A PRE-INDEXING ADVERSARIAL ATTACK ON NEURAL SUBGRAPH RETRIEVERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Given a query graph, a neural subgraph retriever (NSR) returns the relevant graphs from a corpus of graphs, where the underlying learned relevance approximates combinatorial subgraph matches. In practice, corpus maintenance is often delegated to a third party vendor, who has read/write access to the corpus and possibly historic query workload. Such a vendor can turn adversarial and corrupt the corpus. Motivated by this threat, we propose QUPAC, a query workload-guided universal corpus attack, where the adversary perturbs a small subset of corpus graphs once, before the NSR indexes and uses the corrupted corpus for future queries. We formulate the adversary's strategy as worst-case maximization of a ranking loss, jointly over the attacked corpus subset and label-preserving node-pair perturbations on selected victim graphs. We show that this can be cast as a cardinality-constrained set function optimization, which can be approximately solved using a greedy algorithm. However, this requires an exhaustive search over all possible perturbations of the entire corpus. To make the attack scalable, the adversary prunes the ground set using a novel application of approximate nearest neighbor (ANN) search. It first computes a query-specific attack score approximating marginal gain, converts it into an ANN-compatible form and then make the index robust to possible future queries. Experiments across multiple datasets show that QUPAC reduces retrieval quality and outperforms existing attack baselines.

1 INTRODUCTION

Given a fixed corpus of graphs $\{G_c\}$ and an arbitrary query graph G_q, a subgraph retrieval system seeks relevant (positive) corpus graphs that contain G_q as a subgraph. Other graphs are non-relevant (negative). It has applications in chemical motif discovery (Willett et al., 1998; Rose et al., 2024), scene graph retrieval (Shankar et al., 2016), hardware Trojan detection (Piccolboni et al., 2017), *etc*. A recent line of work develops neural subgraph retriever (NSR) models (Roy et al., 2022; Ranjan et al., 2022; Ramachandran et al., 2024; Li et al., 2019), which approximate the combinatorial subgraph matching task with a differentiable surrogate. For each query, an NSR ranks corpus graphs by a learned relevance distance and returns the top-K graphs with the smallest distances.

As in many information retrieval systems (Zhong et al., 2023; Su et al., 2025; Li et al., 2025; Zou et al., 2025), an outsider **adversary** may attack the **victim** NSR by perturbing corpus graphs before indexing and search. For example, a third-party vendor with read/write access to raw corpus graphs may also observe a *query workload*, *i.e.*, samples of past queries from query logs. If compromised, the vendor can use this workload to perturb corpus graphs *once* before deployment, so that the modified corpus is reused for all subsequent queries. At query time, these perturbations alter the relevance distances computed by the victim NSR and degrade its retrieval accuracy.

Our goal We aim to design a *pre-indexing, universal* adversarial attack under the above threat model. It works in two steps: **(I)** Before deployment, the adversary uses the query workload to perturb a select subset of corpus graphs. Thereafter, the corpus remains fixed. **(II)** At query time, the victim NSR computes relevance distances between the query and all graphs in the *modified* corpus and returns the top-K candidates. A successful attack reduces the number of relevant graphs in the top-K set, degrading the victim NSR's ranking quality.

1.1 PRESENT WORK

In this work, we achieve our goal using QUPAC, a novel method for **Qu**ery-workload-guided **P**re-indexing **A**ttack on **C**orpus. Adversarial attacks must be imperceptible; otherwise, they are easily detectable. Therefore, our adversary attacks only a small subset of corpus graphs by adding or delet-

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.

ing a single edge in each selected graph and leaves the rest unperturbed. This presents two challenges: **(1)** Identifying the corpus subset and the underlying perturbations involves a prohibitively large search space of corpus graphs and candidate node pairs. **(2)** Since the attack is fixed before query time, it cannot adapt to individual queries. Thus, the key challenge is to ensure that the attack strategy, *i.e.*, the choice of the corpus subset and the perturbations, remains effective beyond the query workload and generalizes to unseen queries. In this work, we address the above challenges with the following technical contributions.

Novel formulation of pre-indexing adversarial attack We formulate the attack as a worst-case maximization of pairwise ranking loss, jointly over the corpus subset and node-pair perturbations, subject to a budget on the subset size. The loss is aggregated over the observed query workload, so that the optimization leads to the selection of a corpus subset and one perturbation per selected graph. These choices remain fixed for all new queries, yielding a pre-indexing, universal attack.

Characterization of adversarial objective The adversarial optimization involves two structured choices: the corpus subset and perturbations of the graphs in the selected subset. For a fixed corpus subset, the optimal graph perturbations depend on the underlying subset. This reduces our attack to a cardinality-constrained set function maximization problem, where the objective is the maximum adversarial ranking loss achieved by optimally perturbing the graphs in the candidate subset. We show that the set function is monotone and approximately submodular (El Halabi et al., 2020; Gatmiry et al., 2018). Consequently, a greedy maximization algorithm that iteratively adds the corpus graph with the highest marginal gain admits an approximation guarantee (Bian et al., 2017).

Robust ANN-based ground-set reduction A vanilla greedy algorithm is computationally expensive, as each iteration scans the full corpus and evaluates candidate perturbations to compute marginal gains. The adversary addresses this bottleneck by pruning the ground set *once, before running the greedy algorithm*, using the following approximate nearest-neighbor (ANN) procedure (Indyk et al., 1998; Andoni et al., 2018; Gionis et al., 1999; Liu et al., 2004).
— *Query specific attack scores:* The adversary first maximizes marginal gain separately for each query in the workload. For a query-corpus pair and a candidate perturbation, it defines an *attack score* as the perturbation-induced change in their relevance distance. This score approximates the query-specific marginal gain.
— *Candidate selection as retrieval:* The adversary views selecting a corpus graph with the highest attack score as an auxiliary top-1 retrieval problem. These query-specific searches identify candidates for the reduced ground set.
— *ANN-compatible distance representation:* The attack score involves a difference of hinge distances, which does not directly support ANN indexing. The adversary uses random projections to approximate relevance distance as a dot product between query and corpus-dependent vectors, enabling maximum inner-product search for high-scoring candidates (Nair et al., 2026).
— *Robust ANN index design:* The observed workload may not accurately represent future queries. Rather than generating new queries from an unknown distribution, the adversary perturbs NSR model parameters as to simulate the effect of variation due to unseen queries. It trains a representation network through min–max adversarial optimization and indexes corpus graphs using the resulting representations. This makes the attack robust to new queries beyond the workload.
— *Fusion into a reduced ground set:* The adversary probes the index with each workload query and merges the resulting candidate sets through uniformly weighted voting. This yields a reduced ground set, smaller than the full corpus.

Our experiments show that QUPAC degrades ranking accuracy by 17% for several state-of-the-art NSR models (depicted in Figure 1). Moreover, it provides more effective attacks than several baselines.

