CAN SUBGRAPH EXPLANATIONS BE WEAPONIZED TO STEAL GRAPH NEURAL NETWORKS?

Anonymous authors
Paper under double-blind review

ABSTRACT

Graph Machine Learning as a Service (GMLaaS) platforms increasingly implement explainability interfaces to meet regulatory transparency requirements. However, this transparency creates exploitable vulnerabilities for model extraction attacks. We present the first model extraction attack specifically designed for graph classification under strict black-box constraints where the attacker observes only discrete class labels and binary explanation masks (no probability scores, gradients, or confidence values). Our method (1) uses model explanation outputs to guide Monte Carlo edge sensitivity estimation toward decision boundaries, with Hoeffding concentration guarantees on estimation accuracy and (2) exploits explanation subgraphs to efficiently narrow the boundary search space. Extensive experiments on benchmark graph datasets across multiple domains demonstrate our method's superiority over comparable baselines. These findings demonstrate that such explainability interfaces create exploitable attack surfaces, informing both defensive mechanisms and policy frameworks for explainable AI mandates. The implementation code is provided in `https://anonymous.4open.science/r/GNN-MEA-288B/`.

1 INTRODUCTION

Graph Machine Learning as a Service (GMLaaS) allows organizations to distribute access to proprietary graph-based machine learning models through publicly accessible APIs while keeping internal model workings secure (Li et al., 2025), widely applied in applications like fraud detection (Amazon Web Services, 2022) and recommendation systems (Wu et al., 2022; UC Berkeley School of Information, 2023). However, this paradigm faces a fundamental security threat: model extraction attacks, which reconstructs functionally similar surrogates to replicate the victim's decision-making behavior by adversarially exploiting API access to query the inner models (Tramèr et al., 2016; Oliynyk et al., 2023; Maslej et al., 2025; Zhou et al., 2020). This threat has motivated growing study across ML paradigms, spanning attacks, defenses, and ownership verification (Shen et al., 2026; Li et al., 2026; Zhao et al., 2025). Compounding this threat, growing regulatory requirements (European Parliament and Council of the European Union, 2024; Colorado General Assembly, 2024) and user demand (Rosenbacke et al., 2024) are pushing providers to further augment their APIs with explainability interfaces, where users would additionally be aware of the most influencing graph componentson the predicted labels (Ying et al., 2019). While such transparency mechanisms help satisfy compliance requirements, they also expose a new attack channel that fundamentally threatens the security premise GMLaaS is built upon. This fundamental tension between mandated transparency and model security has been identified as an open challenge for MLaaS (Nimase et al., 2025). This raises a critical question: can the explanations mandated for transparency be weaponized to steal the same models they explain?

Recent work has begun exploiting this vulnerability, demonstrating that explanations can enable more efficient model extraction attacks (Oksuz et al., 2024; Miura et al., 2024; Ma et al., 2025). However, this emerging area remains sparsely studied, especially for graph neural networks. Current studies face three key limitations. **First**, existing GNN extraction methods lack a principled understanding of which training data is most effective for extraction. Current approaches construct query samples by centering on high-confidence components (Oksuz et al., 2024; Miura et al., 2024; Ma et al., 2025), but never formalize why these samples aid extraction or whether better strategies exist. Intuitively, a high-fidelity surrogate must replicate the victim's behavior not only in high-confidence regions

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.

but also near decision boundaries where predictions change. Previous work on general (non-graph) model extraction has empirically demonstrated the effectiveness of boundary sampling (Biton Dor & Mirsky, 2024), yet this insight has not been extended to the GNN domain where the discrete, combinatorial nature of graph data introduces fundamentally different challenges. **Second**, boundary search techniques developed for continuous domains cannot transfer to discrete graph spaces. Existing methods utilizing boundary points (Biton Dor & Mirsky, 2024; Davies et al., 2026) require input space continuity to efficiently locate decision boundaries. On graphs, these techniques fundamentally fail: given two graphs G_1, G_2 with different predicted labels, there is no unique "midpoint" as combinatorially many graphs exist at equal edit distance from both (Figure 1) (Piao et al., 2023). Moreover, naively estimating edge sensitivities over all $O(n^2)$ possible edges incurs prohibitive query costs (Jin et al., 2025), compounded by the hybrid discrete-continuous nature of graph data that prevents direct application of standard optimization techniques (Dai et al., 2018; Molina-Pérez et al., 2024). **Third**, every existing GNN extraction method either requires access to rich continuous information (probability scores, continuous importance vectors, or gradient-derived signals) that realistic API deployments do not provide or leaves the explanation channels of these deployments unexploited (Miura et al., 2024; Oksuz et al., 2024; Ma et al., 2025; Shen et al., 2021; Zhuang et al., 2024; Zhu et al., 2024). Under emerging regulatory mandates (European Parliament and Council of the European Union, 2024; Colorado General Assembly, 2024), GMLaaS providers are increasingly required to furnish explanations alongside predictions. In practice, rational providers would expose the minimum information necessary for compliance (discrete class labels and binary explanation masks) rather than continuous scores or gradients that reveal additional proprietary model internals. No existing GNN extraction method exploits explanations under these minimal-information constraints.

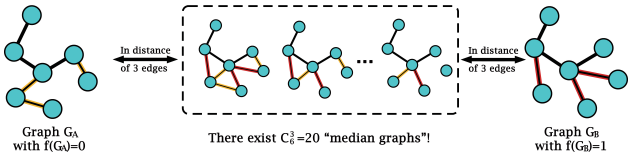


Figure 1: Multiple possible "midpoint graphs" at equal edit distance from G_A and G_B.

To address these limitations, we (1) define and investigate the effectiveness of boundary samples for GNN extraction, formalizing the extraction problem as a boundary pair search. (2) Propose an explanation-guided search attack that locates such boundary samples under strict black-box constraints where the attacker observes only discrete predictions and binary explanation masks. Our search iteratively estimates edge sensitivities via Monte Carlo sampling from discrete label responses, flips the most sensitive edges, and verifies boundary crossings. We restrict candidates to edges within or adjacent to the explanation subgraph to reduce the search space from $O(|V|^2)$ to $O(|V_{\text{sub}}|^2)$. We focus on graph classification, a setting that presents unique challenges for model extraction and has not been previously addressed under strict black-box constraints with explanation access. Extensive experiments across molecular, visual, and synthetic graph benchmarks validate our framework. In summary, our contributions are:

- **Boundary Sample Effectiveness for GNNs**: We provide the first formalized definition of boundary samples for GNN extraction and empirically validate their superiority over random and high-confidence sampling strategies, recasting model extraction as a boundary pair search problem.
- **Novel Explanation-Guided Extraction Attack**: Under strict black-box constraints with only discrete predictions and binary explanation masks, we develop the first model extraction attack for graph classification that combines a provably accurate estimator of discrete edge sensitivity with explanation-guided candidate reduction to efficiently generate boundary samples.
- **Comprehensive Empirical Evaluation**: We conduct extensive experiments across eight graph datasets spanning molecular, visual, and synthetic domains, three victim architectures, and two explainer types, demonstrating consistent superiority over adapted baselines in surrogate-victim fidelity.

2 Preliminaries and Motivations

Graph Neural Networks and Explanations. A graph $G = (V, E, X)$ consists of nodes V, edges $E \subseteq V \times V$, and node features $X \in \mathbb{R}^{|V| \times d}$. Graph neural networks (GNNs) (Kipf & Welling, 2017; Veličković et al., 2018; Hamilton et al., 2018) process G via a message passing mechanism to obtain

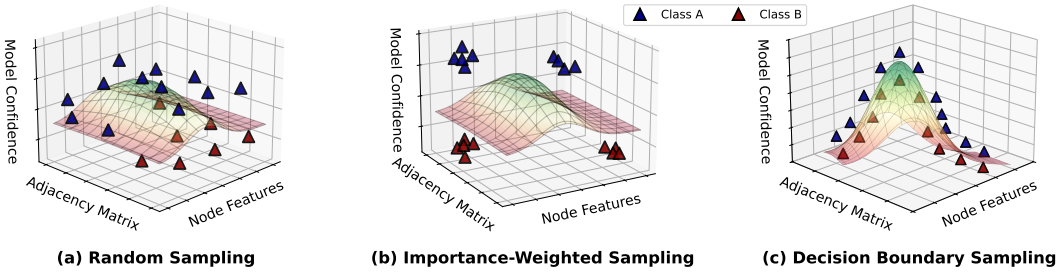


Figure 2: Sampling strategies for model extraction: surfaces are decision boundaries and triangles are classifications with smoothing for visualization clarity. Existing methods are (a) or (b), focusing on randomly sampling or sampling areas with high confidence. Our strategies are (a) and (c).

node-level embeddings. In graph classification tasks, the GNN, denoted as f, further applies a pooling operation for a graph-level embedding, and maps the embedding to a categorical label: $f : \mathcal{G} \to [C]$, where $[C] = \{1, \ldots, C\}$ denotes the label space. Under the specific model f and the graph G, a GNN explainer h then returns an explanatory subgraph $G_{\text{sub}} = (V_{\text{sub}}, E_{\text{sub}})$ where $V_{\text{sub}} \subseteq V$ and $E_{\text{sub}} \subseteq E$, identifying nodes and edges most influential for the prediction $f(G)$. Representative methods include GNNExplainer (Ying et al., 2019) and PGExplainer (Luo et al., 2020).

GNN Boundary Sampling Motivation. To better address the existing gap on extraction problem formulating and answer the question we raised in Section 1: "What kind of sampling data is effective for extracting a GNN?", we investigate the effectiveness of boundary sampling on GNN extraction to solidify our motivation beforehand. We conduct a controlled experiment isolating the effect of boundary proximity on extraction fidelity across five binary benchmark datasets (AIDS, MUTAG, NCI1, PTC_FM, Tox21_AhR; dataset statistics are provided in Appendix E.1). Full experimental details are provided in Appendix C.

For each graph G in a shadow dataset, we exhaustively test all single-edge toggles (adding or removing an edge) to identify *boundary pairs* (G, G') where $f(G) \neq f(G')$, and classify remaining graphs as *non-boundary samples*. We compare four training strategies on equal-sized training sets: (1) Boundary: trained exclusively on boundary pair samples, (2) Non-boundary: trained on non-boundary samples only, (3) Hybrid: trained on 50% boundary pairs + 50% random samples from the full shadow set, and (4) Shadow: trained only on random samples from the full shadow set.

Across all five datasets, surrogates with boundary data consistently outperform those without (Figure 3), confirming that decision boundary information disproportionately aids model extraction. We provide full experimental details and a theoretical analysis in Appendix C.

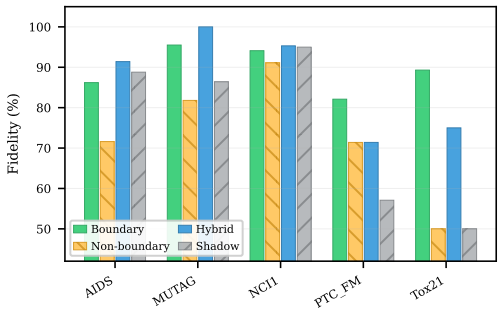


Figure 3: Fidelity (%) by sampling strategy on equal-sized training sets. Full results in Appendix Table 3.

Problem Formulation. In this work we consider a *strict black-box* model extraction attack against graph neural networks. Let $f : \mathcal{G} \to [C]$ denote a victim GNN classifier that maps graphs to discrete class labels $[C] = \{1, \ldots, C\}$. The attacker has query access only: given any graph G, the attacker can observe the predicted label $y(G) = f(G) \in [C]$, but cannot access probability scores, logits, confidence values, or internal model parameters. This strict assumption reflects realistic deployment scenarios where APIs return only classification decisions.

In general model extraction attack, the attacker's goal is to train a surrogate model g replicating the decision behavior of f, which is however uneasy to be formalized in black-box setting. Motivated by the stated effectiveness of GNN boundary samples, we instead define our problem as constructing a query set $\bar{\mathcal{G}}$ consisting of *boundary pairs*: pairs of similar graphs that receive different predictions. Formally, we aim to design an effective algorithm M to find as much boundary pairs (G, G') as

possible, and replicate a model g trained on the pairs to rebuild the same decision boundary:

$$M(Q, f, h, \delta) = \arg\max_{M} |\bar{\mathcal{G}}|, \bar{\mathcal{G}} = \{(G, G') : f(G) \neq f(G') \wedge \mathrm{dis}(G, G') \leq \delta\}$$
$$g = \arg\max_{g} \sum_{(G,G') \in \bar{\mathcal{G}}} \mathbb{I}(g(G) = f(G)) + \mathbb{I}(g(G') = f(G')) \tag{1}$$

Q is a budget that the adversary can afford to query on the victim model. h is a GNN explainer on the holder side that returns a discrete explanatory subgraph $G_{\mathrm{sub}} = (V_{\mathrm{sub}}, E_{\mathrm{sub}})$ identifying nodes and edges important for the prediction $f(G)$. $\delta > 0$ is a distance threshold and $\mathrm{dis}(\cdot, \cdot)$ measures graph similarity, and the distance function is defined as: $\mathrm{dis}(G, G') = \frac{1}{2}\|A - A'\|_0$, i.e., the number of undirected edges that differ. We focus on structural perturbation in this work, while our later experimental evaluation empirically demonstrates that edge flips alone are sufficient for constructing our boundary search. Feature perturbation can be incorporated if needed for feature-dominated tasks. It is worth noting that a smaller δ yields boundary pairs that more precisely characterize the local decision boundary geometry of f: a queried boundary pair forces every surviving hypothesis to place its decision boundary within an $O(\delta)$ neighborhood of the pair, as we prove in Lemma C.2 (Appendix C.2).

3 XSTEAL Framework

In this section we describe how to construct our method to search boundary pairs (G, G') satisfying the constraint in Eq. (1) using only black-box queries to the victim GNN f and its coupled explainer h. To begin with, we will firstly introduce up a warm-up algorithm that serves as the background of our designed method, and briefly introduce the arising challenges along with our corresponding technique solutions as an overview. Afterwards, we will formalize these components in details.

Warm-Up. We notice that in convex machine learning on continuous data, it's easy to construct a binary search method: firstly, sampling two data x and x' with different prediction results; secondly, choose the middle data point $\frac{x+x'}{2}$ and make prediction on it; thirdly, replace x/x' with the middle point that holds the same prediction, and check if the distance of the new point pair satisfies δ. If not, we repeat from the first step on the new pair until we reach a satisfied boundary pair.

Challenges and Solutions. This simple binary search is efficient and ideal, however, facing a primary challenge when practically fitting into our problem: **(C1)** in graph learning domain, even we assume the GNN model is simple as a convex model, the discrete nature of the graph data makes it impossible to find proper middle point efficiently. To address this primary challenge, we propose **(S1)** a boundary search procedure that optimizes a sample data towards decision boundary guided by the data's local gradients. However, this requirement on gradients raised another challenge, that **(C2)** in our black-box extraction attack setting we are unable to obtain the local gradients on the victim model directly. Therefore, we accordingly propose **(S2)** a query-based Monte Carlo estimator on edge sensitivities, which approximates the local gradients by summarizing query results on the neighboring data. While this query-based estimation is provably accurate as an estimator of edge sensitivity, nevertheless, it requires certain cost on queries and calculations. **(C3)** Noticing in a graph data there are $\mathcal{O}(|V|^2)$ edges, including existent and nonexistent ones, waiting to be estimated, the query burden in total makes the estimation extremely inefficient. To further address this efficiency challenge, we refine our method with **(S3)** an explanation-guided reduction on candidate edges for query efficiency.

