline set $\mathcal{B}_{\text{main}}$ comprising GCFExplainer, GlobalGCE, COMRECGC, CM-CReM+, and RLHEX. With L denoting LAPCE, define

$$C_d^\star = \max_{b \in \mathcal{B}_{\text{main}}} C_{db}, \qquad M_d^\star = \min_{b \in \mathcal{B}_{\text{main}}} M_{db},$$

$$g_C = \frac{100}{|\mathcal{T}|} \sum_d \frac{C_{d,L} - C_d^\star}{C_d^\star}, \qquad g_M = \frac{100}{|\mathcal{T}|} \sum_d \frac{M_d^\star - M_{d,L}}{M_d^\star}.$$

The resulting macro improvements are 3.01% and 3.19%, with equal task weights and metric-specific references. RLHEX is the strongest coverage baseline on all four tasks. The exact covered-parent differences are 30, 53, 5, and 36 on HIV, Mutagenicity, BACE, and TasteMolNet, respectively, giving 2.29, 2.68, 1.07, and 1.31 percentage points. The exact-count macro coverage gain is $3.006415012649\ldots\%$. Cost references are RLHEX on HIV, Mutagenicity, and TasteMolNet, and CM-CReM+ on BACE. Using the source-precision inputs below gives an unrounded cost improvement of $3.191265136200\ldots\%$.

Task	LAPCE cost input	Baseline cost input	Cost reference
HIV	0.064468982	0.0665	RLHEX
Mutagenicity	0.062	0.0645	RLHEX
BACE	0.061696509	0.0621	CM-CReM+
TasteMolNet	0.0384	0.0405	RLHEX

Coverage ratios use exact counts and task denominators; cost ratios use retained source precision. Negative margins are retained, and relative margins require a positive reference. Coverage differences are percentage points; ratios calculated from four-decimal ablation costs are approximate. Ties receive the same rank; underline denotes the second distinct value. Descriptive counts are not ranked.

BACE's full proposal, backbone, main-comparison, and S6 references remain 39.10% coverage and 0.0617 cost. Within-study effects use their own matched controls. Native-element budgets do not equate atomic-edit, generation, or GNN-query budgets (Appendix D). Appendix F.1 uses the same five-baseline set for paired coverage inference; Appendix F.2 retains CM-CReM+ as its fixed conditional-cost comparator.

A.5 Fixed-Library Residual-Policy Control

This control holds the trained models, complete ordered candidate library, parent cohort, and thresholds fixed. It rejects a matched deletion when multiple residual components remain instead of retaining the largest component. Other validity checks are unchanged. Consequently, the admissible match set for every parent and prefix is a subset of the original set. With $\min \varnothing = +\infty$,

$$d_i^{\text{nd}}(S_k) \geq d_i(S_k), \qquad F_\theta^{\text{nd}}(S_k) \leq F_\theta(S_k), \qquad \text{Reach}^{\text{nd}}(S_k) \leq \text{Reach}(S_k).$$

Thus MeanCov and normalized threshold area cannot increase, while capped mean cost cannot decrease. Conditional medians use different finite subsets and have no corresponding monotonicity guarantee.

	BACE		Mutagenicity	
Metric	Full	No-drop	Full	No-drop
MeanCov (%)	34.34	33.21	73.33	71.82
A_θ	0.165	0.158	0.360	0.350
Cov@20 (%)	39.10	37.82	82.05	80.49
Covered parents	183	177	1623	1592
Reach@20 (%)	41.88	39.96	85.09	83.72
Finite parents	196	187	1683	1656
C_{cap}	0.0835	0.0842	0.0640	0.0650
C_{cond}	0.0617	0.0619	0.0620	0.0622

The control loses 6 covered parents on BACE and 31 on Mutagenicity, corresponding to 1.28 and 1.57 percentage points. Thus 177/183 (96.72%) and 1592/1623 (98.09%) of the originally covered parents remain covered, respectively, at $\theta = 0.1$. These retention fractions are conditional on original coverage, rather than full-cohort coverage percentages. Reach decreases by 9 and 27 parents. The comparison quantifies the effect of component retention with the trained models and selected libraries fixed.

A.6 Atom-Indexed Cross-Parent Case

Figure 1 displays the deletion of one nitro-group occurrence from each of three Mutagenicity parents. The three displayed interventions are evaluated by the frozen GNN used in the corresponding explanation experiment. The atom-indexed records below identify each tested occurrence, label flip, and parent–residual WNode distance.

Row	Parent atom tuple	WNode	Label
1	[10,9,11]	0.035765	$1 \to 0$
2	[9,8,10]	0.029782	$1 \to 0$
3	[0,1,2]	0.030863	$1 \to 0$

Indices are zero-based in the recorded parent order. Each displayed intervention removes three heavy atoms and yields the recorded sanitized residual without extra component removal. Its WNode is the cost of that exact occurrence, rather than the minimum from a different occurrence, and is below 0.1.

Structural identity and records. The deleted motif is `O=[N+][O-]`. Its nitrogen and singly bonded oxygen carry formal charges $+1$ and -1, respectively. The ordered atom tuples fix the correspondence to each parent and specify the displayed interventions. The three parent IDs, in display order, are `MUT_91856C6EA776E576AFD7`, `MUT_A15B49FCC6CBF7E9FE2C`, and `MUT_EF5671CF9B774B3160DD`. For the three cases, we retained the parent and residual SMILES, original atom indices, and occurrence costs in our experiment records.

Interpretation. The shared candidate has independently verified effects on all three parents. The measured response belongs to the frozen predictor and the stated processing policy; biological activity and synthetic accessibility are separate properties.

B CONTROLLED SELECTOR STUDY

The controls separate explicit cost, prefix utility, threshold aggregation, and three regularizers on fixed intervention evidence. Table 4 gives the primary outcomes.

B.1 FIXED EVIDENCE AND MATCHED CONTROLS

For each task, all variants share calibration and test parents, candidate pool, frozen GNN, verified matrices, and intervention policy. Let $L = |\pi_y|$ be the returned library length. Reporting uses $S_k = \{s_1, \ldots, s_{\min(k,L)}\}$, so a budget beyond L reuses the full returned prefix. All thirteen reported variants return $L = K_{\text{eff}} = 20$ on both tasks. For BACE and Mutagenicity, S6 is the LAPCE result in Figures 3 and 4. MeanCov and A_θ follow Appendix A.3.

The full utility and sequence objective are

$$\begin{aligned} U_6(S) &= F_\Theta(S) + \lambda_{\text{cost}} Q_{\text{cost}}(S) - \lambda_{\text{cov}} R_{\text{cov}}(S) \\ &\quad - \lambda_{\text{str}} R_{\text{str}}(S) - \lambda_{\text{size}} R_{\text{size}}(S), \\ J_6(\pi) &= \sum_{k=1}^{20} \rho_k U_6(S_k). \end{aligned}$$

S0–S5 are reduced coverage-based controls. S7–S10 respectively remove coverage overlap, structural similarity, candidate size, and explicit cost. S11 retains U_6 but uses $\rho'_k = 0$ for $k < 20$ and $\rho'_{20} = \sum_k \rho_k = 15$. S12 replaces only coverage aggregation by $F_{\text{single}} = 18\,\text{CCRCov}_{y,0.1}$, preserving the full prefix weights, cost term, regularizers, and original multi-threshold R_{cov} with normalized weights $\bar{w}_j = w_j/18$.

Table 5: Selector configurations. Regularizer switches are ordered as coverage overlap, structural similarity, and candidate size. S6–S12 share their initial ordering; S12 changes only coverage aggregation.