2 PRELIMINARIES

Notation We write a corpus graph as $G_c = (V_c, E_c)$ and a query graph as $G_q = (V_q, E_q)$. Given $[N] = \{1, \ldots, N\}$, we use $C = [N]$ to index the set of corpus graphs $\mathcal{G} := \{G_c : c \in C\}$. For any two sets S and T, their symmetric difference is written as $S \oplus T = (S \backslash T) \cup (T \backslash S)$. We define the hinge function as $[a]_+ = \max\{0, a\}$ and the pairwise ranking loss as $\ell(d, d') = [\gamma + d - d']_+$, where d and d' are distances and γ is a tunable margin. Let $\Pi_c = (V_c \times V_c) \cup \perp$ be the set of all possible perturbations for a corpus graph G_c, where $\perp$ means no perturbation.

Set functions and their properties Consider a ground set C and a set function $F : 2^C \to \mathbb{R}$. For $S \subset C$ and $c \notin S$, we define the marginal gain from adding c to S as $\Delta_F(c \mid S) = F(S \cup \{c\}) - F(S)$. We call F monotone if $\Delta_F(c \mid S) \geq 0$ for every $S \subseteq C$ and $c \notin S$. We call F α-submodular if there exists an $\alpha \in (0, 1]$ such that $\Delta_F(c \mid S) \geq \alpha \Delta_F(c \mid T)$ for every $S \subseteq T \subseteq C$ and $c \notin T$. When $\alpha = 1$, the above inequality condition reduces to exact submodularity (Nemhauser et al., 1978; Krause et al., 2008; Mirzasoleiman et al., 2015). When $\alpha \in (0, 1)$, the function F is called $\alpha-$approximately submodular (El Halabi et al., 2020; Gatmiry et al., 2018; Bian et al., 2017).

Subgraph matching based graph retrieval We write $y(G_q, G_c) \in \{\pm 1\}$ to denote the binary relevance label between G_q and G_c. In our context, $y(G_q, G_c) = +1$ if G_q is a subgraph of G_c and -1, otherwise. The goal of a graph retriever is to return the corpus graphs that are relevant to a given query G_q. Since subgraph matching is NP-hard, selecting relevant graphs for even a single query is computationally prohibitive, as the retriever must solve NP-hard problems for many graph pairs.

To address this bottleneck, neural subgraph retrievers (NSR) relax the discrete subgraph-matching condition using continuous graph embeddings. For example, Li et al. (2019) use a graph neural network (GNN) to obtain graph embeddings $\boldsymbol{h}^{(q)}, \boldsymbol{h}^{(c)} \in \mathbb{R}^{\dim_h}$ and approximate the condition $G_q \subseteq G_c$ using $\boldsymbol{h}^{(q)} \leq \boldsymbol{h}^{(c)}$ (elementwise). Violations of this condition induce an asymmetric relevance distance $d_\theta(G_q, G_c) = \|[\boldsymbol{h}^{(q)} - \boldsymbol{h}^{(c)}]_+\|_1$, where θ consists of the trainable parameters. The exact choice of relevance distance $d_\theta(G_q, G_c)$ differ across NSR models (Roy et al., 2022; Ramachandran et al., 2024) and is mentioned in Appendix F.

Imperceptibility in graph perturbation In image-based attacks, imperceptibility is enforced by bounding the perturbation in various ways to keep the modified image indistinguishable from the original (Guo et al., 2025). In the same spirit, our adversary seeks imperceptibility by limiting perturbation to exactly one node pair in each attacked graph. Given a corpus graph $G_c = (V_c, E_c)$, let $\Pi_c = (V_c \times V_c) \cup \perp$ be the set of perturbations, where $\perp$ indicates no perturbation.

The adversary flips the state of one node pair $\delta_c = (u_c, v_c) \in \Pi_c$. Specifically, it removes δ_c if $\delta_c \in E_c$ and adds it otherwise, yielding the perturbed graph $G_c \oplus \delta_c = (V_c, E_c \oplus \delta_c)$. As we shall see, the adversary selects δ_c before deployment and keeps it fixed across queries. For query-specific perturbations, we will use δ_{qc} to make the dependence explicit.

Algorithm 1: A pre-indexing NSR attack

Require: $\mathcal{G} = \{G_c : c \in C\}$, Query workload Q, Budget B, substitute d_θ, victim NSR, real queries $\{G_q : q \in Q^*\}$
`# Stage (1): Pre-indexing (adversary)`
$S, \{\delta_c\} \leftarrow$ **Attack**$(\mathcal{G}, \{G_q : q \in Q\}, d_\theta, B)$
$\mathcal{G}_{S,\{\delta_c\}} \leftarrow \{G_c \oplus \delta_c : c \in S\} \cup \{G_c : c \in C \setminus S\}$
`# Stage (2): Pre-processing (victim)`
Embed and index $\mathcal{G}_{S,\{\delta_c\}}$
`# Stage (3), Retrieval (victim)`
for each G_q, $q \in Q^*$ **do**
    Compute relevance distance between G_q, G'_c using victim NSR for all $G'_c \in \mathcal{G}_{S,\{\delta_c\}}$ and return a ranked list.

NSR search system It works in three stages:

(1) Pre-Indexing: A data-preprocessing stage where the corpus set is curated and stored.
(2) Preprocessing for search: Corpus graphs $\{G_c : c \in C\}$ are embedded and indexed.
(3) Retrieval: The NSR model embeds each incoming query G_q, compute the relevance distance $d_\theta(G_q, G_c)$ against each corpus graph G_c, and return the relevant graphs with the lowest distances.

3 Proposed Approach

The threat model The owner of the victim NSR system may delegate pre-indexing corpus preprocessing to a third-party vendor and grant it read/write access to the raw corpus graphs, while retaining control over the subsequent indexing and retrieval stages. We model the adversary as a compromised vendor with full access to stage **(1)**, and it misuses this access to manipulate the corpus before indexing.

— *Victim and substitute:* In general, the adversary does not have access to the owner's victim NSR model. Instead, it relies on a substitute model, with relevance distance d_θ, as a proxy for the victim. In white box attacks, the substitute model d_θ matches the victim exactly, whereas in black box attacks, they generally differ.

— *Access privilege:* The adversary has access to a finite query workload $\{G_q : q \in Q\}$ of historical queries, read and write access to every corpus graph in $\{G_c\}$, and a subgraph matching solver, which can be used to compute the relevance labels for any query corpus graph pairs. The adversary does not have access to the future queries, denoted as $\{G_q : q \in Q^*\}$.

— *Pre-Indexing attack:* Given the corpus $\{G_c : c \in C\}$, the adversary seeks a subset $S \subseteq C$ with

a budget $|S| \leq B$ and a node-pair perturbation $\delta_c = (u_c, v_c) \in \Pi_c$ to obtain $G_c \oplus \delta_c$ for each $c \in S$. The remaining corpus graphs with $c \notin S$ remain unperturbed. Based on this selection, the modified corpus $\mathcal{G}_{S,\{\delta_c:c\in S\}}$ is prepared with corpus graphs

$$\overline{G}_c = \begin{cases} G_c \oplus \delta_c & \text{if } c \in S, \\ G_c & \text{if } c \notin S. \end{cases} \quad \mathcal{G}_{S,\{\delta_c:c\in S\}} = \{\overline{G}_c : c \in C\} = \{G_c \oplus \delta_c : c \in S\} \cup \{G_c : c \notin S\} \quad (1)$$

The adversary's goal is to select the subset S and the perturbations $\{\delta_c : c \in C\}$ such that the modified corpus minimizes the victim NSR's ranking accuracy on unseen queries in Q^*.