3.1 Discrete Edge Sensitivity Estimation from Label Queries

First, we introduce our estimator of edge sensitivity. Fix a base graph G and let $G^{(t)}$ denote the current graph at iteration t of our search procedure. We write $G^{(t)} = (A^{(t)}, X^{(t)})$ and $y^{(t)} \coloneqq y\big(G^{(t)}\big)$. Let $y^{(0)} = y(G)$ denote the original prediction of the unperturbed graph.

Black-box constraint. In contrast to white-box attacks that compute $\nabla_A \mathcal{L}$ directly, we have access only to the discrete prediction $y(\cdot) \in [C]$. We cannot observe probabilities, gradients, logits, or loss values. The same constraint applies to the explanation: the provider runs the explainer h white-box

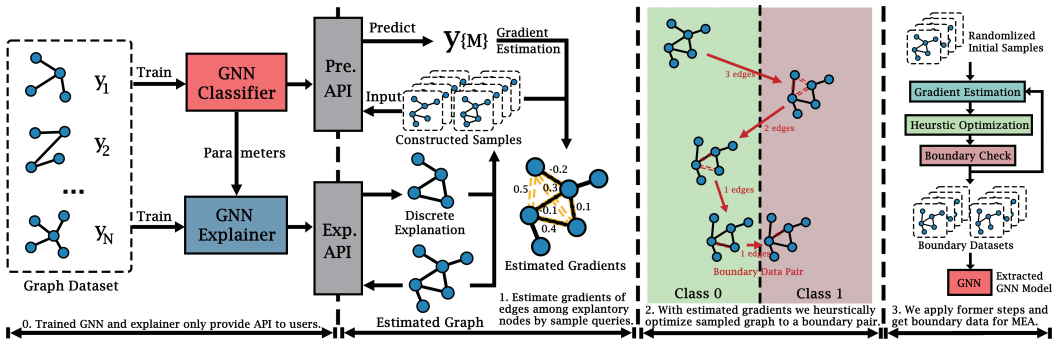


Figure 4: A visual depicting the setup of MEA and the pipeline of our proposed attack.

on its own model to produce the mask, but the attacker receives only the resulting discrete node/edge set with no access to h's gradients, importance scores, or internal parameters. This *strict black-box* assumption reflects realistic deployment scenarios where APIs return only class labels and their accompanying discrete explanation masks.

Estimator of edge sensitivity. To estimate the sensitivity of edge e without access to continuous outputs, we employ a Monte Carlo sampling approach. The key insight is that an edge's contribution to pushing the prediction away from $y^{(0)}$ can be estimated by observing how often the prediction changes when that edge is present versus absent across random perturbations of the graph.

For each candidate edge $e = (i, j) \in \mathcal{C}(G^{(t)})$, we define the graph obtained by flipping that edge:

$$G^{(t)} \oplus e \;=\; \big(A^{(t)} \oplus e, X^{(t)}\big), \tag{2}$$

where $A^{(t)} \oplus e$ is obtained from $A^{(t)}$ by setting $A^{(t)}_{ij} \leftarrow 1 - A^{(t)}_{ij}$ and $A^{(t)}_{ji} \leftarrow 1 - A^{(t)}_{ji}$.

For the m-th Monte Carlo sample, we generate a random perturbation ξ_m applied to edges other than e. Let $\tilde{G}^{(t)}_m$ denote the perturbed graph with edge e in its original state, and $\tilde{G}^{(t)}_m \oplus e$ denote the same perturbation but with edge e flipped. We define the binary observation:

$$Z^{(t)}_{e,m} \;=\; \mathbf{1}\big[y(\tilde{G}^{(t)}_m \oplus e) \neq y^{(0)}\big] - \mathbf{1}\big[y(\tilde{G}^{(t)}_m) \neq y^{(0)}\big], \tag{3}$$

where $\mathbf{1}[\cdot]$ is the indicator function. Each $Z^{(t)}_{e,m} \in \{-1, 0, +1\}$ measures whether flipping edge e pushes the prediction away from the original label $(+1)$, back toward it (-1), or has no effect (0), under perturbation ξ_m. This class-agnostic formulation naturally handles both binary and multi-class settings: for binary classification it is equivalent to targeting the opposite class, while for multi-class it rewards any boundary crossing without privileging a specific target class. This spreads search across all $\binom{C}{2}$ pairwise boundaries which is the fragmentation effect analyzed in Section 4.2 (RQ3), for which targeted per-transition search is the natural refinement.

We estimate the edge sensitivity by the empirical mean over n_e samples:

$$\hat{g}^{(t)}_e \;=\; \frac{1}{n_e} \sum_{m=1}^{n_e} Z^{(t)}_{e,m}. \tag{4}$$

A positive $\hat{g}^{(t)}_e$ indicates that flipping edge e tends to push the prediction away from $y^{(0)}$ toward a decision boundary, while a negative value suggests it reinforces the current prediction. The magnitude reflects the strength of this effect.

Perturbation scheme. Each perturbation ξ_m is generated by independently flipping each edge $(i', j') \neq e$ with probability $p_{\text{flip}} \in (0, 0.5)$. This creates diverse graph neighborhoods around $G^{(t)}$ while preserving the overall structure. When $p_{\text{flip}} = 0$, we recover the deterministic finite-difference estimate. Larger values are more robust to local decision boundary geometry, however at the cost of increased variance.

Concentration guarantees. Since each $Z_{e,m}^{(t)} \in [-1, +1]$, Hoeffding's inequality provides a concentration bound for our estimator.

Theorem 3.1 (Hoeffding bound for edge sensitivities)**.** *Fix an iteration t and an edge $e \in \mathcal{C}(G^{(t)})$. Let $g_e^{(t)} = \mathbb{E}[Z_{e,m}^{(t)}]$ denote the true expected sensitivity and $\hat{g}_e^{(t)}$ its empirical estimate. Then for any $\epsilon > 0$,*

$$\Pr\Big(\big|\hat{g}_e^{(t)} - g_e^{(t)}\big| \geq \epsilon\Big) \;\leq\; 2\exp\Big(-\frac{n_e \epsilon^2}{2}\Big). \tag{5}$$

Equivalently, for any confidence level $1 - s \in (0, 1)$, with probability at least $1 - s$:

$$\big|\hat{g}_e^{(t)} - g_e^{(t)}\big| \;\leq\; \sqrt{\frac{2\log(2/s)}{n_e}}. \tag{6}$$

Proof. (Sketch) The binary observations Z_e^t we made under the Monte Carlo sampling are independent and all satisfy $Z_{e,m}^t \in [-1, 1], \forall m \in [1, n_e]$. Therefore by substituting the range $[-1, 1]$ and n_e into Hoeffding inequality (Boucheron et al., 2013), we could directly reach above equations. □

The estimation error decays as $O\Big(\sqrt{\frac{1}{n_e}\log\frac{1}{s}}\Big)$: a per-estimate guarantee on the error of a single edge sensitivity at a given iteration using only discrete predictions rather than a uniform bound over the search or a convergence result for what remains a heuristic procedure. Because the stopping criterion in Eq. equation 8 is verified directly against the victim, every recorded boundary pair holds regardless of estimate quality. Moreover, the search uses $\hat{g}_e^{(t)}$ only to rank candidates (Eq. equation 7) so it needs correct ordering, not accurate magnitudes. Corollary I.1 (Appendix I) gives the sample count under which top-k selection is order-preserving; it exceeds our practical $n_e = 5$ for small sensitivity gaps, so it characterizes the regime rather than certifying each run.

3.2 Boundary Search via Estimated Edge Sensitivities

Using the estimated sensitivities $\{\hat{g}_e^{(t)}\}_{e \in \mathcal{C}(G^{(t)})}$, we perform a discrete, gradient-guided search in graph space to locate decision boundaries. Since we aim to push the prediction away from the original label $y^{(0)}$, we select the k edges with the largest positive estimated sensitivities:

$$S^{(t)} \;=\; \mathrm{TopK}_{\max}\Big(\big\{\hat{g}_e^{(t)} : e \in \mathcal{C}(G^{(t)})\big\}, k\Big), \tag{7}$$

where $S^{(t)} \subseteq \mathcal{C}(G^{(t)})$ contains the k edges most likely to push the prediction across a decision boundary. We then update the graph by flipping all edges in $S^{(t)}$: $G^{(t+1)} = \mathrm{Update}(G^{(t)}, S^{(t)})$, where Update applies the flip operation in Eq. equation 2 to each $e \in S^{(t)}$. Starting from $G^{(0)} = G$, we iterate until reaching an iteration $t^\star$ such that

$$y\big(G^{(t^\star)}\big) \neq y^{(0)} \quad \text{and} \quad \delta_e\big(G^{(0)}, G^{(t^\star)}\big) \leq \delta. \tag{8}$$

At this point, we record the boundary pair $(G^{(0)}, G^{(t^\star)})$. Its exit conditions are detailed in Appendix D.2.

3.3 Explanation-Guided Query Reduction

Naively, $\mathcal{C}(G^{(t)})$ could contain all possible edges, leading to $O(|V|^2)$ queries per iteration. To make the attack query-efficient, we leverage the explanatory subgraph produced by h to restrict attention to a smaller, high-impact region of the graph.

Given $G^{(t)}$, we query the explainer for $G_{\mathrm{sub}}^{(t)}$ and write $N(V_{\mathrm{sub}}^{(t)})$ for the one-hop neighborhood of its nodes (formal definitions in Appendix H). We construct the candidate edge set as

$$\mathcal{C}\big(G^{(t)}\big) = \underbrace{\big\{(i,j) : i,j \in V_{\mathrm{sub}}^{(t)},\, i<j\big\}}_{\text{within-explanation edges}} \cup \underbrace{\big\{(i,j) : i \in V_{\mathrm{sub}}^{(t)},\, j \in N(V_{\mathrm{sub}}^{(t)}),\, i<j\big\}}_{\text{explanation-to-neighbor edges}} \cup \underbrace{\mathcal{S}_{\mathrm{carry}}^{(t)}}_{\text{carry-over edges}} \cup \underbrace{\mathcal{S}_{\mathrm{rand}}^{(t)}}_{\text{random exploration}}. \tag{9}$$

Here $\mathcal{S}_{\mathrm{carry}}^{(t)}$ contains edges that were highly influential in previous iterations, and $\mathcal{S}_{\mathrm{rand}}^{(t)}$ contains randomly sampled edges for global exploration. The resulting candidate-set size bound, query reduction, and a worked example are given in Appendix H.

3.4 SUMMARY: METHOD OVERVIEW

With all above components we design, we are able to summarize our whole method in the following overview. The resulting boundary-pair set, surrogate training set, and training objective are formalized in Eqs. equation 10–equation 12 (Appendix D.1). We employ a two-phase approach. **Phase 1** cheaply tests all single-edge removals ($O(|E(G)|)$ queries): if any removal changes the prediction we record the boundary pair, else the graph passes to Phase 2. Restricting to removals rather than also adding edges trades some yield for this lower cost, most where boundaries are scarce (Appendix C.1, Table 5); Phase 2 recovers the difference. **Phase 2** triggers only when Phase 1 yields insufficient pairs, applying the full MC boundary search (Sections 3.1–3.3) to remaining graphs. This design reflects that *predictions find easy boundaries; explanations unlock hard ones*. Phase 1 is quantity-limited on low boundary-rate datasets and quality-limited on feature-dominated datasets; our phase ablation (Appendix F.5) confirms both effects. The complete algorithm is provided in Appendix D (Appendix Algorithm 1 for the overall procedure, Appendix Algorithm 2 for the BOUNDARYSEARCH subroutine).

4 EXPERIMENTS

We evaluate our method across eight graph classification benchmarks (Table 6 in Appendix E.1), three victim architectures (GCN, GAT, GraphSAGE), and two explainer types (GNNExplainer, PGExplainer). We organize our evaluation around three research questions:

- **RQ1**: Does boundary sampling improve extraction fidelity over random and high-confidence strategies?
- **RQ2**: Does our method generalize across victim architectures and explainer types? (Appendix F.1, F.2)
- **RQ3**: How do individual components (explanation guidance, MC estimation, two-phase design) contribute to performance? (Appendix F.4, F.3, F.5)

4.1 EXPERIMENTAL SETUP

We report fidelity (surrogate-victim agreement on a class-balanced test set) and accuracy (surrogate vs. ground truth). All surrogates use the same architecture family as the victim with fixed hyperparameters to isolate the effect of training data quality. Full details are in Appendix E.

Datasets. We evaluate on eight graph classification benchmarks spanning molecular (AIDS, MUTAG, NCI1, PTC_FM, Tox21_AhR), visual (MNIST, Letter-low), and synthetic (Synthie) domains. Dataset sizes range from 188 to 70,000 graphs, with five binary and three multi-class tasks (up to 15 classes), testing our method across diverse scales and decision boundary complexities. Full dataset statistics are provided in Table 6 (Appendix E.1).

Evaluation Metrics. We report fidelity, defined as the agreement rate between surrogate and victim predictions on a class-balanced test set. Fidelity is our primary metric because it measures how faithfully the surrogate replicates the victim's decision logic, independent of the victim's own correctness. We additionally report accuracy (surrogate vs. ground-truth labels) as a secondary measure. Details on test set balancing are in Appendix E.2.

Baselines. Existing GNN extraction attacks require either continuous victim outputs (probability scores, embeddings, or importance vectors) (Shen et al., 2021; Podhajski et al., 2024; Ma et al., 2025; Miura et al., 2024) or operate on node classification over a shared graph (Wu et al., 2021; Zhuang et al., 2024; Guan et al., 2024). Since no existing method operates under our discrete-only graph classification constraints, we construct baselines from straightforward strategies and adapt EGSteal (Ma et al., 2025), which targets the same task but under a white-box setting with continuous outputs. Our adaptation retains only its style-invariance augmentation, as the core explanation alignment mechanism requires continuous importance vectors that are unavailable in our setting. Full details are in Appendix E.10:

- **Shadow-Only**: Random shadow samples labeled by the victim, with no explanation access or boundary-seeking strategy.