ID	Coverage	Objective	Refine	Cost	Reg. switches
S0	Single	Terminal	No	0	(0,0,0)
S1	Multi	Terminal	No	0	(0,0,0)
S2	Single	Terminal	Yes	0	(0,0,0)
S3	Multi	Terminal	Yes	0	(0,0,0)
S4	Single	Prefix	Yes	0	(0,0,0)
S5	Multi	Prefix	Yes	0	(0,0,0)
S6	Multi	Prefix	Yes	1	(1,1,1)
S7	Multi	Prefix	Yes	1	(0,1,1)
S8	Multi	Prefix	Yes	1	(1,0,1)
S9	Multi	Prefix	Yes	1	(1,1,0)
S10	Multi	Prefix	Yes	0	(1,1,1)
S11	Multi	Terminal	Yes	1	(1,1,1)
S12	Single	Prefix	Yes	1	(1,1,1)

S6–S12 share one initial ordering from multi-threshold greedy selection, fixed before changing any component. S12 does not recompute a single-threshold initialization. S0, S2, and S4 use single-threshold greedy initialization; S1, S3, and S5 use multi-threshold initialization. Matched variants retain the search neighborhood, proposal budget, tie breaking, tolerance, and all unablated coefficients.

Every output ordering is frozen before test evaluation. S11 may change both membership and order; it is not a fixed-membership permutation control.

B.2 Calibration and Refinement Settings

The maximum explanation budget is 20, the reporting threshold is $\theta^\star = 0.1$, and $d_{\text{cap}} = 0.1$. The thresholds are quantiles of finite calibration strict-flip costs, with quantile levels and weights

$$q = (0.05, 0.10, 0.20, 0.30, 0.50, 0.70, 0.90),$$
$$w = (4, 4, 3, 3, 2, 1, 1), \qquad \textstyle\sum_j w_j = 18, \qquad \bar{w}_j = w_j/18.$$

F_Θ uses unnormalized w, whereas R_{cov} uses $\bar{w}$. Coincident numerical thresholds are merged by summing their weights before normalization. Prefix weights are $\rho_k = 1$ for $k \le 10$ and $\rho_k = 0.5$ for $11 \le k \le 20$, with total 15; they differ from the equal weights used to evaluate MeanCov.

With coefficients ordered as $(\lambda_{\text{cost}}, \lambda_{\text{cov}}, \lambda_{\text{str}}, \lambda_{\text{size}})$, the configurations are

$$S6 : (0.20, 0.02, 0.01, 0.005), \quad S7 : (0.20, 0, 0.01, 0.005),$$
$$S8 : (0.20, 0.02, 0, 0.005), \quad S9 : (0.20, 0.02, 0.01, 0),$$
$$S10 : (0, 0.02, 0.01, 0.005).$$

S11 and S12 retain S6's coefficients and only make the substitutions specified in the preceding subsection. Candidate size is normalized by the largest heavy-atom count in the fixed full candidate pool. Greedy ties use ascending canonical candidate ID. Refinement considers swaps, relocations, and replacements, with proposal-order seed 7, at most 20,000 proposal evaluations, at most 50 accepted changes, and improvement tolerance 10^{-12}. All coefficients not targeted by an ablation remain fixed.

B.3 Detailed Outcomes and Interpretation

Tables 6 and 7 report coverage, finite reachability, both cost summaries, redundancy, candidate size, and selection time for all thirteen configurations. Within each matched comparison, the candidate pool and intervention evidence are fixed.

Across all thirteen variants, S6 has the highest coverage at each of the three reported budgets and the lowest redundancy and candidate size on both tasks. Its finite reachability also exceeds S10–S12. The cost and time columns show the accompanying operating trade-offs.

Table 6: Coverage and finite-action reachability of all thirteen selectors (percentages; higher is better). K_{eff} is the returned library size. All reported runs return 20 candidates; descriptive sizes are not ranked.

BACE						Mutagenicity					
ID	Cov@5	Cov@10	Cov@20	Reach	K_{eff}	ID	Cov@5	Cov@10	Cov@20	Reach	K_{eff}
S0	21.37	26.92	30.13	34.62	20	S0	53.59	66.73	71.84	75.43	20
S1	25.64	32.91	36.32	40.17	20	S1	60.87	74.62	78.82	82.05	20
S2	22.22	27.78	31.84	35.68	20	S2	54.75	68.00	72.90	76.59	20
S3	26.71	33.97	37.82	41.03	20	S3	61.88	75.94	80.13	83.42	20
S4	24.79	30.56	32.05	36.11	20	S4	58.04	71.18	73.46	77.05	20
S5	30.13	36.32	37.82	41.24	20	S5	65.82	78.61	80.43	83.72	20
S6	**31.20**	**37.18**	**39.10**	41.88	20	S6	**67.09**	**79.83**	**82.05**	85.09	20
S7	30.98	36.75	38.46	42.09	20	S7	66.63	79.27	81.70	85.44	20
S8	30.98	36.97	38.68	41.67	20	S8	66.94	79.52	81.95	85.04	20
S9	30.77	36.75	38.68	**42.31**	20	S9	66.48	79.02	82.00	**85.59**	20
S10	30.77	36.75	38.46	41.67	20	S10	66.28	79.22	81.75	84.93	20
S11	27.99	34.83	38.46	41.45	20	S11	63.20	77.20	81.80	84.83	20
S12	28.85	35.04	38.03	40.81	20	S12	62.79	76.79	80.79	84.18	20

Table 7: Cost, redundancy, candidate size, and selection time for all thirteen selectors (lower is better). Atoms is the mean candidate heavy-atom count, not total removal under the residual policy. Seconds excludes verification; best and second-best distinct values are marked within each task and metric.

BACE						
ID	C_{cond}	C_{cap}	CovRed	StructRed	Atoms	Sec.
S0	0.0714	0.0896	0.238	0.325	8.8	0.197
S1	0.0648	0.0848	0.216	0.305	8.5	0.256
S2	0.0702	0.0887	0.231	0.315	8.6	0.125
S3	0.0631	0.0840	0.213	0.302	8.4	**0.076**
S4	0.0695	0.0885	0.229	0.308	8.3	0.077
S5	0.0615	0.0834	0.208	0.295	8.2	0.082
S6	0.0617	0.0835	**0.180**	**0.250**	**7.0**	0.598
S7	**0.0613**	0.0833	0.226	0.260	7.2	0.690
S8	0.0618	0.0835	0.187	0.325	7.3	2.015
S9	0.0614	**0.0831**	0.192	0.266	8.5	0.590
S10	0.0623	0.0841	0.186	0.255	7.1	0.610
S11	0.0618	0.0836	0.198	0.266	7.3	0.575
S12	0.0620	0.0844	0.190	0.258	7.2	0.605

Mutagenicity						
ID	C_{cond}	C_{cap}	CovRed	StructRed	Atoms	Sec.
S0	0.0718	0.0735	0.368	0.316	6.9	**0.411**
S1	0.0652	0.0665	0.342	0.300	6.7	0.702
S2	0.0704	0.0724	0.359	0.310	6.8	1.054
S3	0.0634	0.0650	0.338	0.295	6.6	1.910
S4	0.0697	0.0720	0.350	0.303	6.6	1.100
S5	0.0618	0.0642	0.327	0.287	6.5	1.962
S6	0.0620	0.0640	**0.280**	**0.240**	**5.5**	19.564
S7	**0.0616**	0.0639	0.354	0.249	5.7	19.469
S8	0.0621	0.0640	0.291	0.317	5.8	19.604
S9	0.0617	**0.0637**	0.298	0.255	6.8	19.391
S10	0.0626	0.0648	0.287	0.244	5.6	19.300
S11	0.0620	0.0641	0.305	0.258	5.8	18.950
S12	0.0627	0.0653	0.294	0.248	5.7	19.200

Table 4 summarizes the explicit-cost, prefix, and threshold controls. Restoring explicit cost raises MeanCov by 0.53 and 0.73 percentage points on BACE and Mutagenicity, respectively. Against terminal-only S11, MeanCov rises from 31.79% to 34.34% and from 70.60% to 73.33%; against single-threshold S12, it rises by 2.14 and 2.53 points. The early-budget effects in Section 5.3 connect these outcomes to the first-covered-prefix expansion in Section 4.4.