Indexing and querying the modified corpus. Because perturbations are applied to raw graphs before embedding, they persist through stages **(2)** and **(3)** by the owner. Following the standard NSR pipeline, the victim embeds and indexes $\mathcal{G}_{S,\{\delta_c:c\in S\}}$ once, which is used to respond to future queries. Algorithm 1 summarizes the overall pipeline. We now develop QUPAC, which implements the Attack step.

3.1 ADVERSARIAL OPTIMIZATION TASK

Non-critical perturbations In images, adversarial perturbations are so subtle to human eyes that the ground-truth label of the perturbed image remains unchanged. However, in graphs, perturbing even one node pair can change whether $G_q \subseteq G_c$, thus flipping the relevance label, *i.e.*, $y(G_q, G_c \oplus \delta_c) \neq y(G_q, G_c)$. Such label flips not only make the attack detectable, but also inadvertently benefit the victim. For example, an adversarial attack may increase the relevance distance of a positive graph to push it lower in the ranking. However, if the perturbation also flips its label to negative, then ranking it lower becomes correct for the victim, thereby improving retrieval accuracy. To this end, we define the sets of *non-critical* node pairs as those whose perturbation preserves the relevance label:

$$\Gamma_{qc} = \{\delta_c \in \Pi_c : y(G_q, G_c) = y(G_q, G_c \oplus \delta_c)\} \text{ and } \quad \Gamma_c = \cap_{q\in Q}\Gamma_{qc} \quad (2)$$

The set Γ_{qc} is query-specific, whereas the adversary needs a single perturbation per corpus graph that is fixed across queries. To this end, the adversary builds Γ_c, a query-agnostic set of non-critical node pairs for each corpus graph G_c. It restricts δ_c to Γ_c, ensuring that the chosen perturbation preserves the relevance label for every query in the observed workload. Note that Γ_c is non-empty as it contains $\perp \in \Pi_c$. If $\Gamma_c = \{\perp\}$, then no perturbation is carried out on G_c.

Objective Given the corpus $\mathcal{G} = \{G_c : c \in C\}$ and budget B, the adversary seeks to find $S \subseteq C$ with $|S| \leq B$, and *non-critical* perturbations $\{\delta_c : c \in S\}$ jointly to maximize the pairwise ranking loss over the modified corpus, aggregated across all workload queries Q. Using the modified graphs and corpus defined in Eq. (2) and the positive-negative graph pairs $(\overline{G}_{c_\oplus}, \overline{G}_{c_\ominus})$, we define the workload-aggregated ranking loss as

$$\mathcal{L}(S, \{\delta_c : c \in S\}, Q) = \textstyle\sum_{q\in Q} \sum_{\overline{G}_{c_\oplus}, \overline{G}_{c_\ominus} \in \mathcal{G}_{S,\{\delta_c:c\in S\}}} \ell(\mathrm{d}_\theta(G_q, \overline{G}_{c_\oplus}), \mathrm{d}_\theta(G_q, \overline{G}_{c_\ominus})). \quad (3)$$

Then the adversary aims to solve the optimization problem:

$$\underset{S,\{\delta_c:c\in S\}}{\text{maximize}} \;\; \mathcal{L}(S, \{\delta_c : c \in S\}, Q) \;\text{ such that }\; \delta_c \in \Gamma_c, \text{ and } |S| \leq B. \quad (4)$$

In the above, restricting each δ_c to Γ_c ensures that the attack remains discrete and preserves relevance labels over the observed workload.

3.2 CHARACTERIZATION OF ADVERSARIAL OBJECTIVE

Coupling between S and δ_c In the optimization (4), the subset S and perturbations $\{\delta_c\}_{c\in S}$ are inherently coupled. This is because a corpus graph can be relevant to one query and irrelevant to another, so the same perturbation can affect different ranking losses in opposite directions. Consequently, we cannot choose the optimal δ_c independently for each graph. It depends on which other graphs are included in S, making the optimal δ_c a function of S.

Reducing the problem (4) into cardinality constrained set function optimization Given a subset S, we write the optimal choice of δ_c as $\{\delta_c(S) : c \in S\}$. This can be obtained after fixing S, by solving (4) over the perturbations alone, *i.e.*, $\{\delta_c(S) : c \in S\} = \mathrm{argmax}_{\{\delta_c\in\Gamma_c:c\in S\}} \mathcal{L}(S, \{\delta_c : c \in S\}, Q)$. The corresponding optimal value of $\mathcal{L}(S, \{\delta_c(S) : c \in S\}, Q)$ depends only on S, yielding a set function $F(S, Q)$. This subsequently turns the adversary's joint optimization (4) into a cardinality constrained set function maximization problem as follows:

$$\underset{S}{\text{maximize}} \;\; F(S, Q) \;\text{ subject to } |S| \leq B, \qquad \text{where, } \; F(S, Q) = \mathcal{L}(S, \{\delta_c(S) : c \in S\}, Q) \quad (5)$$

Monotonicity and approximate submodularity of $F(S, Q)$ Below, we present formal results showing that $F(S, Q)$ is monotone and approximately submodular (Proven in Appendix E).

Theorem 1 (Monotonicity and approximate submodularity of F)**.** Let the set function $F(S, Q)$ be defined as in Eq. (5). Let Q denote the query workload, C the index for the set of corpus graphs, and γ be the tunable margin. Let Γ_c be as defined in Eq (2). Assume that $A_q(c, \delta_c) \geq 0$ for all $\delta_c \in \Gamma_c$. Then, we have the following properties. **(1)** The function $F(S, Q)$ is a monotone set function. **(2)** If there exists $\beta_0 \in (0, 1]$, such that, at least β_0 fraction of the positive-negative graph pairs $(\overline{G}_{c_\oplus}, \overline{G}_{c_\ominus})$ containing G_c satisfy $\mathrm{d}_\theta(G_q, \overline{G}_{c_\oplus}) - \mathrm{d}_\theta(G_q, \overline{G}_{c_\ominus}) + \gamma \geq 0$ then $F(\bullet, Q)$ is α-submodular with $\alpha \geq \beta_0$.

Greedy routine The properties derived in Theorem 1 allow the adversary to optimize F using a greedy algorithm (El Halabi et al., 2020; Bian et al., 2017) (Algorithm 2). Starting from $S = \emptyset$, at each iteration t, the algorithm adds corpus item c with the largest marginal gain $\Delta_F(c \,|\, S, Q) = F(S \cup c, Q) - F(S, Q)$. Owing to monotonicity and α-submodularity (Theorem 1), this greedy procedure achieves a $(1 - e^{-\alpha})$-approximation to the optimal value of F (Bian et al., 2017).