Dataset	Expl.	Shadow Fid.	Shadow Acc.	Non-boundary Fid.	Non-boundary Acc.	EGSteal-BB Fid.	EGSteal-BB Acc.	Boundary (XSTEAL) Fid.	Boundary (XSTEAL) Acc.	Hybrid (XSTEAL) Fid.	Hybrid (XSTEAL) Acc.
AIDS	GNNExp	86.8±2.3	72.1±2.3	71.3±0.8	66.4±0.0	65.8±1.1	67.2±0.7	**91.1±0.8**	75.9±0.0	85.1±3.5	71.0±3.9
	PGExp					68.4±0.4	70.4±0.4	83.6±7.4	70.7±6.0	88.8±1.9	74.1±1.9
Letter-low	GNNExp	68.6±2.1	69.6±1.8	44.4±1.6	43.0±1.0	68.9±1.6	68.4±1.9	63.0±3.7	65.7±2.7	**72.3±0.7**	72.3±0.7
	PGExp					66.4±3.9	66.9±5.5	69.9±3.7	71.6±4.9	69.4±1.8	69.1±1.7
MNIST	GNNExp	53.4±1.3	26.4±0.8	48.3±2.5	24.4±0.4	53.8±1.7	26.5±0.6	54.9±2.3	26.2±0.6	**55.1±1.7**	26.6±0.4
	PGExp					50.2±0.8	25.1±0.3	52.0±2.6	25.7±0.8	53.1±3.1	25.9±0.7
MUTAG	GNNExp	81.8±3.7	68.2±3.7	84.8±2.1	65.2±2.1	90.9±0.0	68.2±0.0	**100.0±0.0**	77.3±0.0	97.0±2.1	74.2±2.1
	PGExp					90.9±0.0	68.2±0.0	**100.0±0.0**	77.3±0.0	95.5±0.0	72.7±0.0
PTC_FM	GNNExp	77.4±2.1	54.8±2.1	75.0±0.0	50.0±0.0	59.5±2.1	44.1±2.1	85.7±3.6	53.6±3.6	**90.5±2.1**	58.3±2.1
	PGExp					59.5±2.1	44.1±2.1	84.5±2.1	52.4±2.1	82.1±9.5	50.0±9.4
Synthie	GNNExp	39.7±5.5	35.3±5.0	47.4±5.5	44.2±4.2	42.9±4.0	37.8±3.3	**51.3±5.5**	44.2±4.2	46.8±3.9	40.4±6.8
	PGExp					44.9±5.0	37.8±3.3	46.2±3.1	43.6±7.4	46.2±0.0	40.4±4.2
NCI1	GNNExp	94.3±1.7	66.3±0.2	91.4±0.3	65.7±0.3	88.9±1.0	65.4±0.3	**96.0±0.2**	66.3±0.2	92.2±2.8	65.9±0.6
	PGExp					84.8±1.3	63.5±0.7	95.2±0.5	66.5±0.2	92.6±2.6	65.9±0.8
Tox21	GNNExp	60.7±5.1	54.8±6.1	54.8±3.4	48.8±3.4	54.8±3.4	48.8±3.4	89.3±2.9	83.3±4.5	77.4±9.4	76.2±7.3
	PGExp					57.1±5.8	51.2±3.4	**92.9±2.9**	82.1±2.9	91.7±1.7	80.9±1.7

Table 1: Extraction results on *GCN* victim at matched training set sizes. Fid. = Fidelity (%), Acc. = Accuracy (%). Best fidelity per dataset in **bold**.

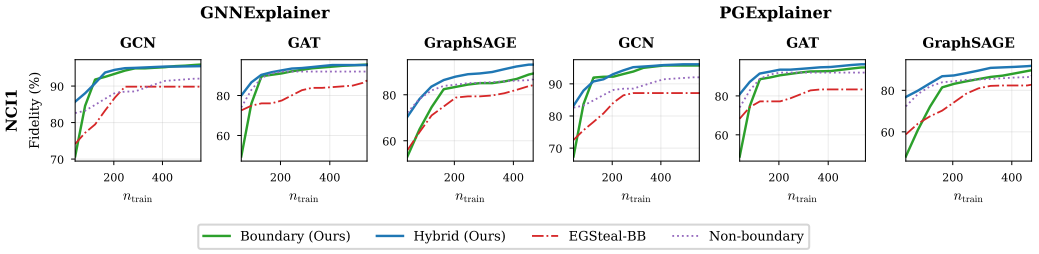


Figure 5: Fidelity (%) across increasing training set size on NCI1.

- **Non-boundary**: Shadow graphs where no single-edge removal changes the victim's prediction, representing interior-region samples far from decision boundaries.
- **EGSteal-BB**: Adaptation of EGSteal's style-invariance augmentation to discrete explanation masks; perturbs edges outside the explanation subgraph while preserving the original label.

We additionally report a budget-matched **No-Exp** control that replaces explanation-guided candidate selection with uniform random selection at equal candidate count and query budget, isolating the explanation signal from raw computation (Appendix F.4).

Our Methods. We evaluate two variants of our boundary-based extraction framework:

- **Boundary**: Full two-phase method (Section 3.4) combining cheap exhaustive $\delta_e = 1$ search with explanation-guided MC boundary search.
- **Hybrid**: Allocates 50% of the query budget to boundary pair generation and 50% to random sampling, combining targeted boundary information with broad distributional coverage.

4.2 Main Results

In Table 1 we present extraction results on GCN victims. While in Appendix, Figure 7 visualizes fidelity across all datasets grouped by each explainer, and Figure 5 shows fidelity-vs-training-set-size trajectories on NCI1. Full results for GAT and GraphSAGE victims are in Appendix F.1–F.2; trajectory plots across all architectures are in Appendix G.

RQ1: Boundary sampling improves extraction fidelity. On all datasets, Boundary or Hybrid achieves the highest fidelity averaged over explainers against the explainer-free baselines, with the largest gains where baselines struggle, such like Tox21 where Boundary (92.9%) outperforms Shadow-Only (60.7%) by 32.2 points, MUTAG (100.0% vs. 81.8%), and PTC_FM (Hybrid 90.5% vs. 77.4%). We notice non-boundary consistently underperforms, confirming interior samples alone are insufficient. Additionally, Figure 5 shows Boundary and Hybrid separate from baselines early on

NCI1 and converge to their advantage as training size grows. EGSteal-BB underperforms Shadow-Only on 4 of 8 datasets (GCN/GNNExplainer)—e.g., PTC_FM: 59.5% vs. 77.4%. We analyze the cause in Appendix F.6.

RQ2: Generalization across architectures and explainers. The outperforming pattern of our methods holds across victim architectures: on GAT victims (Appendix Table 9), Boundary achieves 97.6% on Tox21 (GNNExplainer) and 95.5% on MUTAG; on GraphSAGE (Appendix Table 10), Boundary reaches 94.9% on Tox21 and 93.3% on PTC_FM. Performance is consistent across both explainer types, with neither GNNExplainer nor PGExplainer systematically dominating. Multi-class results are analyzed in Appendix F. This robustness reflects a key design advantage: our method uses explanations *heuristically* to narrow the candidate edge set rather than *prescriptively* to assign labels. An imperfect explanation may degrade search *efficiency* as we would explore a suboptimal region, but not *correctness* since every boundary pair for surrogate training is verified by the victim's own predictions.

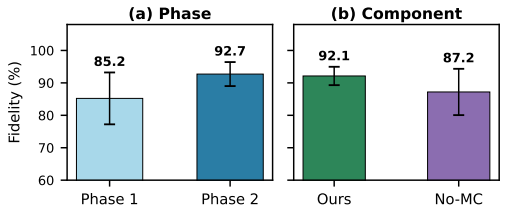


Figure 6: Phase and Component Ablations averaged across datasets and methods with error bars.

RQ3: Component contributions. Figure 6 summarizes both ablation studies. Phase 2 improves mean fidelity by 7.5 points over Phase 1 (92.7% vs. 85.2%) with substantially lower variance, confirming that explanation-guided MC search recovers boundary pairs that cheap exhaustive search misses—especially on low boundary-rate datasets. In the component study we evaluate No-Exp which draws candidate edges uniformly at random instead of the explanation subgraph and No-MC which uses a single sample to estimate each edge sensitivity. Under a budget-matched comparison (Appendix F.4), explanation guidance improves fidelity over the No-Exp control on all three tested datasets, by $+10.7$, $+0.3$, and $+10.7$ points on PTC_FM, NCI1, and Tox21_AhR at matched training-set size with consistent gains at equal query budget. This confirms that the improvement comes from which candidates are selected rather than from additional computation. For the No-MC method, the component ablation (Figure 6b) shows that removing Monte Carlo sampling largely degrades its fidelity compared with ours (87.2% vs 92.1%). Per-dataset breakdowns are in Appendix F.3 and F.5.

5 Conclusion

In this work we answer the question posed at the start of the paper affirmatively: subgraph explanations can be weaponized to steal graph neural networks. By recasting model extraction as a boundary pair search problem, we developed the first explanation-guided model extraction attack for GNN graph classification under strict black-box constraints. Our proposed Monte Carlo edge sensitivity estimation, backed by Hoeffding concentration guarantees, enables efficient boundary pair search by restricting estimated candidates to explanation-relevant subgraph regions. Extensive experiments across eight datasets, three victim architectures, and two explainer types demonstrate our attack's consistent superiority over adapted baselines, with fidelity gains of up to 32 points on challenging datasets. Our baselines are necessarily adapted from methods built for other threat models since no prior attack runs under our constraints; we discuss this and further directions in Appendix B. These findings reveal a fundamental tension between regulatory demands for model transparency and the security of deployed GNN services, underscoring the need for defenses that can provide meaningful explanations without exposing decision boundary information.

AI Use Statement

We used AI assistants to polish the writing and to assist in proving the mathematical claims in this paper. All AI-assisted text and proofs were reviewed and verified by the authors, who take responsibility for the final content of this work.

Ethics Statement

This work studies an attack on deployed GNN services, and we disclose it to help providers and regulators build defenses rather than to enable misuse. All experiments use public benchmark datasets and locally trained victim models. As such, we attack no live or third-party system, and no private or personal data is involved.

Reproducibility Statement

We provide an anonymized implementation at `https://anonymous.4open.science/r/GNN-MEA-288B/`. The full attack is specified in Appendix D (Algorithms 1 and 2) with all hyperparameters, victim and surrogate configurations, explainer settings, data splits, and query budgets in Appendix E. All datasets are public benchmarks described in Appendix E.1. Complete proofs of our theoretical results with stated assumptions appear in Appendix C.2 and I.

References

Amazon Web Services. Build a GNN-based real-time fraud detection solution using Amazon SageMaker, Amazon Neptune, and the Deep Graph Library. `https://aws.amazon.com/blogs/machine-learning/build-a-gnn-based-real-time-fraud-detection-solution-using-amazon-sagemaker-amazon-neptune-and-the-deep-graph-library/`, August 2022.

Maor Biton Dor and Yisroel Mirsky. Efficient model extraction via boundary sampling. In *Proceedings of the 2024 Workshop on Artificial Intelligence and Security*, AISec '24, pp. 1–11, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400712289. doi: 10.1145/3689932.3694756. URL `https://doi.org/10.1145/3689932.3694756`.

Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. *Concentration Inequalities: A Nonasymptotic Theory of Independence*. Oxford University Press, 02 2013. ISBN 9780199535255. doi: 10.1093/acprof:oso/9780199535255.001.0001. URL `https://doi.org/10.1093/acprof:oso/9780199535255.001.0001`.

Zhan Cheng, Bolin Shen, Tianming Sha, Yuan Gao, Shibo Li, and Yushun Dong. Atom: A framework of detecting query-based model extraction attacks for graph neural networks, 2025. URL `https://arxiv.org/abs/2503.16693`.

Colorado General Assembly. SB24-205: Concerning Consumer Protections in Interactions with Artificial Intelligence Systems, 2024. URL `https://leg.colorado.gov/bills/sb24-205`.

Hanjun Dai, Hui Li, Tian Tian, Xin Huang, Lin Wang, Jun Zhu, and Le Song. Adversarial attack on graph structured data, 2018. URL `https://arxiv.org/abs/1806.02371`.

Xander Davies, Giorgi Giglemiani, Edmund Lau, Eric Winsor, Geoffrey Irving, and Yarin Gal. Boundary point jailbreaking of black-box llms. *arXiv preprint arXiv:2602.15001*, 2026.

Vijay Prakash Dwivedi, Chaitanya K. Joshi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Benchmarking graph neural networks, 2022. URL `https://arxiv.org/abs/2003.00982`.

European Parliament and Council of the European Union. Regulation (eu) 2024/1689 laying down harmonised rules on artificial intelligence (artificial intelligence act). Official Journal of the European Union, L 2024/1689, July 2024.

Faqian Guan, Tianqing Zhu, Hanjin Tong, and Wanlei Zhou. A realistic model extraction attack against graph neural networks. *Knowledge-Based Systems*, 300:112144, 2024. ISSN 0950-7051. doi: https://doi.org/10.1016/j.knosys.2024.112144. URL `https://www.sciencedirect.com/science/article/pii/S0950705124007780`.

William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs, 2018. URL `https://arxiv.org/abs/1706.02216`.

Ruiyang Jin, Yuke Zhou, Yujie Tang, Jie Song, and Siyang Gao. Query-efficient zeroth-order algorithms for nonconvex optimization, 2025. URL https://arxiv.org/abs/2510.19165.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks, 2017. URL https://arxiv.org/abs/1609.02907.

Lincan Li, Bolin Shen, Chenxi Zhao, Yuxiang Sun, Kaixiang Zhao, Shirui Pan, and Yushun Dong. Intellectual property in graph-based machine learning as a service: Attacks and defenses, 2025. URL https://arxiv.org/abs/2508.19641.

Zhengao Li, Xiaopeng Yuan, Bolin Shen, Kien Le, Haohan Wang, Xugui Zhou, Shangqian Gao, and Yushun Dong. Query-efficient domain knowledge stealing against large language models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(38):31870–31878, Mar. 2026. doi: 10.1609/aaai.v40i38.40456. URL https://ojs.aaai.org/index.php/AAAI/article/view/40456.

Dongsheng Luo, Wei Cheng, Dongkuan Xu, Wenchao Yu, Bo Zong, Haifeng Chen, and Xiang Zhang. Parameterized explainer for graph neural network. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 19620–19631. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/e37b08dd3015330dcbb5d6663667b8b8-Paper.pdf.

Bin Ma, Yuyuan Feng, Minhua Lin, and Enyan Dai. How explanations leak the decision logic: Stealing graph neural networks via explanation alignment, 2025. URL https://arxiv.org/abs/2506.03087.

Nestor Maslej, Loredana Fattorini, Raymond Perrault, Yolanda Gil, Vanessa Parli, Njenga Kariuki, Emily Capstick, Anka Reuel, Erik Brynjolfsson, John Etchemendy, Katrina Ligett, Terah Lyons, James Manyika, Juan Carlos Niebles, Yoav Shoham, Russell Wald, Toby Walsh, Armin Hamrah, Lapo Santarlasci, Julia Betts Lotufo, Alexandra Rome, Andrew Shi, and Sukrut Oak. Artificial intelligence index report 2025, 2025. URL https://arxiv.org/abs/2504.07139.

Smitha Milli, Ludwig Schmidt, Anca D. Dragan, and Moritz Hardt. Model reconstruction from model explanations. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, FAT* '19, pp. 1–9, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450361255. doi: 10.1145/3287560.3287562. URL https://doi.org/10.1145/3287560.3287562.

Takayuki Miura, Toshiki Shibahara, and Naoto Yanai. Megex: Data-free model extraction attack against gradient-based explainable ai. In *Proceedings of the 2nd ACM Workshop on Secure and Trustworthy Deep Learning Systems*, SecTL '24, pp. 56–66, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400706912. doi: 10.1145/3665451.3665533. URL https://doi.org/10.1145/3665451.3665533.

Daniel Molina-Pérez, Edgar Alfredo Portilla-Flores, Efrén Mezura-Montes, Eduardo Vega-Alvarado, and María Bárbara Calva-Yañez. Efficiently handling constraints in mixed-integer nonlinear programming problems using gradient-based repair differential evolution. *PeerJ Computer Science*, 10:e2095, May 2024. ISSN 2376-5992. doi: 10.7717/peerj-cs.2095. URL https://doi.org/10.7717/peerj-cs.2095.

Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodolà, Jan Svoboda, and Michael M. Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns, 2016. URL https://arxiv.org/abs/1611.08402.