An append-only step with earlier prefixes fixed has the same marginal maximizer under a common positive prefix multiplier. Sequence refinement instead evaluates changes across the ordering. The reduced controls illustrate this distinction: moving from terminal refinement S3 to prefix refinement S5 increases MeanCov by 2.37 and 2.76 points on BACE and Mutagenicity. Figure 5 shows their reported coverage checkpoints.

Restoring overlap control (S7 to S6) reduces CovRed from 0.226 to 0.180 on BACE and from 0.354 to 0.280 on Mutagenicity. Structural regularization (S8 to S6) reduces StructRed from 0.325 to 0.250 and from 0.317 to 0.240; size regularization (S9 to S6) reduces mean candidate heavy-atom counts from 8.5 to 7.0 and from 6.8 to 5.5.

B.4 Fingerprint Contract

Chemical fingerprint similarity. For R_{str}, sim_{chem} is Tanimoto similarity between radius-2, 2048-bit Morgan fingerprints represented by binary `ExplicitBitVect`s. We use RDKit `rdFingerprintGenerator.GetMorganGenerator` with

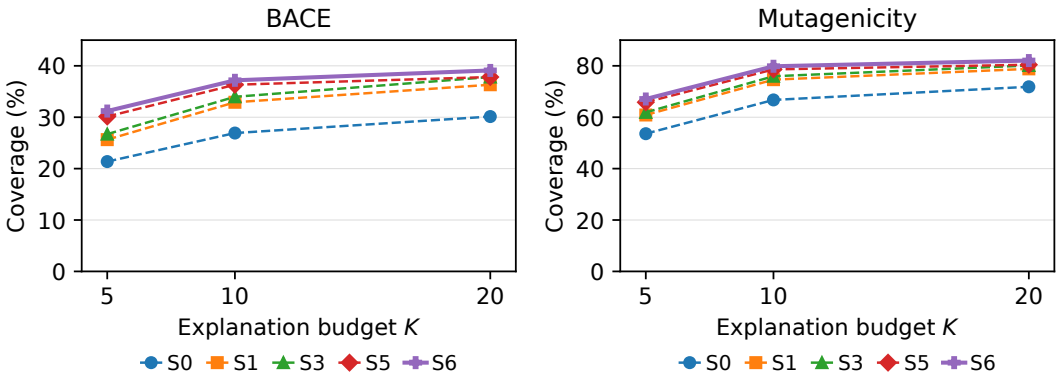


Figure 5: Selector coverage at $K \in \{5, 10, 20\}$. Points reproduce Table 6; lines are visual guides. S0/S1 use single/multi-threshold greedy selection, S3 adds terminal refinement, S5 prefix refinement, and S6 is the full selector.

`includeChirality=False`, `useBondTypes=True`, `includeRingMembership=True`, and `includeRedundantEnvironments=False`. Candidate fragments are sanitized and canonicalized; atom-map numbers are removed and dummy atoms are rejected. Formal charges and aromaticity are preserved. For the active-bit sets B_s, B_t,

$$\mathrm{sim}_{\mathrm{chem}}(s,t) = \begin{cases} |B_s \cap B_t| / |B_s \cup B_t|, & |B_s \cup B_t| > 0, \\ 0, & |B_s \cup B_t| = 0. \end{cases}$$

The nonempty-union calculation uses `DataStructs.TanimotoSimilarity`. This fragment-level structural penalty is distinct from WNode, which compares a complete parent and its processed residual using contextual node embeddings.

C Proposer Adaptation and Candidate Diagnostics

This appendix records the fixed proposal interface and adaptation settings, then distinguishes structural usability, intervention yield, and cross-parent reuse. The main candidate comparison is Table 3.

C.1 Prompt and Supervised Adaptation

The system and user message bodies below are fixed; model-specific chat serialization is separate. The response target is one connected fragment SMILES conditioned on the canonical parent and GNN-predicted source class.

System message.

```
You are a chemical subgraph proposer for counterfactual
explanation of a frozen molecular graph neural network.
Given a parent molecule and its predicted source class,
propose exactly one connected molecular subgraph that occurs
in the parent and is a plausible candidate for
deletion-based counterfactual testing.  Return only one
valid fragment SMILES. Do not predict the class, do not
explain your reasoning, and do not output any text other
than the fragment.
```

User message.

```
Task:  Propose one molecular subgraph candidate for
counterfactual explanation.
Parent molecule:
{canonical_parent_smiles}
Source class:
{source_class_label}
The candidate should be a connected substructure of the
parent.  Its removal will be tested independently by a
frozen GNN; you do not need to decide whether the prediction
will actually flip.
Response:
```

SFT targets come from BRICS and ring-preserving substituent fragments on non-test parents. A retained target is connected, contains 2–12 heavy atoms, directly matches its parent, and admits a sanitized residual under τ; neither a strict flip nor a WNode cutoff is required. Canonical targets are deduplicated per parent, and at most eight are retained, ordered by fewer heavy atoms, BRICS origin, then canonical SMILES. Target text is only the canonical fragment. A 90/10 split by original parent ID uses seed 42; pair totals need not exactly follow the parent ratio.

Task	Total pairs	Train pairs	Validation pairs
HIV	12,288	11,059	1,229
Mutagenicity	8,192	7,373	819
BACE	4,096	3,686	410
TasteMolNet	10,240	9,216	1,024
Total	34,816	31,334	3,482

Each task has its own adapted proposer. SFT uses Eq. (4), three epochs, AdamW learning rate 2×10^{-5}, $(\beta_1, \beta_2) = (0.9, 0.999)$, weight decay 0.01, per-device batch 16, gradient accumulation 4 (effective batch 64), maximum sequence length 512, warm-up ratio 0.05, a cosine schedule, gradient clipping 1.0, and seed 42. Test molecules do not supply SFT targets.

C.2 Frozen-GNN Feedback and Final Proposal Pool

PPO starts from SFT, keeping a frozen SFT reference. The optimization settings are:

Setting	Value
Learning rate	10^{-6}
Rollout batch / minibatch / PPO epochs	64 / 16 / 4
Policy clip / value clip	0.20 / 0.20
Discount / GAE parameter	1.0 / 0.95
KL coefficient / gradient norm	0.02 / 1.0
Rollout temperature / top-p	0.8 / 0.95
Maximum new tokens / samples per parent	96 / 4

Equation (5) uses $(\alpha_{\text{val}}, \alpha_{\text{sub}}, \alpha_{\text{cf}}, \alpha_{\text{size}}, \alpha_{\text{proj}}) = (1.0, 1.0, 2.0, 0.10, 0.50)$ and $\beta_{\text{KL}} = 0.02$.

Candidate processing and reward branches. Let u be a generated response and n_H the heavy-atom count, with $n_H(G) > 0$. The molecular feedback uses the following branches:

Path	r_{val}	r_{sub}	r_{cf}	Size argument	r_{proj}
Parse/connection failure	0	0	0	$r_{\text{size}} = 0$	0
Direct match	1	1	match maximum	raw fragment	0
Projection succeeds	1	0	match maximum	projected fragment	1
Projection fails	1	0	0	raw parsed fragment	1

For a retained direct or projected path,

$$r_{\text{size}} = \min\left\{1, \frac{n_H(s_{\text{eff}})}{n_H(G)}\right\}.$$

If parsing succeeds but both matching and projection fail, the same ratio uses the raw parsed fragment. A parse/connection failure has $r_{\text{size}} = 0$. The projection penalty marks invocation of the fallback, including unsuccessful projection. This is a parent-relative PPO penalty, not the pool-normalized selector penalty.

For each valid processed residual $G'_m = \tau(G; s_{\text{eff}}, m)$, let $\Delta p_y(m) = p_y(G) - p_y(G'_m)$. Then

$$r_{\text{cf}}(G, s_{\text{eff}}) = \begin{cases} \max\limits_{m \in \mathcal{M}_{\text{valid}}} [\Delta p_y(m) + \mathbf{1}[\hat{y}_\phi(G'_m) \neq y]], & \mathcal{M}_{\text{valid}} \neq \varnothing, \\ 0, & \mathcal{M}_{\text{valid}} = \varnothing. \end{cases}$$

The probability decrease and flip bonus come from the same occurrence. Equal feedback prefers a larger decrease, fewer total removed heavy atoms, then a lexicographically smaller atom-index tuple. PPO maximizes this feedback; verification instead minimizes WNode among strict-flip matches.