Algorithm 2: Greedy routine for attack

1: **Require:** $\mathcal{G} = \{G_c : c \in C\}$, Query workload Q, Budget B, substitute d_θ, non-critical set Γ_c.
2: $S^{(0)} \leftarrow \emptyset$
3: **for** $t = 1, \ldots, B$ **do**
4: $\quad c^* \leftarrow \mathrm{argmax}_{c \in C \setminus S} \Delta_F(c \,|\, S, Q)$
5: $\quad S \leftarrow S \cup \{c^*\}$
6: $\{\delta_c(S)\} \leftarrow \mathrm{argmax}_{\delta_c \in \Gamma_c} \mathcal{L}(S, \{\delta_c : c \in S\}, Q)$
7: **Return** S, $\{\delta_c(S) : c \in S\}$

Challenges Each greedy selection in line 4 requires computing the marginal gain $\Delta_F(c \,|\, S, Q)$ for $O(N)$ corpus graphs. Moreover, if n is the maximum number of nodes in the corpus graphs, then evaluating this marginal gain for a single graph G_c requires $|\Gamma_c| = O(n^2)$ calls to the substitute model per query. Therefore, the greedy routine has a time complexity of $O\big(n^2\, B\, N\, |Q|\big)$.

3.3 Pruning the ground set

High level overview An effective attack requires at least one scan over the corpus and its candidate perturbations, but repeating this scan across all B greedy iterations is prohibitively expensive. The adversary therefore performs a one-time preprocessing step to prune the full ground set C to a much smaller candidate set $\widehat{C}$ with $|\widehat{C}| \ll N$. The greedy routine then operates only over $\widehat{C}$.

The adversary first focuses on a *query-specific attack problem*. For each workload query G_q with query $q \in Q$, it seeks to select an adversarial subset $S_q \subseteq C$ and corresponding node-pair perturbations $\delta_{qc} \in \Gamma_{qc}$, both tailored to G_q. The adversary uses an ANN-based selection procedure to retrieve promising (S_q, δ_{qc}) candidates and then apply a suitable *aggregation* operation on them to obtain the pruned ground set R. We now detail these two steps.

Query specific adversarial optimization First, the adversary seeks a subset S_q and perturbations $\{\delta_{qc} : c \in S_q\}$ that degrade the ranking quality for each query G_q separately. This is equivalent to maximizing $\mathcal{L}(S, \{\delta_c\}, q)$ in Eqn. (3) after restricting to one query.

Attack score Even for a single query, maximizing $\mathcal{L}(S, \{\delta_c\}, q)$ requires repeated marginal-gain evaluations over the entire corpus, which is computationally expensive. The adversary approximates the marginal gain using a *signed distance shift*. For (G_q, G_c) and any non-critical perturbation δ_c. The signed distance shift is defined as:

$$A_q(c, \delta_c) = y(G_q, G_c) \cdot (\mathrm{d}_\theta(G_q, G_c \oplus \delta_c) - \mathrm{d}_\theta(G_q, G_c)) \quad \text{where } \delta_c \in \Gamma_{qc}. \tag{6}$$

The intuition behind the design of $A_q(c, \delta_c)$ goes as follows. To degrade the victim NSR's retrieval accuracy, the adversary should select δ_c to push relevant graphs lower in the ranking and pull non-relevant graphs higher. In terms of the relevance distance, it therefore wants the raw shift $\mathrm{d}_\theta(G_q, G_c \oplus \delta_c) - \mathrm{d}_\theta(G_q, G_c)$ to be large positive for a relevant graph ($y(G_q, G_c) = +1$), and large negative for a non-relevant one ($y(G_q, G_c) = -1$). Apart from being an intuitive attack score, $A_q(c, \delta_c)$ also provides a multiplicative approximation of the marginal gain (proven in Appendix E).

Theorem 2. Given a corpus $\mathcal{G} = \{G_c : c \in C\}$ and a query G_q with $q \in Q$, let $\mathcal{L}$ be the set function defined in Eq. (3). Then, under mild assumptions, there $\exists \beta_1, \beta_2 > 0$, such that for every perturbation $\delta_c \in \Gamma_{qc}$ defined in Eq. (2), we have: $\beta_1 A_q(c, \delta_c) \leq \Delta_{\mathcal{L}}((c, \delta_c) : S, q) \leq \beta_2 A_q(c, \delta_c)$.

Viewing $\Delta_F(c \,|\, S, q)$ maximization as a top-1 retrieval task Because $A_q(c, \delta_c)$ approximates the marginal gain of $\mathcal{L}$, the adversary replaces marginal-gain maximization in line 4 of Algorithm 2 with maximization of the attack score $A_q(c, \delta_c)$. The adversary views this as an auxiliary retrieval

problem, where the corpus consists of the set of perturbed graphs indexed by $\mathcal{C} = C \times \Pi_c$ and the relevance between the query G_q and corpus item (G_c, δ_c) is measured using attack score $A_q(c, \delta_c)$. For each query G_q, the adversary retrieves the top-$\mathcal{B}$ graph–perturbation pairs and uses their graph indices to form S_q. Unlike the exact marginal gain Δ_F, the attack score $A_q(c, \delta_c)$ does not depend on perturbations assigned to other graphs or on the current selected set S_q. This allows the adversary to preprocess all candidate perturbations once and replace the iterative greedy search with a single top-$\mathcal{B}$ retrieval for each workload query.

Selecting top perturbed graphs with high attack score $A_q(c, \delta_c)$ still requires $O(Nn^2)$ graph perturbation pairs *per query*. The adversary avoids this by using approximate nearest neighbors (ANN) search: it first indexes all perturbed graphs and then retrieves the most promising corpus graphs in time that is sublinear in corpus size N. However, significant enhancements are needed, because standard ANN methods (Indyk et al., 1998; Andoni et al., 2018; Douze et al., 2025; Simhadri et al., 2023) work in Euclidean space—often dot product or cosine similarity—between independently computed query and corpus representations. In its current form, attack score (6) is not ANN-amenable. Therefore, the adversary will approximate $A_q(c, \delta_c)$ as the dot product between suitable query and corpus graph representations. This is achieved in three steps.

Representing $\mathrm{d}_\theta(G_q, G_c)$ in ANN-compatible form The attack score in Eq. (6) is computed as the difference between the relevance distances of perturbed and clean graphs from the query. Therefore, we first obtain an ANN-amenable form for the relevance distance $\mathrm{d}_\theta(G_q, G_c)$. Observing that $\mathrm{d}_\theta(G_q, G_c)$ is asymmetric, and represents the extent of inability to map G_q into a subgraph of G_c, we draw upon recent neural estimators (Ramachandran et al., 2024) of the asymmetric hinge form $\mathrm{d}_\theta(G_q, G_c) = \sum_{j\in[\dim_h]}[\boldsymbol{h}^{(q)}[j] - \boldsymbol{h}^{(c)}[j]]_+$, where $\boldsymbol{h}^{(\cdot)} \in \mathbb{R}^{\dim_h}$ denotes a GNN-contextualized, pooled, single-vector embedding of the respective graphs. The hinge distance is readily rewritten in terms of the ReLU of an inner product in an augmented vector space:

$$\mathrm{d}_\theta(G_q, G_c) = \sum_{j\in[\dim_h]} \left[\boldsymbol{a}_j(G_q, \theta)^\top \boldsymbol{b}_j(G_c, \theta)\right]_+, \text{ with } \begin{cases} \boldsymbol{a}_j(G_q, \theta) = \left[\boldsymbol{h}^{(q)}[j]; -1\right] \\ \boldsymbol{b}_j(G_c, \theta) = \left[1; \boldsymbol{h}^{(c)}[j]\right] \end{cases} \in \mathbb{R}^{2\times 1} \tag{7}$$

Note that $\boldsymbol{b}_j$ depends only on G_c and can therefore be precomputed.