Christopher Morris, Nils M. Kriege, Franka Bause, Kristian Kersting, Petra Mutzel, and Marion Neumann. Tudataset: A collection of benchmark datasets for learning with graphs, 2020. URL https://arxiv.org/abs/2007.08663.

Jiaming Mu, Binghui Wang, Qi Li, Kun Sun, Mingwei Xu, and Zhuotao Liu. A hard label black-box adversarial attack against graph neural networks. In *Proceedings of the 2021 ACM SIGSAC*

Conference on Computer and Communications Security, CCS '21, pp. 108–125, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450384544. doi: 10.1145/3460120.3484796. URL https://doi.org/10.1145/3460120.3484796.

Ojas Nimase, Yue Zhao, and Yushun Dong. Navigating between explainability and extractability in machine learning as a service. In *2025 IEEE International Conference on Data Mining Workshops (ICDMW)*, pp. 2483–2487, 2025. doi: 10.1109/ICDMW69685.2025.00305.

Abdullah Caglar Oksuz, Anisa Halimi, and Erman Ayday. Autolycus: Exploiting explainable artificial intelligence (xai) for model extraction attacks against interpretable models. *Proceedings on Privacy Enhancing Technologies*, 2024(4):684–699, October 2024. ISSN 2299-0984. doi: 10.56553/popets-2024-0137. URL http://dx.doi.org/10.56553/popets-2024-0137.

Daryna Oliynyk, Rudolf Mayer, and Andreas Rauber. I know what you trained last summer: A survey on stealing machine learning models and defences. *ACM Comput. Surv.*, 55(14s), July 2023. ISSN 0360-0300. doi: 10.1145/3595292. URL https://doi.org/10.1145/3595292.

Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning, 2017. URL https://arxiv.org/abs/1602.02697.

Chengzhi Piao, Tingyang Xu, Xiangguo Sun, Yu Rong, Kangfei Zhao, and Hong Cheng. Computing graph edit distance via neural graph matching. *Proc. VLDB Endow.*, 16(8):1817–1829, April 2023. ISSN 2150-8097. doi: 10.14778/3594512.3594514. URL https://doi.org/10.14778/3594512.3594514.

Marcin Podhajski, Jan Dubiński, Franziska Boenisch, Adam Dziedzic, Agnieszka Pregowska, and Tomasz P. Michalak. Efficient model-stealing attacks against inductive graph neural networks, 2024. URL https://arxiv.org/abs/2405.12295.

Kaspar Riesen and Horst Bunke. Iam graph database repository for graph based pattern recognition and machine learning. In Niels da Vitoria Lobo, Takis Kasparis, Fabio Roli, James T. Kwok, Michael Georgiopoulos, Georgios C. Anagnostopoulos, and Marco Loog (eds.), *Structural, Syntactic, and Statistical Pattern Recognition*, pp. 287–297, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg. ISBN 978-3-540-89689-0.

R. Rosenbacke, Å. Melhus, M. McKee, and D. Stuckler. How explainable artificial intelligence can increase or decrease clinicians' trust in AI applications in health care: Systematic review. *JMIR AI*, 3:e53207, 2024. doi: 10.2196/53207. URL https://doi.org/10.2196/53207.

Bolin Shen, Zhan Cheng, Neil Zhenqiang Gong, Fan Yao, and Yushun Dong. Credit: Certified ownership verification of deep neural networks against model extraction attacks, 2026. URL https://arxiv.org/abs/2602.20419.

Yun Shen, Xinlei He, Yufei Han, and Yang Zhang. Model stealing attacks against inductive graph neural networks, 2021. URL https://arxiv.org/abs/2112.08331.

Florian Tramèr, Fan Zhang, Ari Juels, Michael K. Reiter, and Thomas Ristenpart. Stealing machine learning models via prediction apis. In *Proceedings of the 25th USENIX Conference on Security Symposium*, SEC'16, pp. 601–618, USA, 2016. USENIX Association. ISBN 9781931971324.

UC Berkeley School of Information. GNNIE: GNN-recommender-as-a-service. https://www.ischool.berkeley.edu/projects/2022/gnnie-gnn-recommender-service, 2023.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks, 2018. URL https://arxiv.org/abs/1710.10903.

Zebin Wang, Menghan Lin, Bolin Shen, Ken Anderson, Molei Liu, Tianxi Cai, and Yushun Dong. Cega: A cost-effective approach for graph-based model extraction and acquisition, 2025. URL https://arxiv.org/abs/2506.17709.

Bang Wu, Xiangwen Yang, Shirui Pan, and Xingliang Yuan. Model extraction attacks on graph neural networks: Taxonomy and realization, 2021. URL https://arxiv.org/abs/2010.12751.

Shiwen Wu, Fei Sun, Wentao Zhang, Xu Xie, and Bin Cui. Graph neural networks in recommender systems: A survey. *ACM Comput. Surv.*, 55(5), December 2022. ISSN 0360-0300. doi: 10.1145/3535101. URL https://doi.org/10.1145/3535101.

Rex Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. Gnnexplainer: Generating explanations for graph neural networks, 2019. URL https://arxiv.org/abs/1903.03894.

Hao Yuan, Haiyang Yu, Jie Wang, Kang Li, and Shuiwang Ji. On explainability of graph neural networks via subgraph explorations, 2021. URL https://arxiv.org/abs/2102.05152.

Kaixiang Zhao, Lincan Li, Kaize Ding, Neil Zhenqiang Gong, Yue Zhao, and Yushun Dong. A survey on model extraction attacks and defenses for large language models, 2025. URL https://arxiv.org/abs/2506.22521.

Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI Open*, 1:57–81, 2020. ISSN 2666-6510. doi: https://doi.org/10.1016/j.aiopen.2021.01.001. URL https://www.sciencedirect.com/science/article/pii/S2666651021000012.

Zhihao Zhu, Chenwang Wu, Rui Fan, Yi Yang, Zhen Wang, Defu Lian, and Enhong Chen. Model stealing attack against graph classification with authenticity, uncertainty and diversity, 2024. URL https://arxiv.org/abs/2312.10943.

Yuanxin Zhuang, Chuan Shi, Mengmei Zhang, Jinghui Chen, Lingjuan Lyu, Pan Zhou, and Lichao Sun. Unveiling the secrets without data: can graph neural networks be exploited through data-free model extraction attacks? In *Proceedings of the 33rd USENIX Conference on Security Symposium*, SEC '24, USA, 2024. USENIX Association. ISBN 978-1-939133-44-1.

APPENDIX CONTENTS

A Related Work **15**

B Limitations and Impact Statement **15**

C Boundary Sampling: Motivation and Theory **16**

C.1 Empirical Validation 16

C.2 Theoretical Analysis of Boundary Data Effectiveness 17

D Algorithms **20**

D.1 Overall Extraction Procedure 20

D.2 Boundary Search Subroutine 21

E Experimental Details **21**

E.1 Datasets 21

E.2 Evaluation Metrics 22

E.3 Victim Model Configuration 23

E.4 Explainer Configuration 23

E.5 Attack Configuration 24

E.6 Surrogate Training 24

E.7 Data Splits and Query Budget 24

E.8 Compute Resources 24

E.9 Implementation Notes 24

E.10 Baseline Details 25

F Additional Results **25**

F.1 GAT Victim Results 26

F.2 GraphSAGE Victim Results 27

F.3 Monte Carlo Sampling Ablation 27

F.4 Matched Queries No-Exp Control 28

F.5 Phase Ablation 28

F.6 EGSteal-BB Failure Analysis 28

G Fidelity vs. Query Budget Trajectories **29**

H Query Complexity Analysis **29**

I Selection Consistency of Top-k Edge Selection **32**

A Related Work

Model Extraction Attacks. Model extraction attacks reconstruct target model functionality through query access. Early work demonstrated equation-solving attacks against simple models (Tramèr et al., 2016), later extended to deep neural networks (Papernot et al., 2017; Milli et al., 2019). Recent approaches show that samples near decision boundaries maximize extraction efficiency (Biton Dor & Mirsky, 2024). However, most methods assume access to probability scores or confidence values, assumptions violated in realistic deployments where APIs return only discrete labels. For GNNs specifically, existing attacks target node classification (Zhuang et al., 2024; Shen et al., 2021), exploiting shared graph topology and node-level embeddings. These methods fundamentally cannot transfer to graph classification, where each query involves an independent graph with only a graph-level prediction. Recent work has further explored both attack and defense strategies specific to GNN extraction (Wang et al., 2025; Cheng et al., 2025), though these also operate on node classification tasks. Zhu et al. (2024) attack graph classification from hard labels by querying where the surrogate is uncertain, whereas XSTEAL searches for victim-verified boundary pairs guided by the explanation mask. To our knowledge, no prior work addresses strict black-box model extraction for GNN graph classification with explanation access and theoretical guarantees. Table 2 positions representative extraction attacks and shows that only our setting operates on graph classification under discrete labels and discrete explanation masks. The closest methodological precedent is hard-label black-box evasion on graph classification (Mu et al., 2021). While it shares our hard-label query constraint and boundary-directed edge estimation, it solves a fundamentally different and simpler task: flipping one graph's own label; whereas our extraction task must reconstruct the victim's entire decision boundary by searching many boundary pairs to train a surrogate, further guided by explanations their setting does not use.

Explanation-Guided Attacks. Regulatory requirements such as the EU AI Act (European Parliament and Council of the European Union, 2024) increasingly mandate that MLaaS providers offer explanations, creating new attack surfaces. Methods like GNNExplainer (Ying et al., 2019), PGExplainer (Luo et al., 2020), and SubgraphX (Yuan et al., 2021) identify influential substructures for predictions. Recent work demonstrates that such explanations leak exploitable information to aid model extraction attacks (Oksuz et al., 2024; Ma et al., 2025). However, these approaches either require white-box access to explanation internals or assume continuous probability outputs. In realistic GMLaaS deployments, providers return only discrete explanation outputs (important nodes/edges) via APIs and not importance scores or gradients. Our work is the first to exploit explanations under strict black-box constraints where both predictions and explanations are discrete.

Defenses. A natural mitigation is input-validity or out-of-distribution filtering: a provider that rejects queries whose structure falls off the data manifold blunts attacks that synthesize boundary graphs. This bears unevenly on our two phases: deletion-only Phase 1 removes existing edges and cannot introduce invalid substructure whereas Phase 2's toggle search can add edges that yield off-manifold graphs and is the more detectable stage. Such filtering trades coverage for security since legitimate rare inputs may also be rejected.

Method	Task	Victim Output	Explanation Access
GNN extraction without explanations (Shen et al., 2021; Wu et al., 2021; Zhuang et al., 2024; Zhu et al., 2024)	Node/graph clf.	Prob./embeddings/labels	None
Explanation-guided extraction (Oksuz et al., 2024; Miura et al., 2024; Ma et al., 2025)	Graph/node clf.	Continuous scores	Continuous (white-box)
Boundary-sample extraction (Biton Dor & Mirsky, 2024)	Non-graph	Probabilities	None
XSTEAL (ours)	Graph clf.	Discrete label	Discrete mask

Table 2: Prior work by task, victim-output granularity, and explanation access. Only our setting operates on graph classification under discrete labels and discrete explanation masks.

B Limitations and Impact Statement

While our work presents the first model extraction attack for GNN graph classification under strict black-box constraints with explanation access, we acknowledge that no existing method operates under these same constraints. Therefore, due to the novelty of this work our baselines are necessarily adapted from methods designed for different threat models rather than direct competitors. We see several concrete directions this enables: native strict-black-box baselines for graph classification

that do not require adapting white-box methods, extension to counterfactual and other explanation families, and joint structural-feature perturbations that generalize the attack beyond edge flips. Our findings demonstrate that subgraph explanations, increasingly mandated by regulations such as the EU AI Act, create exploitable attack surfaces for model extraction under strict black-box constraints. This work is intended to inform GMLaaS providers and regulators so that appropriate defenses can be developed.

Deployment Case Study: Toxicity Screening. Regulatory transparency mandates increasingly reach high-risk classifiers: the EU AI Act (European Parliament and Council of the European Union, 2024) and Colorado SB24-205 (Colorado General Assembly, 2024) both require providers of such systems, including chemical toxicity screens, to return an explanation alongside each decision. Such a screen is a valuable target since it embeds proprietary assay data and substantial training cost, both of which a faithful surrogate lets a competitor bypass. To make this concrete, we instantiate our attack on Tox21_AhR, where the adversary registers as an ordinary API user and holds a shadow set of public molecular graphs. On each query it observes only a discrete toxicity label and a binary explanation mask, with no access to gradients, logits, probabilities, or model parameters—precisely the interface the regulation obliges the provider to expose. Running the attack within a budget of 70% of the shadow set (Appendices E.7 and H), the resulting surrogate reaches 92.9% fidelity to the victim, compared with 60.7% for undirected shadow sampling at the same budget. The attack therefore succeeds using only the discrete outputs the mandate requires so the transparency requirement itself widens the very attack surface it is meant to make accountable.

C Boundary Sampling: Motivation and Theory

C.1 Empirical Validation

Table Results. Our results are outlined below, confirming the superiority of the boundary-sampling approach across all datasets.

Dataset	Boundary	Non-boundary	Hybrid	Shadow
AIDS	86.2	71.6	**91.4**	88.8
MUTAG	95.5	81.8	**100.0**	86.4
NCI1	94.1	91.1	**95.3**	95.0
PTC_FM	**82.1**	71.4	71.4	57.1
Tox21_AhR	**89.3**	50.0	75.0	50.0

Table 3: Fidelity (%) on test sets with equal training set sizes. Best per dataset in **bold**.

Data Splits. Each dataset is split 60% victim training / 20% shadow / 20% test, stratified by class labels. The shadow set serves as the attacker's query pool; the test set is held out for fidelity evaluation.

Victim Models. We train 3-layer GCN victim models via hyperparameter search over epochs $\in \{300, 500, 700, 1000\}$ and hidden dimensions $\in \{64, 128\}$, selecting the configuration with highest test accuracy. This reflects a realistic threat model: a model owner naturally deploys the best-performing model, and stronger victims represent a harder extraction target. Full configurations and test accuracies across all architectures are provided in Appendix Table 7.

Boundary Detection. For each graph G in the shadow set, we exhaustively test all single-edge toggles ($\delta_e = 1$). Boundary detection statistics are reported in Table 4.

Training Set Construction. All four strategies (Boundary, Non-boundary, Hybrid, Shadow-Only) use equal-sized training sets, capped by the boundary pair count (the scarcest resource). Each boundary pair contributes two samples (one per class). Non-boundary and random sets are sampled to match. Hybrid uses 50% boundary + 50% random from the full shadow set.

Surrogate Training. Surrogates use the same 3-layer GCN architecture (hidden dim 64). Hyperparameters: learning rate 0.001, no dropout, no weight decay, batch size 8, 500 epochs, Adam optimizer. Dropout is disabled because boundary pairs are near-identical graphs with opposite labels; dropout further obscures the subtle structural differences the model must learn.

Test Set Balancing. Fidelity measures how well the surrogate replicates the victim's decision logic, not merely its majority-class tendency. We evaluate on a class-balanced subset of the test set (equal samples per victim-predicted class, fixed seed). Without balancing, a surrogate predicting only the majority class achieves artificially high fidelity. Balancing ensures fidelity reflects agreement on both sides of the decision boundary.