Sampled reference regularization. Let $\mathcal{T}(u)$ contain sampled response tokens after the prompt, excluding padding. A generated EOS token is included, as are the tokens actually sampled before truncation. Prompt tokens and padding-only mask tokens are excluded. With $L = |\mathcal{T}(u)|$, define

$$\widehat{k}_{\text{ref}}(u; G, y) = \begin{cases} \dfrac{1}{L} \displaystyle\sum_{t \in \mathcal{T}(u)} [\log q_\psi(a_t \mid h_t) - \log q_{\text{ref}}(a_t \mid h_t)], & L > 0, \\ 0, & L = 0. \end{cases}$$

A truly empty API response is a parse failure. An EOS-only generation has a sampled token and is handled by the nonempty branch even if its decoded fragment is empty. This token-mean sample statistic can take either sign; it supplies the reference-policy regularization used in the scalar feedback. Non-finite model log probabilities are numerical errors, not an empty response. The final scalar feedback is

$$R_{\text{prop}} = r_{\text{val}} + r_{\text{sub}} + 2r_{\text{cf}} - 0.10 r_{\text{size}} - 0.50 r_{\text{proj}} - 0.02 \widehat{k}_{\text{ref}}.$$

The reference contribution is included once, without an independent KL loss. The scalar reward and probability difference are not additionally clipped; the policy and value clips above remain unchanged. Advantages are centered and scaled per rollout batch with stabilizer 10^{-8}. The four responses per parent belong to PPO rollouts, not the final pool budget.

Each task uses 1,000 PPO updates, four PPO epochs per rollout, seed 42, and no early stopping. Checkpoints are assessed every 50 updates on non-test PPO validation parents, ranked by strict-flip yield, direct-match yield, then lower mean reference statistic; test coverage and cost do not select checkpoints.

Task	Adaptation parents	PPO validation parents
HIV	1,024	128
Mutagenicity	1,024	128
BACE	512	64
TasteMolNet	1,024	128

The clipped surrogate is (Schulman et al., 2017)

$$\mathcal{L}_{\text{clip}}(\psi) = \mathbb{E}_t\big[\min\{r_t^{\text{pol}}(\psi)\widehat{A}_t, \text{clip}(r_t^{\text{pol}}(\psi), 1-\epsilon, 1+\epsilon)\widehat{A}_t\}\big], \qquad r_t^{\text{pol}}(\psi) = \frac{q_\psi(a_t \mid h_t)}{q_{\psi_{\text{old}}}(a_t \mid h_t)}.$$

h_t is the generation history, a_t the sampled token, and $\widehat{A}_t$ the advantage based on regularized proposal feedback. The rollout policy differs from the frozen SFT reference; its sampled reference regularizer is included once through the reward.

After adaptation, the reference $R_0 = 8$ draws one response from each pair in $\{0.6, 0.8, 1.0, 1.2\} \times \{7, 19\}$ (temperature, seed), with `top_p=0.95`, `do_sample=True`, at most 96 new tokens, and repetition penalty 1.0. Canonicalization gives:

Task	Proposal parents	Responses	Unique pool	Calibration parents
HIV	256	2,048	1,742	144
Mutagenicity	256	2,048	1,588	256
BACE	128	1,024	812	153
TasteMolNet	256	2,048	1,616	311

Pool and calibration counts are distinct from explanation-test and candidate-diagnostic denominators.

C.3 Parent-Only Projection

A generated fragment without a direct match in its proposal parent uses fixed RDKit FMCS projection. Candidates are connected parent-supported common substructures with at least two heavy atoms. Atom comparisons use element, formal charge, and aromaticity; bonds use order and aromaticity. Chirality is ignored and `ringMatchesRingOnly=True`. Rank by decreasing heavy-atom count, decreasing bond count, then increasing canonical fragment SMILES. Reject attempts without an admissible projection. Neither a prediction change nor a test outcome selects the projection. This is a generation processing policy; projected candidates still undergo independent verification on each evaluation parent.

C.4 Candidate Yield and Cross-Parent Reuse

For source v, let $\mathcal{A}_v$ contain all raw attempts. Parse/conn. counts parseable connected outputs; Direct match counts outputs directly matching their proposal parents; Projection-free counts accepted attempts whose retained path never invokes projection. Valid residual counts attempts with at least one sanitized direct or allowed projected residual. Strict flip additionally requires a changed frozen-GNN class. All five rates divide by $|\mathcal{A}_v|$ and count an attempt once, even when multiple matches succeed. Unique is $100\,|\{\text{distinct canonical candidates}\}|/|\{\text{parseable connected attempts}\}|$.

Table 8: Expanded candidate diagnostics (percentages). The first five columns divide by raw attempts; Unique divides by parseable-and-connected attempts. Valid residual and Strict flip include allowed projection paths.

Source	Parse/conn.	Direct match	Projection-free	Valid residual	Strict flip	Unique
BRICS	**100.0**	**100.0**	**100.0**	**96.8**	5.8	72.4
ChemLLM-2B	83.1	49.6	45.3	77.2	4.7	81.6
ChemLLM-7B	90.7	62.8	58.9	84.6	7.1	**86.3**
7B + SFT	96.4	79.6	76.8	91.7	11.2	83.8
7B + SFT + PPO	96.1	81.3	79.4	92.5	**17.6**	82.7

Raw-attempt counts. Each source uses 1,024 raw attempts on 128 fixed BACE proposal parents with the same frozen BACE GINE. An attempt is counted once even if multiple matches succeed.

Source	Raw	Parse	Direct	P-free	Valid	Flip	Unique
BRICS	1024	1024	1024	1024	991	59	741
ChemLLM-2B	1024	851	508	464	791	48	694
ChemLLM-7B	1024	929	643	603	866	73	802
7B+SFT	1024	987	815	786	939	115	827
7B+SFT+PPO	1024	984	833	813	947	180	814

Unique divides by the parseable-and-connected count; the other five rates divide by 1,024. Candidate identity is canonical and source-specific. These counts are distinct from final-pool sizes, eligible-candidate counts, and the 468-parent BACE explanation cohort. For candidate s, let O_s contain all originating proposal-parent IDs and H_y be the held-out source class. Define

$$E_s = \{G \in H_y : G \notin O_s,\ \mathcal{M}(G, s) \neq \varnothing\},$$

$$r_{\text{reuse}}(s) = \frac{\sum_{G \in E_s} \mathbf{1}[d_y(G, s) < \infty]}{|E_s|}, \qquad r_{\text{transfer}}(s) = \frac{\sum_{G \in E_s} \mathbf{1}[d_y(G, s) \leq 0.1]}{|E_s|}.$$

For $\mathcal{C}_{\text{eligible}} = \{s : |E_s| > 0\}$, variant rates are candidate-macro averages, e.g. $\text{Reuse}_v = |\mathcal{C}_{\text{eligible}}|^{-1} \sum_{s \in \mathcal{C}_{\text{eligible}}} r_{\text{reuse}}(s)$. Table 9 reports the eligible and verified counts for all five sources. Eligibility is structural matching; verification additionally requires an eligible held-out strict flip.

Table 9: Cross-parent reuse for all five candidate sources. Eligible candidates match at least one non-origin held-out parent; verified candidates achieve at least one strict flip. Mean and median count successful eligible parents among verified candidates. Transfer adds WNode ≤ 0.1 on the same eligible cohort. Only percentage rates are ranked.