— *Augmented representations:* Although the above form of d_θ separates query and corpus vectors, the hinge nonlinearity still makes it difficult to reuse a standard ANN index. To overcome this, the adversary approximates each hinged dot product with a high-probability estimate expressed as a dot product between randomized vectors. Given a random hyperplane $\boldsymbol{\omega} \sim \mathcal{N}(0, \boldsymbol{I}_2)$, the adversary defines a feature map $\Phi_{\boldsymbol{\omega}}$ which performs random projection on any vector $\boldsymbol{a}$ to obtain:

$$\Phi_{\boldsymbol{\omega}}(\boldsymbol{a}) = \tfrac{1}{\sqrt{2}}\left[\boldsymbol{a};\ \mathrm{sign}(\boldsymbol{\omega}^\top \boldsymbol{a})\, \boldsymbol{a}\right] \tag{8}$$

For any two vectors $\boldsymbol{a}$ and $\boldsymbol{b}$, we have $\Phi_{\boldsymbol{\omega}}(\boldsymbol{a})^\top \Phi_{\boldsymbol{\omega}}(\boldsymbol{b}) = \frac{1}{2}\left(1 + \mathrm{sign}(\boldsymbol{\omega}^\top \boldsymbol{a})\mathrm{sign}(\boldsymbol{\omega}^\top \boldsymbol{b})\right) \boldsymbol{a}^\top \boldsymbol{b}$. which equals to $\boldsymbol{a}^\top \boldsymbol{b}$ or 0, never exceeds $[\boldsymbol{u}^\top \boldsymbol{v}]_+$ and equals $[\boldsymbol{u}^\top \boldsymbol{v}]_+$ with probability at least $1/2$ (Nair et al., 2026; Gionis et al., 1999). The adversary then draws R i.i.d. random hyperplanes $\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_R$, applies $\Phi_{\boldsymbol{\omega}_r}$ to $\boldsymbol{a}_j(G_q, \theta)$ and $\boldsymbol{b}_j(G_c, \theta)$, and takes the maximum dot product over the R projections. This yields a high-probability estimate of $\mathrm{d}_\theta(G_q, G_c)$, as stated below (Proven in Appendix E).

Proposition 3 (Adapted from (Nair et al., 2026))**.** Given R random hyperplanes $\boldsymbol{\omega}_r$ drawn i.i.d. from $\boldsymbol{\omega}_r \sim \mathcal{N}(0, \boldsymbol{I}_2)$ and the relevance distance $\mathrm{d}_\theta(G_q, G_c)$ computed in Eq. (7), let $\Phi_{\boldsymbol{\omega}}$ be the feature map defined in Eq. (8). Then, with probability at least $1 - \dim_h / 2^R$, we have:

$$\mathrm{d}_\theta(G_q, G_c) = \sum_{j\in[\dim_h]} \max_{r\in[R]} \Phi_{\boldsymbol{\omega}_r}(\boldsymbol{a}_j(G_q, \theta))^\top \Phi_{\boldsymbol{\omega}_r}(\boldsymbol{b}_j(G_c, \theta)) \tag{9}$$

Similarly, we have: $\mathrm{d}_\theta(G_q, G_c \oplus \delta_c) = \sum_{j\in[\dim_h]} \max_{r\in[R]} \Phi_{\boldsymbol{\omega}_r}(\boldsymbol{a}_j(G_q, \theta))^\top \Phi_{\boldsymbol{\omega}_r}(\boldsymbol{b}_j(G_c \oplus \delta_c, \theta))$ w.h.p. These embeddings $\Phi_{\boldsymbol{\omega}_r}(\boldsymbol{a}_j(G_q, \theta))$ and $\Phi_{\boldsymbol{\omega}_r}(\boldsymbol{b}_j(G_c, \theta))$ serve as augmented representations of query and corpus graphs.

Incorporating robustness to unseen queries using parameter perturbation Using Eq. (9), the adversary can rewrite the attack score in Eq. (6) in an ANN-compatible form, index the resulting corpus embeddings, and probe the index with each workload query G_q to obtain the subset S_q. However, these subsets are tailored to the historic workload queries Q, which may differ significantly from real queries Q^* when the NSR is deployed. The adversary does not have access to queries from Q^*, so it cannot fine-tune its attack model.

Algorithm 3: QUPAC (Indexing)

1: **Input**: Corpus graphs $\mathcal{G}$, query workload Q, d_θ for substitute model, non-critical sets Γ_c, random hyperplanes $\boldsymbol{W}_j, j \in [\dim_h]$, $\boldsymbol{\omega}_r, r \in [R]$.
2: `# Train` $\boldsymbol{f}_\psi$ `on` $\mathcal{G}$ `and` Q
$\psi \leftarrow \text{Train}(\boldsymbol{f}, \mathcal{G}, Q, \mathrm{d}_\theta, \{\boldsymbol{\omega}_r\})$
3: `# Compute precomputable embeddings for each corpus graphs and perturbations`
Compute $\boldsymbol{\zeta}_{\psi,j}(G_c \oplus \delta_c, \theta)$ for all $(c, \delta_c) \in C \times \Gamma_c$
Compute $\boldsymbol{x}_j(G_c, \delta_c) = \boldsymbol{\zeta}_{\psi,j}(G_c, \theta) - \boldsymbol{\zeta}_{\psi,j}(G_c \oplus \delta_c, \theta)$ for all $c \in C, \delta_c \in \Gamma_c$
4: `# Perform indexing`
$\mathcal{I} \leftarrow \text{Index}(\{\boldsymbol{x}_j(G_c, \delta_c)\}, \{\boldsymbol{W}_j\})$

Algorithm 4: QUPAC (Efficient Attack)

1: **Input**: Index I and all inputs from Algorithm 3, pruned ground set size $\mathcal{B} > B$.
2: `# Compute query specific subsets and deltas`
3: Compute $\boldsymbol{\eta}_{\psi,j}(G_q, \theta)$ for $q \in Q$ with Eq. (11)
4: $\{S_q, \{\delta_c : c \in S_q\}\} \leftarrow \cup_j \text{Retrieve}(\mathcal{I}, \pm\boldsymbol{\eta}_{\psi,j}(G_q, \theta))$.
5: `# Obtain pruned set`
$\text{votes}(c, \delta_c) = \sum_q |(c, \delta_c) \in S_q \times \{\delta_c : c \in S_q\}|$
$\widehat{C}, \{\delta_c : c \in \widehat{C}\} \leftarrow \text{top-}\mathcal{B}\big(\{\text{votes}(c, \delta_c)\}\big)$
6: `# Replace` C `with` $\widehat{C}$ `in line 4 of Algorithm 2`
$S, \{\delta_c\} \leftarrow \textbf{Attack}(\mathcal{G}_{\widehat{C}, \{\delta_c : c \in \widehat{C}\}}, \{G_q\}, d_\theta, B)$
7: Run Algorithm 1, line 4 onwards.