Boundary Detection Statistics. Table 4 reports single-edge boundary counts from the motivating experiment's toggling procedure ($\delta_e = 1$) across all datasets using GCN victims. Each boundary graph produces one pair (original + flipped) = 2 training samples. Low-rate datasets such as Tox21_AhR rely heavily on Phase 2 for boundary pair generation.

Dataset	Shadow	Boundary	Non-Boundary	Rate
AIDS	400	127	273	31.8%
MUTAG	38	11	27	28.9%
NCI1	822	338	484	41.1%
PTC_FM	70	32	38	45.7%
Tox21_AhR	1,634	69	1,565	4.2%

Table 4: Single-edge boundary detection ($\delta_e = 1$, GCN victim). Counts are from the motivating experiment's *toggling* procedure not Phase 1's deletion-only search.

Toggle vs. Deletion Yield. Table 5 compares two single-edge detectors on the same shadow splits and GCN victims: toggling over all $O(|V|^2)$ node pairs (our motivating experiment) versus deletion over the $O(|E|)$ present edges (Phase 1). Insertions reach boundaries deletions cannot, and the gap widens where boundaries are scarce (Tox21_AhR, $3.83\times$). Phase 1 trades this yield for its cheaper $O(|E|)$ cost; Phase 2 recovers the difference on low-yield datasets.

Dataset	Shadow graphs	Toggle (add/remove)	Deletion (remove only)	Ratio
AIDS	400	127 (31.8%)	80 (20.0%)	1.59×
NCI1	822	338 (41.1%)	264 (32.1%)	1.28×
Tox21_AhR	1,634	69 (4.2%)	18 (1.1%)	3.83×

Table 5: Single-edge boundary yield: toggling (add or remove) vs. deletion only, GCN victim. Toggle counts match Table 4; insertions raise yield most where boundaries are scarce.

C.2 Theoretical Analysis of Boundary Data Effectiveness

We first record that the object our search seeks—a single-edge boundary pair—exists for an *arbitrary* victim, with no continuity, smoothness, or Lipschitz assumption. The Lipschitz assumptions introduced afterward bound only how tightly such a pair constrains the surviving hypotheses, not whether it exists.

Lemma C.1 (Existence of single-edge boundary pairs)**.** *Let $f : \mathcal{G} \to [C]$ be any victim classifier on graphs over a fixed node set V, with no assumption on f beyond being a function. Suppose two graphs G, G'' on V satisfy $f(G) \neq f(G'')$, and let $d = \frac{1}{2}\|A(G) - A(G'')\|_0$ be their edge-edit distance. Then there exist a graph H and a single edge e with*

$$\delta_e(G, H) < d, \qquad \delta_e(H, H \oplus e) = 1, \qquad f(H) \neq f(H \oplus e),$$

i.e., a single-edge boundary pair lying on the edit path between G and G''.

Proof. Identifying each graph on V with the upper-triangular part of its adjacency matrix, the graphs form the Boolean hypercube $\{0,1\}^{\binom{|V|}{2}}$ under single-edge toggles, in which any two vertices are

joined by a path that toggles their differing coordinates one at a time. Toggling the d edges on which G and G'' differ yields a sequence $G = H_0, H_1, \ldots, H_d = G''$ with $\delta_e(H_{i-1}, H_i) = 1$ for every i. The label sequence $f(H_0), \ldots, f(H_d)$ starts at $f(G)$ and ends at $f(G'') \neq f(G)$, so some index i satisfies $f(H_{i-1}) \neq f(H_i)$. Setting $H = H_{i-1}$ and e the edge toggled at step i gives $f(H) \neq f(H \oplus e)$ with $\delta_e(G, H) = i - 1 < d$. □

Lemma C.1 certifies that the boundary-pair search is never vacuous: whenever a shadow graph has any differently labeled graph within edit distance d, a single-edge (toggle) boundary pair exists inside that ball—exactly the object the motivating experiment and Phase 2 seek. The remaining results add an L-Lipschitz score assumption only to quantify how strongly a queried pair collapses the version space.

Let $\mathcal{H}$ be the hypothesis class of the extraction model mapping $\mathcal{X} \to \{-1, +1\}$. We assume the victim model $h^* \in \mathcal{H}$ and satisfies L-Lipschitz continuity. Given a set of graph data and corresponding query responses $\mathcal{G}_T = \{(G_t, h^*(G_t))\}$, the victim space $\mathcal{V}_T \subseteq \mathcal{H}$ is the set of all hypotheses consistent with the predicted graph data:

$$\mathcal{V}_T = \{h \in \mathcal{H} | h(G_t) = h^*(G_t), \forall t \in [1, \ldots, T]\}$$

Besides, we define a pair of graph data (G, G') as γ-boundary data if their distance satisfies $\delta_e + \delta_f = \gamma$ and they hold different queried predictions on the victim model, i.e., $h^*(G) \neq h^*(G')$. We also define the disagreement region $DIS(\mathcal{V}_T)$ as the subset of the input graph space $\mathbb{G}$ where the surviving models in the victim space yield conflicting predictions:

$$DIS(\mathcal{V}_T) = \{G \in \mathbb{G} | \exists h_1, h_2 \in \mathcal{V}_T : h_1(G) \neq h_2(G)\}$$

For each $h \in \mathcal{H}$, let $f_h : \mathbb{G} \to \mathbb{R}$ be its score function and $h(G) = \text{sign}(f_h(G))$. We assume every f_h is L-Lipschitz with respect to the graph distance $d_{\mathbb{G}}(G, G') = \delta_e + \delta_f$.

Lemma C.2 (Boundary-pair margin constraint)**.** *Let (G, G') be a queried γ-boundary pair with $d_{\mathbb{G}}(G, G') = \gamma$ and, without loss of generality, $h^*(G) = +1$ and $h^*(G') = -1$. Suppose every hypothesis $h \in \mathcal{H}$ is induced by an L-Lipschitz score function f_h. Then any surviving hypothesis $h \in \mathcal{V}_{T+2}$ satisfies*

$$0 < f_h(G) < L\gamma, \qquad -L\gamma < f_h(G') < 0.$$

Thus, for sufficiently small γ, every surviving hypothesis must place its decision boundary inside an $O(\gamma)$ neighborhood between G and G'.

Proof. Since $h \in \mathcal{V}_{T+2}$ is consistent with the newly queried labels, we have $h(G) = h^*(G) = +1$ and $h(G') = h^*(G') = -1$. Hence $f_h(G) > 0$ and $f_h(G') < 0$. By L-Lipschitz continuity,

$$|f_h(G) - f_h(G')| \leq L d_{\mathbb{G}}(G, G') = L\gamma.$$

Because $f_h(G) > 0$ and $f_h(G') < 0$, we have $f_h(G) - f_h(G') > f_h(G)$ and $f_h(G) - f_h(G') > -f_h(G')$. Therefore,

$$0 < f_h(G) < L\gamma, \qquad -L\gamma < f_h(G') < 0.$$

□

This lemma gives the local mechanism by which boundary pairs constrain the version space: a tightly coupled pair with opposite victim labels forces every surviving hypothesis to have scores close to the decision threshold at both endpoints. Consequently, hypotheses whose decision boundary does not enter an $O(\gamma)$ neighborhood of the queried pair become inconsistent with the observed victim responses and are removed.

To give a further bound on extraction sample complexity, we define $B(h^*, r)$ be the ball of hypotheses with an error rate of at most r,

$$B(h^*, r) = \left\{h \in \mathcal{H} : \mathbb{P}_{G \sim \mathcal{D}}\big(h(G) \neq h^*(G)\big) \leq r\right\}.$$

and a disagreement coefficient $\theta_{h^*}(\epsilon)$ bounds the rate of version space collapse:

$$\theta_{h^*}(\epsilon) = \sup_{r \geq \epsilon} \frac{\mathbb{P}_{G \sim \mathcal{D}}(G \in DIS(B(h^*, r)))}{r}$$

We assume that the boundary-pair sampler is non-degenerate in the following sense: at each round, the queried γ-boundary pairs cover the current disagreement region. More precisely, letting $A_T = DIS(\mathcal{V}_T)$, the marginal distribution of queried boundary endpoints dominates the conditional distribution $\mathcal{D}(\cdot \mid A_T)$ up to a universal constant. This prevents the algorithm from repeatedly querying the same boundary pair and ensures that boundary queries explore the informative part of the version space.

Theorem C.3 (Sample Complexity of Boundary Extraction)**.** *Assume realizability, $h^* \in \mathcal{H}$, and suppose the boundary-pair sampler is non-degenerate: at epoch k, if $A_k = DIS(\mathcal{V}_k)$, the endpoint distribution Q_k of the queried boundary pairs satisfies*

$$Q_k(S) \geq c_0\, \mathcal{D}(S \mid A_k)$$

for every measurable $S \subseteq A_k$ and for some constant $c_0 > 0$. Then, with probability at least $1-\delta$, the number of queried endpoints needed to obtain a hypothesis with generalization error at most ϵ is

$$N = O\left(\frac{\theta_{h^*}(\epsilon)}{c_0}\left(d_{VC}\log\frac{\theta_{h^*}(\epsilon)}{c_0} + \log\frac{\log(1/\epsilon)}{\delta}\right)\log\frac{1}{\epsilon}\right).$$

In particular, when c_0 and logarithmic factors are suppressed,

$$N = \widetilde{O}\left(\theta_{h^*}(\epsilon) d_{VC}\log\frac{1}{\epsilon}\right).$$

Counting boundary pairs instead of endpoints changes the bound only by a constant factor.

Proof. Let

$$\mathrm{err}(h) = \mathbb{P}_{G\sim\mathcal{D}}\big(h(G) \neq h^*(G)\big), \qquad r_T = \max_{h\in\mathcal{V}_T}\mathrm{err}(h).$$

Since the realizability assumption gives $h^* \in \mathcal{V}_T$, for any $h \in \mathcal{V}_T$ and any graph G such that $h(G) \neq h^*(G)$, the graph G must belong to the disagreement region $DIS(\mathcal{V}_T)$. Therefore,

$$\{G : h(G) \neq h^*(G)\} \subseteq DIS(\mathcal{V}_T).$$

Moreover, by definition of r_T, we have $\mathcal{V}_T \subseteq B(h^*, r_T)$. Hence

$$DIS(\mathcal{V}_T) \subseteq DIS(B(h^*, r_T)).$$

By the definition of the disagreement coefficient, for any $r_T \geq \epsilon$,

$$\mathbb{P}_{G\sim\mathcal{D}}\big(G \in DIS(\mathcal{V}_T)\big) \leq \mathbb{P}_{G\sim\mathcal{D}}\big(G \in DIS(B(h^*, r_T))\big) \leq \theta_{h^*}(\epsilon) r_T.$$

Denote $\theta = \theta_{h^*}(\epsilon)$ and $A_T = DIS(\mathcal{V}_T)$. Consider any hypothesis $h \in \mathcal{V}_T$ whose error is larger than $r_T/2$. Since its disagreement set with h^* is contained in A_T, we have

$$\mathbb{P}_{G\sim\mathcal{D}}\big(h(G) \neq h^*(G) \mid G \in A_T\big) = \frac{\mathbb{P}_{G\sim\mathcal{D}}\big(h(G) \neq h^*(G)\big)}{\mathbb{P}_{G\sim\mathcal{D}}(G \in A_T)} \geq \frac{r_T/2}{\theta r_T} = \frac{1}{2\theta}.$$

Thus, conditioned on the current disagreement region, every hypothesis with error larger than $r_T/2$ disagrees with the victim model on at least a $1/(2\theta)$ fraction of the informative region.

Now let Q_T be the marginal distribution of queried boundary endpoints at round T. By the non-degeneracy assumption, for any measurable $S \subseteq A_T$,

$$Q_T(S) \geq c_0\, \mathcal{D}(S \mid A_T).$$

For any hypothesis $h \in \mathcal{V}_T$ with $\mathrm{err}(h) > r_T/2$, define

$$S_h = \{G : h(G) \neq h^*(G)\}.$$

Since $S_h \subseteq A_T$, the previous inequality gives

$$Q_T(S_h) \geq c_0\, \mathcal{D}(S_h \mid A_T) \geq \frac{c_0}{2\theta}.$$

Therefore, under the endpoint query distribution Q_T, every hypothesis with error larger than $r_T/2$ disagrees with the victim on a set of probability at least $c_0/(2\theta)$.

Apply the VC ϵ-net theorem to the class

$$\mathcal{A} = \big\{\{G : h(G) \neq h^*(G)\} : h \in \mathcal{H}\big\},$$

whose VC dimension is $O(d_{VC})$. A batch of

$$m = O\bigg(\frac{\theta}{c_0}\left(d_{VC}\log\frac{\theta}{c_0} + \log\frac{1}{\delta_T}\right)\bigg)$$

queried endpoints forms an ϵ-net for all sets in $\mathcal{A}$ of Q_T-measure at least $c_0/(2\theta)$ with probability at least $1-\delta_T$. Hence every surviving hypothesis with error larger than $r_T/2$ is hit by at least one queried endpoint and is eliminated. Therefore, after this batch,

$$r_{T+m} \leq \frac{r_T}{2}.$$

Starting from $r_0 \leq 1$, after

$$K = \left\lceil \log_2 \frac{1}{\epsilon} \right\rceil$$

epochs, the version-space radius satisfies $r_K \leq \epsilon$. Therefore the total number of queried boundary endpoints is

$$\begin{aligned} N &= \sum_{k=1}^{K} O\bigg(\theta_{h^*}(\epsilon)\left(d_{VC}\log\theta_{h^*}(\epsilon) + \log\frac{K}{\delta}\right)\bigg) \\ &= O\bigg(\theta_{h^*}(\epsilon)\left(d_{VC}\log\theta_{h^*}(\epsilon) + \log\frac{\log(1/\epsilon)}{\delta}\right)\log\frac{1}{\epsilon}\bigg). \end{aligned}$$

Suppressing logarithmic confidence factors and the mild $\log\theta_{h^*}(\epsilon)$ term gives

$$N = O\bigg(\theta_{h^*}\cdot d_{VC}\log\frac{1}{\epsilon}\bigg).$$

Since each γ-boundary pair contributes two labeled endpoints, counting queries by boundary pairs only changes the bound by a constant factor. This completes the proof. □

D Algorithms

D.1 Overall Extraction Procedure

Let $\bar{\mathcal{G}}$ denote the set of boundary pairs obtained by applying the two-phase search procedure to all initial graphs:

$$\bar{\mathcal{G}} \;=\; \big\{(G, G') : y(G) \neq y(G'),\ \delta_e(G, G') \leq \delta\big\}. \tag{10}$$

We construct a training set for the surrogate model g:

$$\mathcal{D}_{\text{train}} \;=\; \big\{(\bar{G}, y(\bar{G})) : \forall \bar{G} \in (G, G'),\ (G, G') \in \bar{\mathcal{G}}\big\}. \tag{11}$$

The surrogate is trained via

$$\min_{g\in\mathcal{H}} \sum_{(G,y)\in\mathcal{D}_{\text{train}}} \ell\big(g(G), y\big), \tag{12}$$

where $\mathcal{H}$ denotes the chosen GNN hypothesis class.

Algorithm 1 presents the complete XSTEAL procedure, combining Phase 1 (exhaustive search) and Phase 2 (explanation-guided MC search).