Source	Eligible	Verified	Mean parents	Median parents	Reuse (%)	Transfer (%)
BRICS	205	184	3.8	2	4.3	3.6
ChemLLM-2B	215	185	3.2	2	3.1	2.6
ChemLLM-7B	247	213	4.1	2	4.6	4.0
7B + SFT	361	327	6.2	4	7.2	6.4
7B + SFT + PPO	502	468	9.5	6	**11.3**	**10.1**

Reuse in Table 3 uses the same candidate-macro success rate; transfer adds the 0.1 cost cutoff on identical eligible cohorts, so transfer does not exceed reuse. From SFT to SFT+PPO, reuse rises from 7.2% to 11.3%, transfer from 6.4% to 10.1%, and the number of verified candidates from 327 to 468. Table 9 also reports the complete 2B and unadapted 7B comparisons. Projection-free describes retained generation paths; the following control tests their contribution through final-pool filtering.

C.5 Direct-Supported Pool Selection

This control retains canonical candidates with at least one directly matched originating proposal and removes those supported exclusively by projected origins. The same selector settings produce a new

ordering from this restricted pool. The proposer, explained GNN, evaluation cohort, and previously generated responses remain fixed.

	BACE		Mutagenicity	
Metric	Full	Direct-supported pool	Full	Direct-supported pool
MeanCov (%)	34.34	33.06	73.33	72.11
A_θ	0.165	0.157	0.360	0.352
Cov@20 (%)	39.10	37.61	82.05	81.04
Covered parents	183	176	1623	1603
Reach@20 (%)	41.88	40.38	85.09	84.02
Finite parents	196	189	1683	1662
C_{cap}	0.0835	0.0843	0.0640	0.0648
C_{cond}	0.0617	0.0618	0.0620	0.0621

For BACE, the twenty-budget numerator sums are 3,214 for the full pipeline and 3,094 for the direct-supported pool, giving MeanCov 34.34% and 33.06% and an unrounded difference of 1.28 percentage points. On Mutagenicity, MeanCov changes from 73.33% to 72.11%. At $K = 20$, coverage decreases by 7 and 20 parents, or 1.50 and 1.01 points on BACE and Mutagenicity, respectively; reach falls and capped cost rises on both tasks. These are measured outcomes after reselection, not metric-wise monotonicity guarantees from candidate-pool containment.

D BASELINE MOLECULAR REALIZATION

Every method retains its native explanation unit and supplies a concrete molecular endpoint for an original parent. Alternative successful realizations of one parent–element pair are summarized by minimum WNode after sanitization and a strict flip under the same frozen GNN. Elements are evaluated independently, not concatenated into cumulative edits. Calibration fixes libraries, orderings, and parameters before evaluation at $K = 20$, $\theta = 0.1$. Table 10 specifies the five baseline realizations; Appendix D.2 gives RLHEX's whole-molecule interface and adaptation settings.

Table 10: Native molecular realizations for the five baselines. Calibration fixes the ordering; K counts native elements, not equal numbers of edits or queries.

Method	Realization and calibration selection
GCFExplainer	Canonical complete counterfactual graphs are distinct elements. Apply the native parent-specific applicability gate before sanitization, strict-flip verification, and WNode scoring; retain minimum successful cost, with $+\infty$ for inapplicable pairs. Select at most 20 by marginal coverage at 0.1, then lower capped WNode and canonical ID; freeze the order.
GlobalGCE	One mapping is one element. Apply each at an atom/bond-consistent occurrence independently on the original parent, sanitize, require a strict flip, and retain minimum WNode. Do not compose mappings.
COMRECGC	One selected cluster is one element. Enumerate its frozen endpoint support, recompute native pair/recourse compatibility, and retain minimum WNode over sanitized strict flips (Appendix D.1).
CM-CReM+	Mask the top-1 attributed region and generate at most 32 context-compatible replacements, reattaching at original points and bond types. Retain minimum WNode over valid strict flips; merge canonical replacement/attachment identities and select at most 20 by marginal coverage at 0.1, then capped WNode and canonical ID.
RLHEX	Generate complete molecules using a frozen PS-VAE and PPO latent adapter. Canonicalize valid non-source endpoints; apply fingerprint-support eligibility $d_{\mathrm{fp}} \leq 0.87$ to each parent–endpoint pair in calibration and test. Select at most 20 by gated WNode coverage at 0.1, then capped WNode and canonical ID; freeze the order (Appendix D.2).

GCFExplainer, GlobalGCE, and COMRECGC use the authors' WSDM 2023, TMLR 2025, and ICML 2025 implementations with the adapters specified here. CM-CReM+ combines Counterfactual Masking and CReM 0.2.13, sampling at most 32 replacements per parent. Baseline runs use seed 7. Completed invalid or non-flipping realizations yield $+\infty$; numerical or resource failures are treated as computation errors rather than intervention outcomes. All reported results use completed evaluations.

GCFExplainer applicability. Let $a_{\mathrm{GCF}}(G, H_j)$ be the native adapter's parent-specific applicability predicate for complete counterfactual graph H_j. An inapplicable pair has cost $+\infty$; an applicable pair contributes its successful molecular realization costs after sanitization and a strict flip, with the minimum retained. Thus GCFExplainer's finite-success subset is parent-dependent. Sharing a complete-graph element type does not imply identical applicability.

Scope of the GlobalGCE comparison. We evaluate one independently applied GlobalGCE mapping per native element under the common parent–element cost interface. Mapping composition and sequential edits are outside this benchmark and are not covered by its reported results.

D.1 COMRECGC ENDPOINT-SUPPORTED RECOURSES

Element and support. One selected common-recourse cluster is one element (Fournier & Medya, 2025). Native search on non-test data produces a finite counterfactual-graph pool. Fixed graph-to-molecule conversion, stereochemistry, canonicalization, sanitization, and canonical-SMILES deduplication form $\mathcal{H}_{\mathrm{freeze}}$, retaining $(\text{parent ID}, \text{endpoint ID}, \text{cluster ID})$ provenance. Cluster j

stores centroid μ_j, selection rank, and endpoint support $\mathcal{H}_j$ from its accepted calibration pairs. A single endpoint can support several clusters because membership is pair-specific.

Native representation and compatibility. A task-specific frozen NeuroSED encoder supplies h_{rec} and D_{NeuroSED}, distinct from WNode's MolCLR encoder. With two PyG directed edges per chemical bond,

$$e(G) = |V(G)| + \tfrac{1}{2}|E_{\text{PyG}}(G)|, \qquad s(G,H) = e(G) + e(H),$$
$$d_{\text{native}}(G,H) = D_{\text{NeuroSED}}(G,H)/s(G,H), \quad \nu(G,H) = [h_{\text{rec}}(H) - h_{\text{rec}}(G)]/s(G,H).$$

No further unit-vector normalization is applied. After the importance and native-pair gates, Euclidean DBSCAN clusters calibration displacements. Centroids are cluster means; noise is discarded. Retained clusters have $\|\mu_j\|_2 < 0.10$, and accepted pairs have $\|\nu(G,H) - \mu_j\|_2 < 0.02$.

New-parent execution. For source-class parent G, enumerate only $H \in \mathcal{H}_j$ and recompute compatibility:

$$\mathcal{A}_j(G) = \{H \in \mathcal{H}_j : d_{\text{native}}(G,H) \le 0.10,\ \|\nu(G,H) - \mu_j\|_2 < 0.02,$$
$$\text{MolValid}(H) = 1,\ \hat{y}_\phi(H) \neq y\},$$
$$d_{\text{Com}}(G, r_j) = \min_{H \in \mathcal{A}_j(G)} D_{\text{WNode}}(G,H), \qquad \min \varnothing = +\infty.$$

Endpoints undergo sanitization without repair. TasteMolNet uses its original three-class GNN and requires Bitter or Tasteless instead of Sweet. The pair gate, cluster radius, and WNode reporting threshold have distinct roles, even when values coincide. WNode's 0.1 cutoff applies only to final coverage, not admissibility or conditional cost.