Rather than explicitly generating new queries, the adversary anticipates workload drift by perturbing the substitute model within a neighborhood of θ and uses the resulting embeddings as surrogates for the embeddings of unseen queries. To this end, the adversary uses a neural network $\boldsymbol{f}_\psi$, which takes the augmented representations from $\Phi_{\boldsymbol{\omega}_r}(\bullet)$ as input and computes embeddings that can operate across all $\theta' \in \{\theta' : ||\theta - \theta'|| \leq \epsilon\}$ in robust manner. Specifically, it computes:

$$\boldsymbol{\eta}_{\psi,j}(G_q, \theta) = \boldsymbol{f}_\psi\big(\Phi_{\omega_1}(\boldsymbol{a}_j(G_q, \theta)), \ldots, \Phi_{\omega_R}(\boldsymbol{a}_j(G_q, \theta))\big), \tag{10}$$

$$\boldsymbol{\zeta}_{\psi,j}(G_c, \theta) = \boldsymbol{f}_\psi\big(\Phi_{\omega_1}(\boldsymbol{b}_j(G_c, \theta)), \ldots, \Phi_{\omega_R}(\boldsymbol{b}_j(G_c, \theta))\big). \tag{11}$$

To adapt $\boldsymbol{\eta}_{\psi,j}(G_q, \theta)$ and $\boldsymbol{\zeta}_{\psi,j}(G_c, \theta)$ over the NSR parameter perturbations, the adversary solves a min-max binary cross entropy (BCE) objective, where it trains $\boldsymbol{f}_\psi$ against the worst-case model perturbation within an ϵ-ball around the NSR model parameters θ:

$$\min_{\psi} \max_{\theta' : ||\theta' - \theta|| < \epsilon} \sum_{c,q,j} \text{BCE}(\text{sigmoid}(-\boldsymbol{\eta}_{\psi,j}(G_q, \theta')^\top \boldsymbol{\zeta}_{\psi,j}(G_c \oplus \delta_c, \theta')), y(G_q, G_c)) \tag{12}$$

Here, $\boldsymbol{\eta}_{\psi,j}(G_q, \theta)^\top \boldsymbol{\zeta}_{\psi,j}(G_c, \theta)$ approximates the relevance *distance* $\mathrm{d}_\theta(G_q, G_c)$. Hence, we apply the sigmoid to its negative, so that larger scores correspond to higher predicted relevance. Such training makes $\boldsymbol{\eta}_{\psi,j}(G_q, \theta)$ and $\boldsymbol{\zeta}_{\psi,j}(G_c, \theta)$ robust to new unseen queries. Appendix D illustrates the variables and functions involved in QUPAC.

Indexing Using the embeddings $\boldsymbol{\eta}_{\psi,j}(G_q, \theta)$ and $\boldsymbol{\zeta}_{\psi,j}(G_c, \theta)$, the adversary computes an approximate attack score $A_q(c, \delta_c) \approx \widehat{A}_q(c, \delta_c)$ as follows:

$$\widehat{A}_q(c, \delta_c) = \sum_{j \in \dim_h} [y(G_q, G_c) \cdot \boldsymbol{\eta}_{\psi,j}(G_q, \theta)]^\top \big(\boldsymbol{\zeta}_{\psi,j}(G_c, \theta) - \boldsymbol{\zeta}_{\psi,j}(G_c \oplus \delta_c, \theta)\big) \tag{13}$$

Thus, for each corpus graph G_c and perturbation δ_c, we obtain a precomputable corpus vector $\boldsymbol{x}_j(G_c, \delta_c) = \boldsymbol{\zeta}_{\psi,j}(G_c, \theta) - \boldsymbol{\zeta}_{\psi,j}(G_c \oplus \delta_c, \theta)$ for each dimension $j \in [\dim_h]$, while the relevance appears on the query side. The adversary can therefore index $\{\boldsymbol{x}_j(G_c, \delta_c) : c \in C, \delta_c \in \Gamma_c\}$ separately for each dimension j. Specifically, it uses locality-sensitive hashing (LSH) (Charikar, 2002; Indyk et al., 1998) where for each dimension j, it draws $\boldsymbol{W}_j \sim \mathcal{N}(0, \boldsymbol{I})$ of compatible dimensions and assigns each corpus item $\boldsymbol{x}(G_c, \delta_c)$ to $\text{bucket}_j = \text{sign}(\boldsymbol{W}_j \boldsymbol{x}(G_c, \delta_c))$.

Query-specific subset retrieval For each query G_q with $q \in Q$ and each dimension j, the adversary probes the LSH index twice, using $+\boldsymbol{\eta}_{\psi,j}(G_q, \theta)$ and $-\boldsymbol{\eta}_{\psi,j}(G_q, \theta)$ as query vectors, and retrieves perturbed corpus graphs from $\text{bucket}_j = \pm\text{sign}(\boldsymbol{W}_j \boldsymbol{\eta}_{\psi,j}(G_q, \theta))$. The adversary then aggregates the retrieved graphs across all dimensions j and both polarities of $\boldsymbol{\eta}_{\psi,j}$ to form the query-specific subset S_q and the corresponding perturbation set $\{\delta_c : c \in S_q\}$.

Fusion into a reduced ground set Given a budget $\mathcal{B}$ for the selected graph, consider the set of perturbation pairs $\{(c, \delta_c) : c \in \widehat{C}\}$ where $\widehat{C}$ is the pruned corpus set. Then, the adversary aggregates query-specific candidates by majority voting, where it selects the top-$\mathcal{B}$ pairs of (c, δ_c) that appear most across Q. These give the reduced ground set $\widehat{C}$ and perturbations $\{\delta_c : c \in \widehat{C}\}$. Line 4 of Algorithm 2 then selects $\text{argmax}_{c \in \widehat{C} \setminus S} \Delta_F(c : S, Q)$.

4 EXPERIMENTS

Datasets We use five graph datasets from the TUDatasets collection (Morris et al., 2020) that are standard in the NSR literature (Ranjan et al., 2022; Roy et al., 2022; Ramachandran et al., 2024; Raj et al., 2025), *viz.*, PTC-FM (FM), PTC-FR (FR), PTC-MM (MM), PTC-MR (MR), and AIDS. AIDS results and other details are in Appendix F.

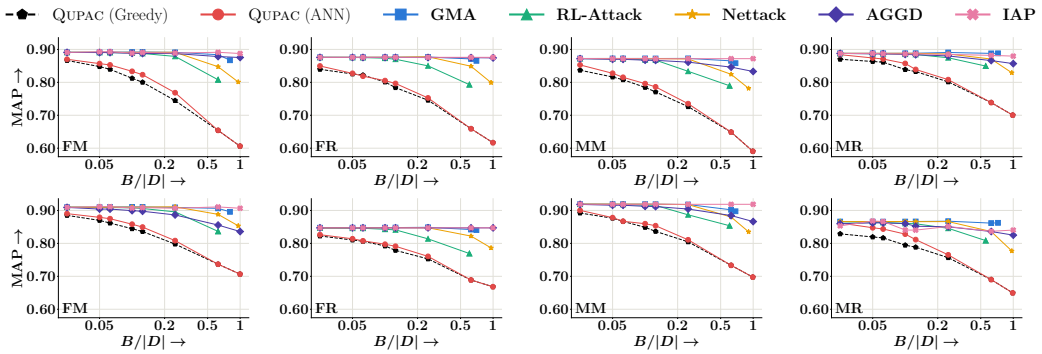


Figure 1: Performance measured in terms of MAP degradation (lower indicates a more successful attack) on the IsoNet++ (Edge) model, for QUPAC (Greedy), QUPAC (ANN) and all the baselines. Top (bottom) row shows results for white box (black box). The past query workload Q consists of 20% of test queries, and we measure performance on the remaining 80%. QUPAC outperforms the baselines by a significant margin.