Algorithm 1 XSTEAL: eXplanation-guided STEALing

Require: Victim f, explainer h, shadow set $\mathcal{G}_{\text{shadow}}$, edge threshold δ, query budget Q
Ensure: Boundary dataset $\bar{\mathcal{G}}$, surrogate g

$\bar{\mathcal{G}} \leftarrow \emptyset, \quad \mathcal{R} \leftarrow \emptyset$ // $\mathcal{R}$: non-boundary graphs
// Phase 1: exhaustive search
for all $G \in \mathcal{G}_{\text{shadow}}$ **do**
  for all $e \in E(G)$ **do**
    $G' \leftarrow G \ominus e$
    if $y(G') \neq y(G)$ **then**
      $\bar{\mathcal{G}} \leftarrow \bar{\mathcal{G}} \cup \{(G, G')\}$; **break**
    end if
  end for
  if no boundary found **then**
    $\mathcal{R} \leftarrow \mathcal{R} \cup \{G\}$
  end if
end for
// Phase 2: explanation-guided MC search
for all $G \in \mathcal{R}$ **while** $Q > 0$ **do**
  $(G', Q_c) \leftarrow$ BOUNDARYSEARCH(G, f, h, δ) // Alg. 2
  $Q = Q - Q_c$
  if $G' \neq \texttt{null} \wedge Q \geq 0$ **then**
    $\bar{\mathcal{G}} \leftarrow \bar{\mathcal{G}} \cup \{(G, G')\}$
  end if
end for
Train surrogate g on $\{(\bar{G}, f(\bar{G})) : \bar{G} \in \bar{\mathcal{G}}\}$
return $\bar{\mathcal{G}}, g$

D.2 BOUNDARY SEARCH SUBROUTINE

Algorithm 2 provides the complete BOUNDARYSEARCH subroutine referenced in Algorithm 1.

The search (Algorithm 2) exits on exactly one of four conditions: **(i)** *success*, when $y(G^{(t)}) \neq y^{(0)}$ within the edit budget δ; **(ii)** *edit-budget exhaustion*, when $\delta_e(G^{(0)}, G^{(t)}) \geq \delta$; **(iii)** *stagnation*, when no candidate edge has positive estimated sensitivity ($S^{(t)} = \emptyset$); and **(iv)** the *iteration cap* $t = T_{\max}$. Condition (iv) alone bounds the search at $T_G \leq T_{\max} = 10$ iterations unconditionally, regardless of δ, the explainer, or the victim, so a stopping iteration $t^\star$ always exists and the procedure always halts. It does not always halt successfully: exits (ii) and (iii) abandon the graph by design since Phase 2 is a per-graph budgeted attempt and the attacker proceeds to the next shadow graph.

E EXPERIMENTAL DETAILS

We provide the details for our experiments here. The implementation code is available at: `https://anonymous.4open.science/r/GNN-MEA-288B/`.

E.1 DATASETS

We evaluate on graph classification benchmarks from the TUDataset collection (Morris et al., 2020) and the GNN Benchmark suite (Dwivedi et al., 2022), with several datasets originating from the IAM Graph Database (Riesen & Bunke, 2008). Table 6 summarizes dataset statistics.

Dataset Selection Rationale. Our benchmark suite spans three key dimensions of diversity:

- **Domain diversity:** AIDS, MUTAG, NCI1, and PTC_FM represent molecular chemistry (drug discovery and toxicity screening), Tox21_AhR encodes toxicological assay activity (regulatory safety testing), MNIST captures handwritten digit recognition via superpixel graphs (Monti et al., 2016) (computer vision), Letter-low captures handwritten letter recognition from the IAM Graph Database (Riesen & Bunke, 2008) (computer vision), and Synthie provides a synthetic benchmark with controlled properties and rich node attributes.
- **Classification complexity:** We include five binary classification datasets (AIDS, MUTAG, NCI1, PTC_FM, Tox21_AhR) where a single decision boundary allows all boundary pairs to reinforce

Algorithm 2 BOUNDARYSEARCH: Find boundary pair for a single graph

Require: Graph G, victim f, explainer h, edge threshold Δ_e, max iterations $T_{\max}$, edges per step k, MC samples n_e, perturbation probability p_{flip}, carry-over size $|\mathcal{S}_{\text{carry}}|$, random exploration size $|\mathcal{S}_{\text{rand}}|$
Ensure: Boundary graph G' and Query Cost Q_c

$G^{(0)} \leftarrow G, \quad y^{(0)} \leftarrow f(G), \quad t \leftarrow 0, \quad \mathcal{S}_{\text{carry}} \leftarrow \emptyset$
while $t < T_{\max}$ **do**
  $y^{(t)} \leftarrow f(G^{(t)})$
  $Q_c \leftarrow Q_c + 1$
  $\delta_e \leftarrow \frac{1}{2}\|A(G^{(0)}) - A(G^{(t)})\|_0$
  if $y^{(t)} \neq y^{(0)}$ **and** $\delta_e \leq \Delta_e$ **then**
    return $(G^{(t)}, Q_c)$
  end if
  if $\delta_e \geq \Delta_e$ **then**
    break
  end if
  $G^{(t)}_{\text{sub}} \leftarrow h(G^{(t)}, y^{(t)})$ // query explainer
  Construct candidate edges $\mathcal{C}(G^{(t)})$ via Eq. equation 9
  for all $e \in \mathcal{C}(G^{(t)})$ **do**
    Estimate $\hat{g}_e^{(t)}$ via Eq. equation 4 with n_e samples
    $Q_c \leftarrow Q_c + 2n_e$
  end for
  $S^{(t)} \leftarrow \text{TopK}_{\max}(\{\hat{g}_e^{(t)} : \hat{g}_e^{(t)} > 0\}, k)$
  if $S^{(t)} = \emptyset$ **then**
    break
  end if
  $G^{(t+1)} \leftarrow$ flip edges in $S^{(t)}$ // apply Eq. equation 2 to each $e \in S^{(t)}$
  $\mathcal{S}_{\text{carry}} \leftarrow$ top-$|\mathcal{S}_{\text{carry}}|$ edges from $S^{(t)}$ by sensitivity
  $t \leftarrow t + 1$
end while
return (`null`,Q_c)

Dataset	#Graphs	#Classes	Avg. $\lvert V\rvert$	Avg. $\lvert E\rvert$	Features
AIDS	2,000	2	15.7	16.2	38
Letter-low	2,250	15	4.7	3.1	2
MNIST	70,000	10	70.6	282.3	3
MUTAG	188	2	17.9	19.8	7
PTC_FM	349	2	14.1	14.5	14
Synthie	400	4	95.0	172.9	15
NCI1	4,110	2	29.9	32.3	37
Tox21_AhR	8,169	2	18.1	18.5	50

Table 6: Dataset statistics.

the same decision surface, and three multi-class datasets (Synthie with 4 classes, MNIST with 10 classes, and Letter-low with 15 classes) to evaluate scalability of boundary search across increasingly complex decision boundary geometries.

- **Scale diversity:** Dataset sizes range from 188 graphs (MUTAG) to 70,000 graphs (MNIST), testing our method under both data-scarce and data-rich regimes. Graph sizes range from very small (Letter-low: $\sim$5 nodes, where single edge flips constitute $\sim$30% structural change) to large dense graphs (MNIST: $\sim$71 nodes, $\sim$282 edges) and large sparse graphs (Synthie: $\sim$95 nodes), testing our method's efficiency across scales. Feature representations vary from rich atom-level attributes (AIDS: 38 features, Tox21: 50 features) to minimal spatial coordinates (Letter-low: 2 features) to superpixel descriptors (MNIST: 3 features), testing robustness to feature informativeness.

E.2 EVALUATION METRICS

We report two primary metrics:

Architecture	Dataset	Hidden Dim	Selected Epochs	Test Accuracy (%)
GCN	AIDS	128	1000	87.0
	Letter-low	128	1000	86.2
	MNIST	128	200	38.1
	MUTAG	64	300	78.9
	PTC_FM	64	500	65.7
	Synthie	128	700	65.0
	NCI1	64	1000	67.3
	Tox21_AhR	64	700	89.0
GAT	AIDS	128	1000	88.5
	Letter-low	64	700	98.0
	MNIST	128	500	59.6
	MUTAG	64	300	78.9
	PTC_FM	64	300	67.1
	Synthie	128	300	75.0
	NCI1	64	1000	69.6
	Tox21_AhR	64	700	89.5
GraphSAGE	AIDS	128	500	90.5
	Letter-low	128	700	99.3
	MNIST	128	500	54.7
	MUTAG	64	300	76.3
	PTC_FM	64	700	64.3
	Synthie	128	1000	93.8
	NCI1	128	1000	73.6
	Tox21_AhR	64	300	89.5

Table 7: Victim model configurations. Best hidden dimension and epochs selected per dataset via grid search.

- **Fidelity**: Agreement rate between surrogate and victim predictions on a class-balanced test set: Fidelity $= \frac{1}{|\mathcal{D}_{\text{test}}|} \sum_{G \in \mathcal{D}_{\text{test}}} \mathbf{1}[g(G) = f(G)]$.
- **Accuracy**: Classification accuracy of the surrogate on ground-truth test labels.

Fidelity is our primary metric as it measures how faithfully the surrogate replicates the victim's decision logic, independent of the victim's own correctness.

E.3 Victim Model Configuration

We employ three representative GNN architectures as victim models: Graph Convolutional Network (GCN), Graph Attention Network (GAT), and GraphSAGE. Each architecture consists of three message-passing layers followed by a global mean pooling operation and a linear classifier. We use ReLU activations and apply dropout ($p = 0.5$) between layers for regularization. For each architecture-dataset combination, we perform a hyperparameter search over hidden dimensions $\in \{64, 128\}$ and training epochs $\in \{300, 500, 700, 1000\}$, selecting the configuration achieving highest test accuracy. All models are trained using the Adam optimizer with learning rate 0.001, weight decay 5e-4, and batch size 32. Table 7 summarizes the selected configurations and test accuracies across all architectures and datasets.

E.4 Explainer Configuration

PGExplainer is trained on the victim model for 100 epochs with learning rate 0.003. Explanations are thresholded at the median importance score to produce binary edge masks. GNNExplainer uses 100 optimization steps per instance with node masks (`node_mask_type='object'`). Both explainer types operate on a `deepcopy` of the victim model to prevent gradient hooks from corrupting victim predictions during the experiment (see Appendix E.9).

E.5 Attack Configuration

Default hyperparameters for Phase 2 (explanation-guided MC search): $T_{\max} = 10$ (maximum iterations), $\delta_{\max} = 20$ (maximum edge edits), $k = 3$ (edges flipped per step), $n_e = 5$ (Monte Carlo samples per candidate edge), $p_{\text{flip}} = 0.05$ (perturbation probability), $|\mathcal{S}_{\text{carry}}| = 5$ (carry-over edges from previous iteration), $|\mathcal{S}_{\text{rand}}| = 10$ (random exploration edges per iteration). Explanation masks are thresholded at the median importance score to select important nodes/edges.

E.6 Surrogate Training

Surrogates use the same architecture family as the victim (GCN, GAT, or GraphSAGE) with fixed hyperparameters across all methods: hidden dimension 64, 3 layers, no dropout, learning rate 0.001, 500 epochs, batch size 8, Adam optimizer. Hyperparameters are deliberately held constant to isolate the effect of training data quality—the only variable across methods is how training samples are selected and generated. This reflects realistic black-box constraints where the attacker does not know the victim's exact configuration.

E.7 Data Splits and Query Budget

Each dataset is split 60% victim training / 20% shadow / 20% test, stratified by class labels. The shadow set serves as the attacker's query pool. For main results, we set the query budget to 70% of the shadow set and match all methods to the same training set size, determined by boundary pair availability (the scarcest resource). Table 8 reports the matched n_{train} for each dataset and victim architecture. This ensures fair comparison by isolating the effect of training data quality rather than quantity.

Boundary's actual training set size (n_{train}) may fall below the query budget when the shadow set contains insufficient boundary pairs. To ensure fair comparison, baseline methods are evaluated at the budget level yielding the closest n_{train} to Boundary's output.

E.8 Compute Resources

All experiments were conducted on Google Colab Pro+ using a single NVIDIA T4 GPU with high-RAM runtime. The full experimental pipeline (including victim training, boundary search, surrogate training, and evaluation across all datasets, architectures, and explainers) required approximately 600 Colab compute units.

Dataset	GCN	GAT	GraphSAGE
AIDS	280	280	280
Letter-low	122	82	116
MNIST	272	88	84
MUTAG	26	26	26
PTC_FM	48	48	48
Synthie	16	24	8
NCI1	574	574	575
Tox21_AhR	384	1,142	768

Table 8: Matched training set size (n_{train}) per dataset and victim architecture.

E.9 Implementation Notes

Data Generation. Boundary and Hybrid are *generative* methods: the flipped graph in each boundary pair is a synthetic sample created by edge perturbation, not an existing shadow graph. In contrast, Shadow-Only, Non-boundary, and EGSteal-BB only relabel existing graphs from the shadow set. This data augmentation around decision boundaries is a key mechanism behind the fidelity improvement, and enables our methods to generate informative training data even on very limited shadow datasets.

Explainer Model Isolation. PyTorch Geometric's explainer frameworks (GNNExplainer, PGExplainer) register forward hooks on the model that modify internal state in-place. Running an explainer on the victim model can corrupt its predictions for subsequent queries. We resolve this by creating a `copy.deepcopy` of the victim model for each explainer instantiation, ensuring the original victim remains unmodified throughout the experiment.

Test Set Balancing. Fidelity measures how well the surrogate replicates the victim's decision *logic*—its learned mapping from graph structure to class label—not merely its majority-class tendency. We evaluate on a class-balanced subset of the test set. Without balancing, a surrogate predicting only the majority class achieves artificially high fidelity. Balancing ensures fidelity reflects agreement on both sides of the decision boundary, with single-class prediction scoring exactly 50%.

E.10 Baseline Details

Shadow-Only. Samples graphs from the shadow dataset and queries the victim for labels, with no explanation access or boundary-seeking strategy. This represents the standard model extraction baseline where training data is drawn from the attacker's available distribution without any targeted selection.

Non-boundary. Trained exclusively on shadow graphs where no single-edge removal changes the victim's prediction (identified during Phase 1). These graphs lie in the interior of decision regions, far from boundaries. This baseline tests whether interior samples alone suffice for extraction, and serves as a direct contrast to boundary-based methods.

EGSteal-BB. We adapt EGSteal (Ma et al., 2025) to our discrete setting. The original method requires continuous importance vectors $E \in \mathbb{R}^{|V|}$ for rank-based explanation alignment, which is infeasible under binary masks. We retain only its style-invariance augmentation: perturbing edges outside G_{sub} while assigning the original victim label, generating 2 augmented samples per graph. This assumes G_{sub} causally determines the prediction, so style perturbations are label-preserving. When explainers produce imprecise masks, this assumption breaks down and the augmented samples introduce label noise, which can degrade fidelity below naive random sampling (see case study in Appendix F.6).

Boundary (XSTEAL). Our full two-phase method (Section 3.4). Phase 1 performs cheap exhaustive $\delta_e = 1$ edge removal search; Phase 2 triggers explanation-guided MC search when Phase 1 produces insufficient pairs. Both original and flipped graphs are added to the training set with their respective victim predictions. Boundary is a *generative* method: the flipped graph in each pair is a synthetic sample created by edge perturbation, not an existing shadow graph.