Selection and freezing. Let C_j be calibration parents in cluster j's accepted member pairs. Greedy ordering maximizes $|C_j \setminus \bigcup_{\ell \in R} C_\ell|$. The adapter breaks ties by lower centroid norm, then ascending cluster ID. It constructs at most 100 clusters and reports the first at most 20. Calibration freezes endpoint supports, encoder, normalization, filters, clusters, centroids, radius, ordering, molecular policy, GNN, and WNode. Test evaluation performs support enumeration and verification only, with no endpoint generation, reclustering, threshold tuning, or reselection. Parent prefix cost is $\min_{j \le k} d_{\text{Com}}(G, r_j)$.

Implementation settings (shared across tasks).

Setting	Value
Native encoder	NeuroSED `NormGEDModel(8,F,64,64)`, where F is the task feature dimension; frozen `data/<task>/neurosed/best_model.pt`, for HIV, Mutagenicity, BACE, or TasteMolNet
Search steps / heads / teleport	50,000 / 5 / 0.10
Candidate / neighbor caps	100,000 (`k`) / 10,000 (`sample_size`)
Candidate pre-filter	`importance_parts[0]` ≥ 0.5
Native pair gate	$d_{\text{native}} \le 0.10$
DBSCAN	Euclidean; `eps`=0.02; `min_samples`=3; discard noise -1
Post-filters	$\|\nu - \mu_j\|_2 < 0.02$; $\|\mu_j\|_2 < 0.10$
Native order / benchmark prefix	At most 100 / first at most 20 clusters
Adapter ties	Lower centroid norm, then ascending cluster ID
Molecular processing	Fixed benchmark conversion; canonical SMILES; `Chem.SanitizeMol`; no repair
WNode threshold / seed	0.10 / 7 for NumPy, Python random, and PyTorch

D.2 RLHEX WITH FINGERPRINT-SUPPORTED ENDPOINTS

Generator and data roles. Our benchmark realization of RLHEX (Wang et al., 2024) uses the released ZINC250K PS-VAE with its encoder and decoder frozen, and a PPO-trained latent adapter. The generator has latent dimension 56 and graph hidden dimension 400. The RLHEX implementation's `TransformerNN` actor and critic consume the 400-dimensional graph representation and

output a 56-dimensional action mean and a scalar value, respectively. Each task uses its frozen GINE and molecular feature pipeline. Adaptation supplies reward-reference parents, validation selects the checkpoint, and the final proposal and calibration roles supply generation and ordering. Explanation-test parents are used only for the frozen library.

Task	Adaptation	Validation	Proposal	Calibration	Test
HIV	1,024	128	256	144	1,309
Mutagenicity	1,024	128	256	256	1,978
BACE	512	64	128	153	468
TasteMolNet	1,024	128	256	311	2,740

RLHEX does not use the LAPCE parent–fragment SFT pairs. For TasteMolNet, the source is Sweet; the smooth probability term is $p_{\neg y}(H) = p_{\text{Bitter}}(H) + p_{\text{Tasteless}}(H)$. Endpoint acceptance separately requires the original three-class GNN's argmax to be Bitter or Tasteless.

Latent action and optimization. The observation/action modes are `emb`/`z`. With the encoder's latent mean and scale $\mu(G), \sigma(G)$, the policy samples

$$a_t \sim \mathcal{N}(\mu_\vartheta(h_G), 0.5I), \qquad z' = \mu(G) + \sigma(G) \odot \epsilon + a_t, \qquad \epsilon \sim \mathcal{N}(0, I).$$

The offset is added once. The covariance is $0.5I$, giving coordinate standard deviation $\sqrt{0.5}$; final generation uses the frozen stochastic policy rather than substituting its mean. The actor uses linear warm-up for the first 5% of the transition budget followed by linear decay to 10^{-6}; the critic learning rate stays constant. Validation ranks checkpoints by coverage, then lower capped cost, then earlier checkpoint. Training-parent sampling uses the implementation's UCB controller and training reward history only. The shared optimizer and generation settings are listed below.

Setting	Value
Actor / critic learning rates	10^{-5} / 5×10^{-6}
Optimizer	Adam; $(\beta_1, \beta_2) = (0.9, 0.999)$; $\epsilon = 10^{-8}$; weight decay 0
Policy clip / discount	0.2 / 0.99
Rollout / PPO updates	64 transitions / one update per rollout
Training episode / total budget	At most one transition / 10^6 transitions per task
Actor warm-up / final rate	First 50,000 transitions / 10^{-6}
Reward mode / weights	Absolute score; non-source probability 1, fingerprint coverage 10
Fingerprint threshold	$d_{\text{fp}} \le 0.87$
Generation trajectory / outputs	At most 20 steps / 10 decodes per step
Decoder	Temperature 1.0; maximum 60 atoms; edge threshold 0.5
Candidate pool / library caps	1,024 / 20 unique complete endpoints
Reference seed	7
Benchmark parent gate	$d_{\text{fp}} \le 0.87$ in calibration and test; WNode report threshold 0.1

Training feedback. Training uses binary radius-2, 2048-bit Morgan fingerprints, with the preprocessing and bit-vector settings in Appendix B.4. Here $v(H)$ denotes molecular validity; strict-flip acceptance is a separate final-pool criterion. For $d_{\text{fp}} = 1 - \text{Tanimoto}$, define

$$F_{\text{fp}}(H) = \frac{1}{N_{\text{adapt}}} \sum_{G \in D_y^{\text{adapt}}} \mathbf{1}[d_{\text{fp}}(G, H) \le 0.87], \qquad s(H) = \mathbf{1}[v(H)]\big[p_{\neg y}(H) + 10F_{\text{fp}}(H)\big].$$

The absolute-score reward averages scores over the valid molecules among 10 decoder outputs; a step with none valid has reward zero. All transitions remain in the rollout. Advantages use Monte-Carlo return minus the critic, standardized within the batch with stabilizer 10^{-10}. No entropy, LAPCE fragment-size, projection, or reference-policy term is added. The training fingerprint threshold is distinct from the final WNode threshold.

Generation and calibration. Each proposal parent starts one trajectory using the generation budgets and decoding settings above.

The next state is the first legal decoder-order output meeting the native atom/bond conditions. Separately, at most one highest-score valid strict-flip molecule is exported per step, with canonical-ID tie breaking; all steps may contribute. A step without an admissible next state terminates the trajectory. Canonical whole-molecule deduplication retains origin metadata. If the pool exceeds 1,024 endpoints, it is truncated by non-test score, then canonical ID.

Using the support-gated costs defined below, calibration greedily maximizes marginal WNode close coverage at $\theta = 0.1$, breaking ties by lower capped mean WNode and then canonical ID. Zero-coverage-gain steps use the same tie rules until $L = \min(20, P)$ distinct endpoints are selected. This is terminal-coverage selection, not LAPCE's multi-threshold prefix objective. These budgets are caps on native operations, not equal computation budgets across methods.

Endpoint validity and held-out evaluation. An endpoint must parse, be connected, contain at least two heavy atoms, pass full sanitization and the shared feature interfaces, and have a non-source GNN prediction. Multi-component outputs are rejected without component retention or prediction-guided repair. The ordered library $\pi_{\mathrm{RL}} = (H_1, \ldots, H_L)$ is frozen before test evaluation. A parent–endpoint pair is eligible when its native-proximity fingerprint support passes the fixed benchmark gate:

$$d_{\mathrm{fp}}(G, H) = 1 - \mathrm{Tanimoto}\big(\mathrm{Morgan}_{r=2,2048}(G), \mathrm{Morgan}_{r=2,2048}(H)\big),$$
$$a_{\mathrm{RL}}(G, H) = \mathbf{1}[d_{\mathrm{fp}}(G, H) \le 0.87].$$

The endpoint is still a complete molecule; the gate specifies which parents it can serve in this benchmark realization. The same gate is applied in calibration and held-out evaluation. Its fingerprint representation is the one used for training feedback, whereas final intervention cost is WNode:

$$d_{\mathrm{RL}}(G, H) = \begin{cases} D_{\mathrm{WNode}}(G, H), & a_{\mathrm{RL}}(G, H) = 1,\ v(H) = 1,\ \hat{y}_\phi(H) \neq y, \\ +\infty, & \text{otherwise.} \end{cases}$$

$$d^{\mathrm{RL}}_{i,k} = \min_{j \le \min(k, L)} d_{\mathrm{RL}}(G_i, H_j), \qquad \min \varnothing := +\infty.$$

The gate and ordering are fixed across all reported K and θ. No subgraph deletion, endpoint composition, or test-time reselection is used. WNode's encoder, exact-EMD solver, and numerical checks follow Appendix A.1. Conditional cost is the median over parents with a finite prefix cost, without a WNode-threshold filter; coverage divides by the complete explanation cohort. The 0.87 fingerprint gate and 0.1 WNode reporting threshold operate in different spaces.