Baselines To the best of our knowledge, research on adversarial attacks that perform a one-time corpus attack on NSR systems is scarce in the literature. Hence, we adapt five adversarial attack methods from other domains to compare against QUPAC, *viz.*, **(1)** Nettack (Zügner et al., 2018), **(2)** RL-Attack (Dai et al., 2018), **(3)** GMA (Zhang et al., 2020b), **(4)** IAP (Zhong et al., 2023) and **(5)** AGGD (Su et al., 2025). These methods select corpus set S and the perturbations δ_c for each $c \in S$, following different heuristics (Appendix F contains details).

Victim and substitute architectures In the main paper, we consider IsoNet (Edge) (Roy et al., 2022) and IsoNet++ (Edge) (Ramachandran et al., 2024), and GMN (Li et al., 2019) as our choices of victim and substitute models. We present results of other NSR models in Appendix G. Regardless of the substitute, we use the GEN model to train the embeddings for the ANN index.

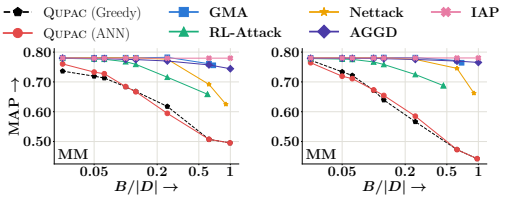
White box (WB) and Black box (BB) attacks We evaluate the attack under three settings: **(1)** white box, where the substitute is identical to the victim in both architecture and parameters; **(2)** black box, where the victim and substitute have identical architectures but different parameterizations, resulting from training in different environments; and **(3)** *transfer* where the substitute and victim use different model architectures.

Evaluation Given a query set and a corpus $\mathcal{G} = \{G_c : c \in C\}$, we partition the queries into training, validation, and test sets using a 60:15:25 split. We train the victim model on the training queries and the corpus $\mathcal{G}$. We set Q as the workload available to the attacker, consisting of 20% of the test set and Q^* as the unseen query set consisting of the remaining 80%. Given a substitute model d_θ, the adversary uses Q to compute $(S, \{\delta_c\})$ and applies the resulting perturbations to $\mathcal{G}$, yielding the modified corpus $\mathcal{G}_{S,\{\delta_c:c\in S\}} = \{\overline{G}_c\}$. For each unseen query G_q from Q^*, the victim NSR computes the relevance distance for every graph in the modified corpus and returns the corresponding ranked list. We compute the average precision (AP) of each ranked list and report the mean average precision (MAP) across $q \in Q^*$. We predominantly consider two variants of QUPAC: QUPAC (Greedy), where the adversary applies greedy routine (Algorithm 2) and QUPAC (ANN), where it uses ANN based method (Algorithm 4) to select S and $\{\delta_c\}$. We set $\mathcal{B} = 400B$.

4.1 RESULTS

Performance in WB and BB attacks We compare QUPAC against the baselines in terms of MAP degradation for different budgets B of the subset S (4). In Figure 1, we make the following observations. **(1)** QUPAC outperforms the baselines by a significant margin in both WB and BB settings. Among the baselines, RL-Attack performs the best. **(2)** QUPAC (Greedy) and QUPAC (ANN) perform comparably across the budget sweep, with QUPAC (Greedy) marginally ahead at lower budgets.

Transfer attack We now evaluate QUPAC in a transfer attack where the victim is GMN and substitute have different architectures. Figure 2 shows the results on the MM dataset. We make the following observations. **(1)** The adversarial attack transfers effectively to the victim architecture. **(2)** The attack is slightly more successful across all methods at lower budgets when the substitute architecture is IsoNet++ (Edge). Since both IsoNet++ (Edge) and GMN are early-interaction ar-



d_θ = IsoNet++ (Edge) d_θ = IsoNet (Edge)

Figure 2: Transfer attack efficacy.

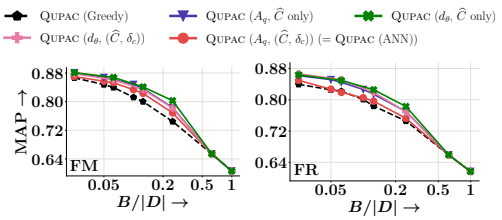


Figure 3: $(S, \{\delta_c\})$ selection and A_q based indexing.

chitectures, IsoNet++ (Edge) is a better substitute than late-interaction architectures such as IsoNet (Edge), clearly evident at lower budgets.

Ablation study on joint $(S, \{\delta_c\})$ selection and A_q based indexing The proposed ANN-based ground-set reduction method jointly selects (c, δ_c) to maximize $A_q(c, \delta_c) \approx \widehat{A}_q(c, \delta_c)$. Here, we perform ablations across these two different axes: **(i)** pruned corpus $\widehat{C}$ only: Instead of joint selection, the adversary first considers the top-$\mathcal{B}$ items based on votes$(c) = \sum_q |S_q|$, retrieves $\widehat{C}$, and then computes S and δ_c using Algorithm 2. **(ii)** d_θ based indexing: Instead of our attack score, the adversary builds the index tailored to maximizing the relevance distance d_θ for the *perturbed* graphs. This yields four adversary variants of QUPAC in addition to greedy from the design space consisting of $\{\mathrm{d}_\theta$ based indexing, A_q based indexing$\} \times \{\widehat{C}$ only, $(S, \delta_c)\}$. Figure 3 reports adversarial MAP for all four methods on IsoNet++ (Edge) across the budget sweep. We observe that **(1)** joint selection $(\widehat{C}, \delta_c)$ performs better than the $\widehat{C}$ only variant for both distances used in indexing (Green vs Pink and Purple vs Red); **(2)** Indexing based on our proposed attack score A_q performs better than d_θ (Green vs Purple and Pink vs Red); and **(3)** QUPAC (ANN) uses both joint selection and A_q-based indexing, resulting in the lowest MAP.

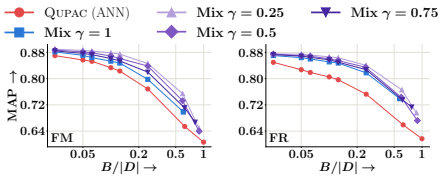


Figure 4: Ablation on subset selection.

Ablation study on subset selection Here, we ablate our proposed subset selection strategy by replacing our ANN-based pruning method with an alternative (Mix (γ)). We compute S_+ (S_-) as the top-B' (bottom-B') graphs that are relevant (non-relevant) to each query graph G_q with $q \in Q$. Then, we mix γ fraction from S_+ and $(1-\gamma)$ fraction from S_- to obtain S. Finally, we obtain $\delta_c(S) = \mathrm{argmax}_{\delta_c \in \Gamma_c} \mathcal{L}(S, \{\delta_c\}, Q)$. Figure 4 shows the effect of increasing γ for FM and FR datasets on IsoNet++ (Edge). Attack performance improves as γ increases and QUPAC (ANN) performs best.