Hybrid (XSTEAL). Allocates 50% of the query budget to boundary pair generation and 50% to random sampling. This combines the complementary strengths of both approaches: boundary pairs provide targeted information about the victim's decision logic near class transitions, while random samples provide broad coverage of the input distribution. Hybrid mitigates the sample limitation of pure Boundary on datasets with low boundary rates.

F Additional Results

We conduct ablations across three dimensions to assess robustness and isolate component contributions.

Victim Architecture Ablation. We evaluate our method across all three victim architectures to assess generalization:

- GCN (Kipf & Welling, 2017): Graph Convolutional Networks (main text, Table 1).
- GAT (Veličković et al., 2018): Graph Attention Networks with learned attention weights (Section F.1).
- GraphSAGE (Hamilton et al., 2018): Sampling and aggregation-based architecture (Section F.2).

Explainer Ablation. We evaluate robustness to the explanation method deployed by the model owner by comparing two representative explainers:

- **GNNExplainer** (Ying et al., 2019): Instance-level optimization-based explanations using node masks.
- **PGExplainer** (Luo et al., 2020): Parameterized explainer trained across graphs using edge masks.

This ablation tests whether our method's gains are tied to a specific explanation format or generalize across explainer types. Performance differences between explainers reflect variation in the informativeness of leaked explanation structure rather than a limitation of our method.

Component Ablations. We evaluate two ablations isolating key contributions of our method:

- **No-Exp**: Boundary search with random edge candidate selection instead of explanation-guided selection (Eq. 9). Candidate edges are sampled uniformly at random from all $\binom{|V|}{2}$ possible node pairs, with the candidate-set size matched to the explanation-guided set per graph and per iteration, and both methods run at the default attack hyperparameters ($\delta_{\max} = 20$, $T_{\max} = 10$, $n_e = 5$, $k = 3$) so that only the candidate selection differs. This isolates the contribution of the explainer; full results are in Appendix F.4.
- **No-MC**: Boundary search with deterministic single-sample estimation ($n_e = 1$, $p_{\text{flip}} = 0$) instead of Monte Carlo sampling ($n_e = 5$, $p_{\text{flip}} = 0.05$). This isolates the contribution of stochastic estimation.

Multi-class Performance. On multi-class datasets (Letter-low, MNIST, Synthie), Hybrid consistently outperforms pure Boundary, as boundary search fragments across multiple decision surfaces and random coverage provides complementary distributional information. Absolute fidelity on MNIST (55.1%) and Synthie (51.3%) is notably lower than on binary tasks. However, these figures should be contextualized against the number of classes: with 10 and 4 classes respectively, uniform-chance fidelity is 10% and 25%, making the achieved rates substantially above random agreement. While we notice that there is a performance gap between binary and multi-class settings, this observation is consistent with the fragmentation effect induced by our class-agnostic estimator (Section 3.1): in binary classification, all boundary pairs reinforce a single decision surface, while in multi-class settings the same query budget must be distributed across $\binom{C}{2}$ pairwise boundaries, diluting per-surface coverage. This analysis also suggests that scaling the query budget proportionally to the number of class boundaries is a promising direction for future work.

F.1 GAT Victim Results

		Shadow		Non-boundary		EGSteal-BB		Boundary (XSTEAL)		Hybrid (XSTEAL)	
Dataset	**Expl.**	Fid.	Acc.	Fid.	Acc.	Fid.	Acc.	Fid.	Acc.	Fid.	Acc.
AIDS	GNNExp	86.9	76.1	68.2	72.2	67.0	61.9	**89.8**	75.6	86.9	72.7
	PGExp					64.2	62.5	88.6	77.8	85.8	75.0
Letter-low	GNNExp	78.3	77.6	71.7	71.2	82.9	82.6	73.6	72.9	81.7	81.7
	PGExp					84.0	83.8	81.7	81.4	**86.0**	86.4
MNIST	GNNExp	33.6	27.6	36.1	29.4	35.9	28.9	28.9	24.8	33.4	27.4
	PGExp					31.6	26.4	28.5	24.5	**37.2**	30.7
MUTAG	GNNExp	81.8	68.2	86.4	63.6	86.4	72.7	**95.5**	72.7	**95.5**	72.7
	PGExp					90.9	68.2	**95.5**	72.7	**95.5**	72.7
PTC_FM	GNNExp	85.7	57.1	64.3	50.0	64.3	35.7	71.4	42.9	**92.9**	64.3
	PGExp					57.1	28.6	64.3	35.7	85.7	57.1
Synthie	GNNExp	38.2	44.1	36.8	30.9	30.9	27.9	38.2	44.1	29.4	32.4
	PGExp					38.2	32.4	32.4	38.2	**39.7**	26.5
NCI1	GNNExp	95.0	68.0	90.8	67.7	87.2	66.5	93.2	68.3	**95.5**	69.0
	PGExp					79.5	64.0	**95.5**	69.0	93.5	68.8
Tox21	GNNExp	91.3	80.2	50.8	66.7	80.2	70.6	**97.6**	80.2	93.7	77.8
	PGExp					55.6	63.5	93.7	77.8	92.9	80.2

Table 9: Extraction results on *GAT* victim at matched training set sizes. Fid. = Fidelity (%), Acc. = Accuracy (%). Best fidelity per dataset in **bold**.

F.2 GRAPHSAGE VICTIM RESULTS

Dataset	Expl.	Shadow Fid.	Shadow Acc.	Non-boundary Fid.	Non-boundary Acc.	EGSteal-BB Fid.	EGSteal-BB Acc.	Boundary (XSTEAL) Fid.	Boundary (XSTEAL) Acc.	Hybrid (XSTEAL) Fid.	Hybrid (XSTEAL) Acc.
AIDS	GNNExp	88.7	75.6	72.0	74.4	70.2	65.5	84.5	75.0	83.9	76.8
	PGExp					72.0	67.3	88.7	80.4	**90.5**	79.8
Letter-low	GNNExp	**89.2**	89.0	**89.2**	89.0	86.2	85.8	88.5	88.3	**89.2**	89.0
	PGExp					88.7	88.5	85.8	85.5	81.6	81.6
MNIST	GNNExp	35.6	26.2	35.0	24.3	37.6	26.6	30.9	21.0	35.2	26.3
	PGExp					35.9	25.8	28.4	19.4	**39.5**	28.7
MUTAG	GNNExp	**100.0**	80.0	85.0	65.0	85.0	75.0	**100.0**	80.0	**100.0**	80.0
	PGExp					95.0	75.0	**100.0**	80.0	**100.0**	80.0
PTC_FM	GNNExp	73.3	60.0	63.3	63.3	53.3	46.7	83.3	50.0	90.0	56.7
	PGExp					53.3	46.7	**93.3**	53.3	86.7	53.3
Synthie	GNNExp	26.3	28.9	23.7	25.0	28.9	28.9	19.7	19.7	**32.9**	36.8
	PGExp					28.9	28.9	21.1	19.7	21.1	25.0
NCI1	GNNExp	90.2	72.1	89.2	71.4	86.2	69.8	77.6	63.2	**91.4**	72.1
	PGExp					84.5	67.0	90.4	72.0	91.0	71.4
Tox21	GNNExp	85.9	74.4	50.0	59.0	55.1	56.4	**94.9**	83.3	**94.9**	80.8
	PGExp					50.0	59.0	80.8	71.8	93.6	79.5

Table 10: Extraction results on *GraphSAGE* victim at matched training set sizes. Fid. = Fidelity (%), Acc. = Accuracy (%). Best fidelity per dataset in **bold**.

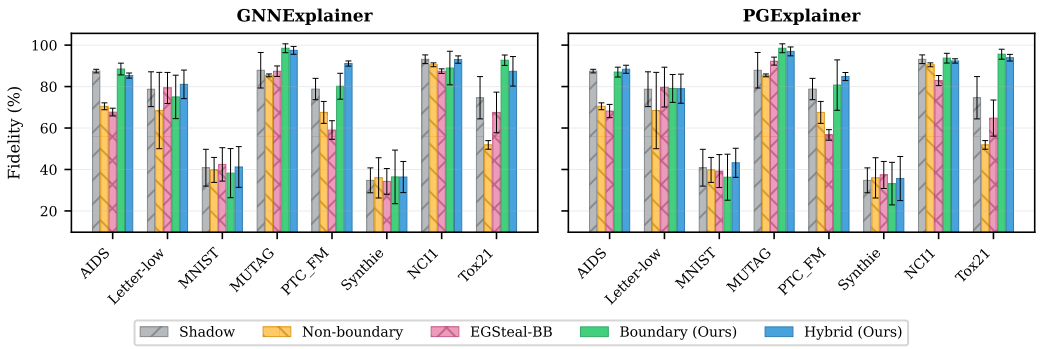


Figure 7: Extraction fidelity (%) across all architectures and datasets, grouped by explainer (left: GNNExplainer, right: PGExplainer).

F.3 MONTE CARLO SAMPLING ABLATION

Table 11 isolates the contribution of Monte Carlo sampling. The contribution of explanation guidance is isolated separately by the matched No-Exp control (Appendix F.4). Moreover, we investigate No-MC's empirical details and found in most cases the boundary search is not able to find a boundary pair till exceeding the iteration limit. The lack of sufficient boundary samples thus further leads to the de-performance. This finding highlights the necessity and effectiveness of the Monte Carlo sampling.

Explainer	Method	AIDS Fid.	AIDS Acc.	NCI1 Fid.	NCI1 Acc.	Tox21 Fid.	Tox21 Acc.
GNNExp.	Boundary (XSTEAL)	91.1	75.9	96.0	66.3	89.3	83.3
	No-MC	77.6	68.1	94.7	66.5	89.3	78.6

Table 11: Component ablation at 70% budget (GCN victim). No-MC removes Monte Carlo sampling.

F.4 MATCHED QUERIES NO-EXP CONTROL

To isolate the contribution of explanation guidance from raw computational budget, we compare Boundary against a No-Exp control that is identical except that candidate edges are drawn uniformly at random from all node pairs rather than from the explanation subgraph (Eq. 9). Both methods run at the default attack hyperparameters with candidate-set size matched per graph and per iteration, so the two methods differ only in which node pairs are scored. We report fidelity both at matched training-set size n_{train} and at equal total victim query count Q.

Dataset	Budget	Boundary	No-Exp	Hybrid
PTC_FM	matched $n_{\text{train}} = 60$	100.0	89.3	82.1
	matched $Q = 24{,}184$	100.0	89.3	92.9
NCI1	matched $n_{\text{train}} = 560$	96.0	95.7	96.5
	matched $Q = 203{,}150$	95.7	95.6	96.9
Tox21_AhR	matched $n_{\text{train}} = 100$	96.4	85.7	82.1
	matched $Q = 609{,}076$	96.4	85.7	100.0

Table 12: Matched No-Exp control. Fidelity (%). GCN victim, GNNExplainer, one seed, class-balanced test sets.

Explanation guidance improves fidelity over the matched control on all three datasets under both accountings, with margins of $+10.7$, $+0.3$, and $+10.7$ points on PTC_FM, NCI1, and Tox21_AhR at matched n_{train}. The near-tie on NCI1 is expected: its single-edge deletion yield in Phase 1 is the highest of the three so Phase 2 (only stage using explanations) supplies little of the training set and there is little room for guidance to help. On PTC_FM and Tox21_AhR, Phase 1 yields far fewer boundary pairs and Phase 2 does most of the work where guidance matters. This also explains the low Phase-2 success rate on Tox21_AhR: the fraction of attempted graphs that yields a boundary pair is 0.098 against 0.686 on PTC_FM and 0.590 on NCI1, reflecting how scarce boundary pairs are on Tox21_AhR rather than a weakness of guidance.

F.5 PHASE ABLATION

Table 13 compares Phase 1 alone (training only on $\delta_e = 1$ boundary pairs, capped by boundary availability) against Phase 2 alone (explanation-guided MC search on all shadow graphs) on the GCN victim for three datasets. Phase 2 outperforms Phase 1.

	Boundary		Hybrid	
Dataset	P1	P2	P1	P2
AIDS	80.2	**85.3**	85.3	**91.4**
NCI1	94.9	**95.2**	93.8	**95.1**
Tox21_AhR	71.4	**96.4**	85.7	**92.9**

Table 13: Phase ablation (GCN + GNNExp): Fidelity (%) for Phase 1 alone vs. Phase 2 alone.

F.6 EGSTEAL-BB FAILURE ANALYSIS

In Section 4.2, EGSteal-BB underperforms Shadow-Only on 4 of 8 datasets (GCN/GNNExplainer). This notable insufficiency is consistent with its origin: EGSteal-BB is designed for white-box settings with continuous importance vectors, and our discrete adaptation necessarily strips its core alignment mechanism, leaving only style-invariance augmentation. This augmentation assumes the explanation mask causally partitions label-relevant from label-irrelevant structure; while under imprecise binary masks, this assumption breaks and the augmented samples introduce systematic label noise, which latter leads to additional fidelity loss in surrogate training (detailed below).

In our general evaluation we noticed that the performance of EGSteal-BB sometimes catastrophically fails, even worse than Shadow. After further analysis, we conclude that this fail is caused by two

reasons. First we note the EGSteal-BB is an adapted version for fitting our black-box model settings, and therefore removed with a few components of it which relies on the white-box settings, largely reducing its applied performance. This performance decrease validates our statement on its lack of generalization on black-box settings. Besides, we also find a partial reason, the lack of generalizability of the stated assumption on the causal relationship between the original graph and the generated graph. Specifically, in EGSteal-BB, a generated graph containing the explanation subgraph of the original graph, is assumed to always hold the same prediction as the original graph, since the subgraph is the main cause for the prediction. However, this causal assumption is based on 100% belief on a extremely accurate graph explainer, which is unlikely to happen in practice. To validate this statement, we pick out the generated graphs and compare the fidelity of its assigned label, i.e., if the victim model truly follow these labels. The results are provided as below:

Data.	AIDS	MUTAG	NCI1	Tox21	PTC	Letter	Syn	MNIST
Fidelity	99.8%	96.8%	96.3%	99.9%	100.0%	96.7%	99.1%	93.3%

Table 14: Fidelity of the generated training data by EGSteal-BB.

We additionally note that all other baselines, including our method, are trained on 100% fidelity sampled data on all datasets. Therefore, these not-fully-faithful generated data add more fidelity loss at the training stage, which leads to its worse performance compared with other baselines.

G Fidelity vs. Query Budget Trajectories

Figure 8 shows surrogate fidelity as a function of training set size (n_{train}) across all binary datasets, three victim architectures (GCN, GAT, GraphSAGE), and both explainers (top: GNNExplainer, bottom: PGExplainer).

Setup. For each dataset-architecture-explainer combination, we evaluate all methods at 14 training budgets from 5% to 70% of the shadow set in 5% increments. At each budget level, $n_{\text{train}} = \lfloor |\mathcal{G}_{\text{shadow}}| \times \text{budget} \rfloor$ samples are drawn per method. Non-boundary does not use explanations; EGSteal-BB, Boundary, and Hybrid are evaluated under both GNNExplainer and PGExplainer. Fidelity trajectories apply a light smoothing (window size 3) for visualization clarity. All plots are x-axis-aligned: for each dataset-architecture-explainer triplet, the x-axis range is clipped to $[\max(\min n_{\text{train}}), \min(\max n_{\text{train}})]$ across methods to ensure fair visual comparison.