Task	N	Covered at 0.1	Finite parents	Reach (%)	C_{cond}
HIV	1309	970	1160	88.62	0.0665
Mutagenicity	1978	1570	1700	85.95	0.0645
BACE	468	178	200	42.74	0.0632
TasteMolNet	2740	1310	1500	54.74	0.0405

Finite support can grow with the prefix, so conditional-median monotonicity is not imposed. At $K = 20$, costs 0.0665, 0.0645, 0.0632, and 0.0405 are consistent with the threshold counts and the corresponding finite subsets. The BACE and TasteMolNet coverages below 50% therefore coexist with conditional medians below 0.1.

E DATASETS, FROZEN GNNS, AND EVALUATION COHORTS

Table 1 counts unique canonical molecules after task-specific filtering, with atoms and bonds summed over each view and each chemical bond counted once. Source shares correspond to HIV-active, mutagenic, BACE-active, and Sweet. The explanation-test cohorts contain 1,309, 1,978, 468, and 2,740 GNN-predicted source-class parents, respectively, and are shared by all methods within a task. Classification-test membership is defined separately and is neither the coverage denominator nor assumed to contain, or be contained in, the explanation cohort.

SFT uses parent–fragment pairs split by parent identity; PPO uses adaptation and validation parents for feedback and checkpoint selection. Final proposal generation and calibration use disjoint non-test parent sets, while explanation-test parents evaluate frozen libraries. Counts and settings appear in Appendices C.1–C.2; RLHEX's corresponding data roles appear in Appendix D.2. Parent counts, supervised pair counts, classification-test counts, and latent-policy transitions are distinct reporting units.

E.1 PREDICTIVE DIAGNOSTICS

Table 11 evaluates the frozen checkpoint associated with each task's explanation experiment. Accuracy is hard-label accuracy. For binary tasks, balanced accuracy averages the two recalls and macro-F1 averages class-specific F1 scores; positive labels are HIV-active, mutagenic, and BACE-active. AUROC and AP use prediction scores. TasteMolNet uses class order Bitter, Sweet, Tasteless, macro one-vs-rest AUROC/AP, and three-class macro-F1.

Table 11: Frozen-GNN classification-test performance. N_{test} is not the explanation-coverage denominator. Accuracy, balanced accuracy, and macro-F1 are percentages.

Task	N_{test}	Acc.	Bal. Acc.	AUROC	AP	Macro-F1
HIV	8,224	96.41	79.22	0.918	0.536	78.32
Mutagenicity	849	81.74	81.53	0.889	0.902	81.61
BACE	303	79.87	79.26	0.867	0.856	79.35
TasteMolNet	2,684	76.64	74.28	0.881	0.773	74.86

The binary confusion matrices, with rows (True 0, True 1) and columns (Pred 0, Pred 1), are

$$\text{HIV}: \begin{bmatrix} 7721 & 159 \\ 136 & 208 \end{bmatrix}, \qquad \text{Mutagenicity}: \begin{bmatrix} 311 & 87 \\ 68 & 383 \end{bmatrix}, \qquad \text{BACE}: \begin{bmatrix} 145 & 29 \\ 32 & 97 \end{bmatrix}.$$

They reproduce the reported hard-label metrics after rounding; AUROC and AP are computed separately from scores.

E.2 BACKBONE CONFIGURATION

The backbone study includes GINE (Hu et al., 2020), GIN (Xu et al., 2019), GatedGCN† (Bresson & Laurent, 2017), GATv2 (Brody et al., 2022), and GCN (Kipf & Welling, 2017). GatedGCN†, our adapted configuration, is bond-aware, with learned edge-state updates, residual connections, and batch normalization. It has five layers, 256-dimensional node and edge states, ReLU, dropout 0.10, and mean pooling, with no virtual node or positional encoding. All explained GNN parameters remain fixed during explanation.

F Bootstrap Intervals and Sensitivity

These analyses quantify sampling uncertainty for the frozen comparison and local sensitivity around the reported algorithmic configuration. They complement systematic graph-explanation evaluation (Amara et al., 2022; Agarwal et al., 2023; Zheng et al., 2024).

F.1 Paired-Parent Bootstrap

At $K = 20$ and $\theta = 0.1$, Table 12 reports LAPCE's coverage and conditional-cost intervals and its paired coverage margin against the same five baselines as Table 2:

$$\mathcal{B}_5 = \mathcal{B}_{\text{main}} = \{\text{GCFExplainer}, \text{GlobalGCE}, \text{COMRECGC}, \text{CM-CReM+}, \text{RLHEX}\}.$$

The analysis uses 1,000 paired-parent percentile bootstrap replicates with seed 7 and the existing task-specific resampling index matrices. Each replicate draws N evaluation-parent IDs with replacement in the original cohort indexing and applies the identical multiset to all six methods. Parent multiplicities are retained. GNNs, LLMs, candidate libraries, and orderings remain frozen.

Table 12: 95% paired-parent bootstrap intervals (1,000 replicates, seed 7). Coverage and cost describe LAPCE; the margin compares LAPCE with the strongest of GCFExplainer, GlobalGCE, COMRECGC, CM-CReM+, and RLHEX within each paired replicate. Coverage is in percent and margins are in percentage points.

Task	LAPCE Cov. 95%	LAPCE Cost 95%	Margin vs. $\mathcal{B}_5$
HIV	[74.102, 78.688]	[0.0629, 0.0662]	[+1.299, +3.361] pp
Mutagenicity	[80.233, 83.721]	[0.0603, 0.0638]	[+1.871, +3.539] pp
BACE	[34.829, 43.376]	[0.0585, 0.0652]	[+0.214, +2.137] pp
TasteMolNet	[47.263, 51.023]	[0.0366, 0.0401]	[+0.803, +1.825] pp

For percentage-valued replicate coverages, define

$$C^{(b)}_{\text{base5}} = \max_{m \in \mathcal{B}_5} C^{(b)}_m, \qquad \Delta C^{(b)}_5 = C^{(b)}_{\text{LAPCE}} - C^{(b)}_{\text{base5}}.$$

The maximum is recomputed within every paired replicate, not fixed to a baseline in advance. RLHEX attains this maximum in all 1,000 replicates on each task. The margin interval contains the 2.5th and 97.5th percentiles of $\Delta C^{(b)}_5$, not differences between marginal interval endpoints. LAPCE's conditional cost is the median of finite best costs in the sampled multiset, without a further $\theta = 0.1$ filter. All 1,000 cost resamples have nonempty finite-success subsets; none is omitted. Completed failures remain $+\infty$, whereas unfinished computations are not recoded as failures.

The five-baseline coverage-margin intervals are [+1.299, +3.361], [+1.871, +3.539], [+0.214, +2.137], and [+0.803, +1.825] percentage points on HIV, Mutagenicity, BACE, and TasteMolNet, respectively. All four lower endpoints are positive; BACE is closest to zero. The exact point margins, from covered-parent differences of 30, 53, 5, and 36, lie within their respective intervals. These intervals quantify parent-sampling uncertainty conditional on the fitted models and frozen libraries, not variation from retraining, regeneration, or reselection.

F.2 Paired Conditional-Cost Difference

CM-CReM+ is the fixed cost comparator. For a common parent resample, each method contributes the median of its own finite best costs and

$$\Delta M^{(b)} = M^{(b)}_{\text{CM-CReM+}} - M^{(b)}_{\text{LAPCE}}.$$

Positive values favor LAPCE. The protocol uses 1,000 paired-parent resamples and seed 7, with frozen models and libraries and without a coverage-threshold filter on the finite costs. Both conditional medians require nonempty finite-success subsets. The interval is computed from paired median differences, with the same comparator in every replicate.

Task	$M_{\text{CM-CReM+}} - M_{\text{LAPCE}}$	Paired 95% interval
HIV	$+0.0023$	$[+0.0011, +0.0036]$
Mutagenicity	$+0.0035$	$[+0.0021, +0.0049]$
BACE	$+0.0004$	$[-0.0004, +0.0012]$
TasteMolNet	$+0.0022$	$[+0.0012, +0.0032]$

These intervals concern conditional medians on each method's own finite-success subset. The fixed comparator is CM-CReM+; the analysis does not test the lowest-cost baseline selected separately for each task in Table 2. BACE's interval includes zero despite its lower LAPCE point estimate. In contrast, Table 12 compares coverage with the strongest of all five baselines in each replicate.

F.3 Proposal-Budget and Regularization Sensitivity

Table 13 changes one factor at a time. The coefficient bases are $(0.01, 0.02, 0.005)$ for structural, overlap, and size penalties. The respective tested triples are

$$(0.005, 0.01, 0.02), \qquad (0.01, 0.02, 0.04), \qquad (0.0025, 0.005, 0.01).$$

All other coefficients remain at reference. Proposal settings are 4, 8, and 16 raw responses per parent around $R_0 = 8$, separate from four samples per PPO rollout. Each budget uses its specified decoding schedule; the reference schedule appears in Appendix C.2.

Table 13: One-factor BACE sensitivity. The four $1.0\times$ rows share one reference and are not independent repetitions. Ranking is within each factor and metric.

Factor	Relative setting	Cov@20↑	Cost@20↓
R/R_0	$0.5\times$	35.90	0.0647
	$1.0\times$	<u>39.10</u>	<u>0.0617</u>
	$2.0\times$	**39.32**	**0.0613**
$\lambda_{\text{str}}/\lambda_{\text{str},0}$	$0.5\times$	**39.10**	<u>0.0618</u>
	$1.0\times$	**39.10**	**0.0617**
	$2.0\times$	<u>38.03</u>	0.0620
$\lambda_{\text{cov}}/\lambda_{\text{cov},0}$	$0.5\times$	**39.10**	0.0619
	$1.0\times$	**39.10**	**0.0617**
	$2.0\times$	<u>38.25</u>	<u>0.0618</u>
$\lambda_{\text{size}}/\lambda_{\text{size},0}$	$0.5\times$	**39.10**	0.0619
	$1.0\times$	**39.10**	<u>0.0617</u>
	$2.0\times$	<u>38.03</u>	**0.0615**

From $0.5R_0$ to R_0, coverage increases by 3.21 percentage points and cost decreases from 0.0647 to 0.0617. From R_0 to $2R_0$, the additional coverage gain is 0.21 points and cost reaches 0.0613. Thus the measured additional gain is smaller at the larger budget. Across regularizer perturbations, coverage ranges from 38.03% to 39.10% and cost from 0.0615 to 0.0620, remaining within 1.07 percentage points and 0.0003 of the reference. These perturbations characterize the tested local neighborhood.

G Computational Cost and Implementation Accounting

G.1 Timing Environment and Stage Results

This appendix reports LAPCE's stage times and graph-level evaluation counts. Adaptation is reported separately from the five sequential explanation stages, whose elapsed times sum to

$$T_{\text{sum}} = T_{\text{proposal}} + T_{\text{parse/dedup}} + T_{\text{cal-verify}} + T_{\text{select}} + T_{\text{heldout-verify}}.$$

The environment has one NVIDIA A800 80GB PCIe GPU, 32 vCPUs on an AMD EPYC 7T83 host, and a 480 GiB RAM limit. Versions are Python 3.10.16, CUDA 11.8, PyTorch 2.7.1+cu118, and RDKit 2025.03.5. LLM inference uses BF16; GNN/encoder inference uses FP32; OT uses FP64. Proposal, residual-GNN, and encoder batch sizes are 8, 256, and 128, with eight preprocessing workers. Parent/residual embedding and duplicate residual caches are enabled. One untimed warm-up batch precedes each stage; stages do not overlap. Times describe one production run.

Table 14: Stage-wise elapsed times (seconds). The final row sums the five sequential explanation stages; adaptation is separate. Main-pipeline selection and controlled-selector refinement retain their distinct timing scopes.

Stage	HIV	Mutagenicity	BACE	TasteMolNet
SFT adaptation	14,760	10,980	9,420	13,860
PPO adaptation	31,840	22,760	19,680	29,940
Proposal generation	7,420	2,180	1,160	4,760
Parsing/deduplication	214	73	42	137
Calibration verification	18,640	5,860	3,120	10,940
Selection	0.84	0.69	0.61	0.93
Held-out verification	11,920	3,640	1,940	7,180
Explanation-stage sum (derived)	38,194.84	11,753.69	6,262.61	23,017.93

Calibration and held-out verification account for 80.01%, 80.83%, 80.80%, and 78.72% of the task sums, identifying verification as the main measured bottleneck. Selection takes 0.84, 0.69, 0.61, and 0.93 seconds, less than 0.01% of each sum. The final table row is a sum of the five stage timers.

Timer boundaries. Table 7 measures controlled selector execution after matrix loading, including construction of variant-specific selector state and refinement. Table 14 measures the production selection call with reusable inputs already available, plus ordering serialization. Both exclude molecular verification. The explanation-stage total is the sum of the five stated timers; it is not a cold-start measurement unless all input preparation is included in those boundaries. Thus 19.564 seconds in the Mutagenicity controlled study and 0.69 seconds in the production pipeline have different accounting scopes.

G.2 Residual Evaluation Counts

The following counters count graph-level residual evaluations, excluding PPO feedback and parent-graph evaluations:

Task	Calibration residuals	Held-out residuals	Total
HIV	158,432	96,870	255,302
Mutagenicity	48,960	31,120	80,080
BACE	25,344	15,872	41,216
TasteMolNet	92,160	60,480	152,640

Logical calibration parent–candidate counts are 250,848, 406,528, 124,236, and 502,576, from the reported calibration sizes and production pools. One logical pair can have multiple residual realizations, so this pair count differs from both residual evaluations and batched calls.

G.3 Logical Work and Storage

For one source class with N calibration parents and P candidates, let $\mathcal{H}$ be the enumerated parent–candidate–matched-atom-set triples, $\mathcal{V} \subseteq \mathcal{H}$ those with valid residuals, and $\mathcal{F} \subseteq \mathcal{V}$ the strict flips. After model fitting,

$$T_{\text{build}} = T_{\text{proposal}} + T_{\text{parse/dedup}} + \sum_{i=1}^{N}\sum_{j=1}^{P} T_{\text{match}}(G_i, s_j) + \sum_{h\in\mathcal{H}} T_{\text{delete/sanitize}}(h) + \sum_{h\in\mathcal{V}} T_{\text{GNN}}(h) + \sum_{h\in\mathcal{F}} T_{\text{distance}}(h).$$

Distance includes contextual node encoding and transport. Batching, parent-embedding reuse, and duplicate-residual caching reduce repeated work. Without reuse, each valid residual needs one graph-level GNN evaluation, plus up to N uncached parent evaluations; this excludes PPO feedback. Graph evaluations are not batched forward calls. The dense matrix uses $O(NP)$ storage, apart from match traces and pairwise candidate diagnostics.

Selection needs no further calibration GNN or molecular-encoder queries once the matrix is available. For Q objective proposals, $T_{\text{select}} = T_{\text{precompute}} + \sum_{q=1}^{Q} T_{\text{score},q}$. Cost depends on thresholds, length, regularizers, and caching; Q is not a training-iteration count. Held-out verification remains a separate stage. Bounded proposal sampling does not remove molecular matching or finite sequence-optimization costs.