Several patterns emerge. First, Boundary (green) and Hybrid (blue) consistently dominate across training budgets, with the gap most pronounced at small n_{train} as boundary pairs provide high-information samples that are especially valuable in low-data regimes. Second, on Tox21 (bottom row), the separation is dramatic: Boundary and Hybrid rise steeply while EGSteal-BB (red dashed) and Non-boundary (gray dotted) remain flat near chance, reflecting Tox21's low 4.2% boundary rate where Phase 2's generative capacity is essential. Third, EGSteal-BB exhibits high variance and sometimes degrades with more data (e.g., AIDS/GAT, PTC_FM/GCN), consistent with label noise from imprecise explanation masks compounding as augmented samples accumulate. Fourth, the patterns are consistent across GNNExplainer (top) and PGExplainer (bottom), confirming our method's robustness to explainer choice.

H Query Complexity Analysis

Given $G^{(t)}$, we query the explainer to obtain

$$G_{\text{sub}}^{(t)} \;=\; G_{\text{sub}}\big(G^{(t)}\big) \;=\; \big(V_{\text{sub}}^{(t)}, E_{\text{sub}}^{(t)}, X_{\text{sub}}^{(t)}\big).$$

We define the one-hop neighborhood of the explanatory nodes as

$$N\big(V_{\text{sub}}^{(t)}\big) \;=\; \big\{j \in V(G^{(t)}) : \exists\, i \in V_{\text{sub}}^{(t)} \text{ with } (i,j) \in E(G^{(t)})\big\}.$$

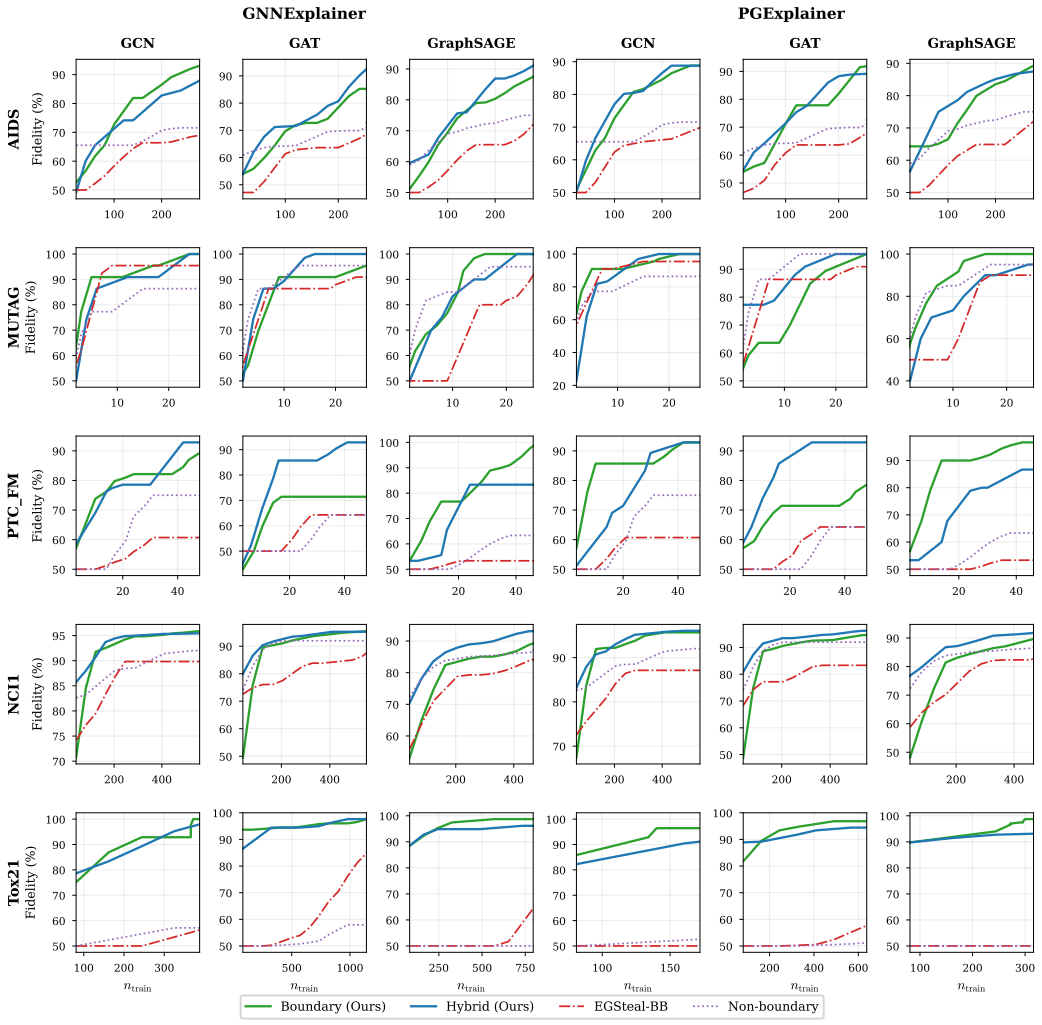


Figure 8: Fidelity (%) vs. training set size across binary datasets, victim architectures, and explainers. Rows correspond to datasets; columns to victim architectures. Top half: GNNExplainer; bottom half: PGExplainer.

Since typical explanations satisfy $|V_{\text{sub}}^{(t)}| \ll n$, the candidate set size is

$$|\mathcal{C}(G^{(t)})| \;=\; O\big(|V_{\text{sub}}^{(t)}|^2 + |V_{\text{sub}}^{(t)}| \cdot d_{\text{avg}} + |\mathcal{S}_{\text{carry}}^{(t)}| + |\mathcal{S}_{\text{rand}}^{(t)}|\big),$$

where d_{avg} is the average degree. This reduces victim queries per iteration from $O(|V|^2 \cdot n_e)$ to $O(|\mathcal{C}(G^{(t)})| \cdot n_e)$, which largely increases our method's efficiency.

To illustrate concretely, consider a typical molecular graph with $n = 18$ nodes and an explanation subgraph of $|V_{\text{sub}}| = 5$ nodes with average degree $d_{\text{avg}} = 3$. The candidate set contains $\binom{5}{2} + 5 \cdot 3 + 5 + 10 = 40$ edges versus $\binom{18}{2} = 153$ possible edges, a 3.8 reduction. With $n_e = 5$ MC samples, Phase 2 requires $\sim$400 queries per graph per iteration versus $\sim$1530 naively and in practice, most graphs resolve within $T \leq 3$ iterations.

We analyze the number of black-box queries required to transform one initial graph G into a boundary pair (G, G'). Let $n = |V(G)|$, let k denote the number of edges flipped per update step, let n_e denote the number of Monte Carlo samples used to estimate each candidate edge sensitivity, and let Δ_e be the maximum allowed edit distance between the initial graph and the returned boundary graph. At iteration t, let

$$C_t = C(G^{(t)})$$

be the explanation-guided candidate edge set, and define

$$C_{\max} = \max_{0 \le t < T_G} |C_t|,$$

where T_G is the number of update iterations needed before the algorithm finds a boundary graph.

Since each iteration flips at most k edges, the number of update iterations is bounded by

$$T_G \le \min\left\{T_{\max}, \left\lceil \frac{\Delta_e}{k} \right\rceil\right\}.$$

The second term is conditional: $\lceil \Delta_e/k \rceil = \lceil 20/3 \rceil = 7$ holds only if every iteration increases the edit distance by exactly k, which requires that no update revisits an already-flipped edge. Because the carry-over set $\mathcal{S}_{\text{carry}}$ can reintroduce such an edge and edge flips are involutions, we rely on the unconditional bound $T_G \le T_{\max} = 10$. More generally, if the first boundary crossing found by the search occurs at edit distance $d_B(G) \le \Delta_e$ from the initial graph, then

$$T_G \le \left\lceil \frac{d_B(G)}{k} \right\rceil.$$

Thus, the step size k directly controls the number of search iterations: larger k reduces the number of rounds, while smaller k yields finer-grained boundary localization.

At each Phase-2 iteration, the algorithm issues a single API call on $G^{(t)}$ that returns both the prediction $y(G^{(t)})$ and the explanation mask $G^{(t)}_{\text{sub}}$ together consistent with our threat model, followed by Monte Carlo sensitivity estimation for all edges in C_t. For each candidate edge $e \in C_t$, each Monte Carlo sample requires two victim prediction queries, one for $\widetilde{G}^{(t)}_m$ and one for $\widetilde{G}^{(t)}_m \oplus e$. Therefore, the number of victim prediction queries in one Phase-2 iteration is

$$1 + 2n_e|C_t|,$$

where the leading term is the single label-and-mask call, so the explanation incurs no separately-charged query.

Therefore, for a single graph entering Phase 2, the total number of victim prediction queries is bounded by

$$\begin{aligned} Q^{(2)}_{\text{pred}}(G) &\le 1 + \sum_{t=0}^{T_G-1} (1 + 2n_e|C_t|) \\ &\le 1 + T_G\,(1 + 2n_e C_{\max}) \\ &= O(T_G n_e C_{\max})\,. \end{aligned}$$

Substituting the step-size bound gives

$$Q^{(2)}_{\text{pred}}(G) = O\left(n_e C_{\max} \min\left\{T_{\max}, \left\lceil \frac{\Delta_e}{k} \right\rceil\right\}\right).$$

If the successful boundary distance is $d_B(G)$, this can be refined to

$$Q^{(2)}_{\text{pred}}(G) = O\left(n_e C_{\max} \left\lceil \frac{d_B(G)}{k} \right\rceil\right).$$

The explanation masks arrive with these calls rather than through separate queries: exactly T_G of the prediction queries above also return a mask, so the number of explanation responses is

$$Q^{(2)}_{\text{exp}}(G) = T_G \le \min\left\{T_{\max}, \left\lceil \frac{\Delta_e}{k} \right\rceil\right\},$$

and adds no cost beyond the queries already counted.

Using the explanation-guided candidate construction in Eq. (9), we have

$$C_{\max} = O\big(s^2 + s d_{\text{avg}} + c_{\text{carry}} + c_{\text{rand}}\big)\,,$$

where

$$s = \max_t |V^{(t)}_{\text{sub}}|, \qquad c_{\text{carry}} = |S_{\text{carry}}|, \qquad c_{\text{rand}} = |S_{\text{rand}}|.$$

Hence the Phase-2 prediction-query complexity becomes

$$Q^{(2)}_{\mathrm{pred}}(G) = O\Bigg(n_e\left(s^2 + sd_{\mathrm{avg}} + c_{\mathrm{carry}} + c_{\mathrm{rand}}\right)\min\left\{T_{\max}, \left\lceil\frac{\Delta_e}{k}\right\rceil\right\}\Bigg).$$

Without explanation-guided reduction, the candidate set could contain all possible edges, giving $C_{\max} = O(n^2)$ and therefore

$$Q^{(2)}_{\mathrm{pred,naive}}(G) = O\Bigg(n_e n^2 \min\left\{T_{\max}, \left\lceil\frac{\Delta_e}{k}\right\rceil\right\}\Bigg).$$

Thus, explanation guidance reduces the per-graph Phase-2 query complexity from quadratic in the graph size to quadratic only in the explanation size.

Including Phase 1, which tests all single-edge removals, the worst-case number of prediction queries for one initial graph is

$$Q_{\mathrm{pred}}(G) \le 1 + |E(G)| + \mathbb{I}_{\mathrm{P2}}(G)\cdot O\Bigg(n_e C_{\max}\min\left\{T_{\max}, \left\lceil\frac{\Delta_e}{k}\right\rceil\right\}\Bigg),$$

where $\mathbb{I}_{\mathrm{P2}}(G) = 1$ if Phase 1 fails to find a boundary pair and the graph is passed to Phase 2, and $\mathbb{I}_{\mathrm{P2}}(G) = 0$ otherwise. Therefore, in the worst case,

$$Q_{\mathrm{pred}}(G) = O\Bigg(|E(G)| + n_e\left(s^2 + sd_{\mathrm{avg}} + c_{\mathrm{carry}} + c_{\mathrm{rand}}\right)\min\left\{T_{\max}, \left\lceil\frac{\Delta_e}{k}\right\rceil\right\}\Bigg).$$

This expression makes explicit the role of the step size k. Increasing k decreases the number of search iterations roughly as $1/k$, thereby reducing the query cost. However, larger k also makes the search less local: the returned boundary pair may lie farther from the initial graph, and the search may overshoot narrow decision-boundary regions. Conversely, smaller k requires more queries but gives finer control over the final boundary-pair distance. In practice, k therefore controls a trade-off between query efficiency and boundary-pair locality.

I Selection Consistency of Top-k Edge Selection

The boundary search (Eq. equation 7) uses the estimated sensitivities $\{\hat{g}_e^{(t)}\}$ only to rank candidates and keep the top k, so its behavior is invariant to any estimation error that preserves the order of the leading candidates. The corollary below makes this precise: the estimator need only order edges whose true sensitivities are separated by a margin, and the required sample size depends on that margin rather than on the absolute accuracy bounded in Theorem 3.1.

Corollary I.1 (Selection consistency)**.** *Fix iteration t and two candidates $e, e' \in \mathcal{C}(G^{(t)})$ with $g_e^{(t)} - g_{e'}^{(t)} \ge \Delta > 0$. With $Z \in [-1, 1]$ and n_e Monte Carlo rounds per edge,*

$$\Pr\left(\hat{g}_{e'}^{(t)} \ge \hat{g}_e^{(t)}\right) \;\le\; \exp\left(-\frac{n_e\Delta^2}{8}\right).$$

Consequently, for confidence $1 - s$, if

$$n_e \;\ge\; \frac{8}{\Delta^2}\,\log\frac{\binom{|\mathcal{C}(G^{(t)})|}{2}}{s},$$

then with probability at least $1 - s$ every pair of candidates whose true sensitivities differ by at least Δ is ordered correctly, and $\mathrm{TopK}_{\max}$ excludes no edge whose true sensitivity exceeds that of a selected edge by more than Δ.

Proof. Let $D = \hat{g}_e^{(t)} - \hat{g}_{e'}^{(t)} = \frac{1}{n_e}\sum_{m=1}^{n_e} U_m$ with $U_m = Z_{e,m}^{(t)} - Z_{e',m}^{(t)} \in [-2, 2]$, independent across m and with mean $\mathbb{E}[U_m] = g_e^{(t)} - g_{e'}^{(t)} \ge \Delta$. A misordering is the event $D \le 0$, which implies $D - \mathbb{E}[D] \le -\Delta$. Applying Hoeffding's inequality to $\sum_m U_m$ (each of range 4),

$$\Pr(D \le \mathbb{E}[D] - \Delta) \;\le\; \exp\left(-\frac{2(n_e\Delta)^2}{n_e\cdot 4^2}\right) \;=\; \exp\left(-\frac{n_e\Delta^2}{8}\right).$$

Taking a union bound over the at most $\binom{|\mathcal{C}(G^{(t)})|}{2}$ candidate pairs and setting the right-hand side to $s/\binom{|\mathcal{C}(G^{(t)})|}{2}$ yields the sample size. The final claim holds because violating the Top-k selection requires some excluded edge to outrank a selected one, i.e., a reversed Δ-separated pair. □

Two consequences follow. A coarse estimator (small n_e) already selects correctly whenever the leading edges are well separated in true sensitivity, so accuracy in $\hat{g}_e^{(t)}$ is not needed where it does not change the ranking. And even a misordering never corrupts the output: every recorded pair is checked against the victim by Eq. equation 8, so a wrong ranking costs at most search efficiency, not correctness.