Variation with pruned set size $|\widehat{C}| = \mathcal{B}$ We study the sensitivity of QUPAC to $\mathcal{B}$, which controls the number of candidates retrieved from the ANN index before greedy selection. Figure 5 reports adversarial MAP on the FM and FR datasets for victim IsoNet (Edge) as $\mathcal{B}$ varies from $200B$ to $400B$. We observe that QUPAC is largely robust to $\mathcal{B}$. This suggests that the QUPAC (ANN) already captures the most important graph–perturbation pairs at $\mathcal{B} = 200B$. This allows us to use a smaller candidate pool without compromising the quality of the adversarial attack, while reducing the ANN retrieval cost and subsequent greedy-search cost.

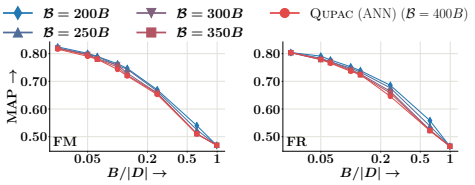


Figure 5: QUPAC is robust to $\mathcal{B}$.

5 Conclusion

We initiate a study of adversarial corpus-level attacks on NSR systems, a setting we argue is critically underexplored despite its practical importance. Here, the adversary designs attacks using historical query workloads that degrade the ranking of corpus graphs for new queries. The space of candidate graph subsets and edge edits is combinatorially large. Hence, an exact greedy algorithm becomes computationally expensive. Therefore, our adversary prunes the underlying search space. It first approximates the marginal gain of the greedy algorithm using an attack score and then build a neural surrogate of this attack score, which is amenable to ANN index. Subsequently, it builds an ANN index that allows the adversary to prune the ground set into small subset, which makes the greedy algorithm computationally efficient. Experiments across several datasets validate the effectiveness of our attack.

AI USE STATEMENT

In our work, we have used generative AI tools for assisting us in deriving and writing of proofs, helping us in adapting existing baselines to our attack settings, and fixing bugs in our code. We do not use AI tools for the following tasks: helping us develop the problem and core formulation, conceptual frameworks, providing mathematical claims, or for the interpretation of results. The following tasks are not applicable to our work: generating synthetic datasets, proposing or refining hypotheses, designing or providing feedback on research methodology or experiments, assisting with translation, cleaning/reformatting datasets, and supporting qualitative and thematic data analyses. We have reviewed all AI-assisted work for correctness. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI tools.

ETHICS STATEMENT

The scope of our work does not raise any ethical concerns. We use publicly available datasets under their respective usage licenses, and no human-subject data is used. QUPAC assumes that the adversary has full corpus write access during the corpus curation and storage stage of NSR retrieval, before the candidate neural retriever is deployed. If a victim NSR's owner delegates the corpus curation task to a third party vendor (pre-indexing stage), a malicious vendor has the scope to perform a query workload-guided, pre-indexing attack on the corpus. QUPAC's attack strategy can furthermore be used for robustness training of candidate NSR models. We can train the model on adversarial corpus items chosen from the adversarial corpus subset generated by QUPAC. This can ensure that the victim model's ranking quality does not degrade when exposed to adversarially perturbed items in the corpus. Hence, QUPAC can serve as a potential pre-indexing diagnostic tool to analyze the vulnerability of a candidate model to such attacks, and help make them more robust.

REPRODUCIBILITY STATEMENT

We provide the code and artifacts such as model checkpoints, hyperparameters and dataset. We add scripts in our code base for building the environment and running our experiments end to end. We also add `readme.md` for easily understanding the structure and entry points of our code.

REFERENCES

Alexandr Andoni, Piotr Indyk, and Ilya Razenshteyn. Approximate nearest neighbor search in high dimensions. In *Proceedings of the International Congress of Mathematicians: Rio de Janeiro 2018*, pp. 3287–3318. World Scientific, 2018.

Andrew An Bian, Joachim M Buhmann, Andreas Krause, and Sebastian Tschiatschek. Guarantees for greedy maximization of non-submodular functions with applications. In *International conference on machine learning*, pp. 498–507. PMLR, 2017.

Aleksandar Bojchevski and Stephan Günnemann. Certifiable robustness to graph perturbations. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019a.

Aleksandar Bojchevski and Stephan Günnemann. Adversarial attacks on node embeddings via graph poisoning. In *International conference on machine learning*, pp. 695–704. PMLR, 2019b.

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks, 2017. URL https://arxiv.org/abs/1608.04644.

Pritish Chakraborty, Indradyumna Roy, Soumen Chakrabarti, and Abir De. Contextual tokenization for graph inverted indices. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Moses S. Charikar. Similarity estimation techniques from rounding algorithms. In *Proceedings of the Thirty-Fourth Annual ACM Symposium on Theory of Computing*, STOC '02, pp. 380–388, New York, NY, USA, 2002. Association for Computing Machinery. ISBN 1581134959. doi: 10.1145/509907.509965.

Hanjun Dai, Hui Li, Tian Tian, Xin Huang, Lin Wang, Jun Zhu, and Le Song. Adversarial attack on graph structured data. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th*

International Conference on Machine Learning, volume 80 of *Proceedings of Machine Learning Research*, pp. 1115–1124. PMLR, 10–15 Jul 2018.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. The faiss library, 2025. URL https://arxiv.org/abs/2401.08281.

Marwa El Halabi and Stefanie Jegelka. Optimal approximation for unconstrained non-submodular minimization. In *International Conference on Machine Learning*, pp. 3961–3972. PMLR, 2020.

Logan Engstrom, Brandon Tran, Dimitris Tsipras, Ludwig Schmidt, and Aleksander Madry. Exploring the landscape of spatial robustness. In *International conference on machine learning*, pp. 1802–1811. PMLR, 2019.

Fuli Feng, Xiangnan He, Jie Tang, and Tat-Seng Chua. Graph Adversarial Training: Dynamically Regularizing Based on Graph Structure . *IEEE Transactions on Knowledge & Data Engineering*, 33(06):2493–2504, June 2021. ISSN 1558-2191. doi: 10.1109/TKDE.2019.2957786.

Khashayar Gatmiry and Manuel Gomez-Rodriguez. Non-submodular function maximization subject to a matroid constraint, with applications. *arXiv preprint arXiv:1811.07863*, 2018.

Simon Geisler, Tobias Schmidt, Hakan Şirin, Daniel Zügner, Aleksandar Bojchevski, and Stephan Günnemann. Robustness of graph neural networks at scale. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 7637–7649. Curran Associates, Inc., 2021.

A. Gionis, Piotr Indyk, and Rajeev Motwani. Similarity search in high dimensions via hashing. In *Very Large Data Bases Conference*, 1999.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *CoRR*, abs/1412.6572, 2014.

Zhongliang Guo, Yifei Qian, Yanli Li, Weiye Li, Chun Tong Lei, Shuai Zhao, Lei Fang, Ognjen Arandjelović, and Chun Pong Lau. Beyond vulnerabilities: A survey of adversarial attacks as both threats and defenses in computer vision systems. 2025. URL https://arxiv.org/abs/2508.01845.

Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. Black-box adversarial attacks with limited queries and information. In Jennifer G. Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2142–2151. PMLR, 2018.

Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pp. 604–613, 1998.

Andreas Krause, H Brendan McMahan, Carlos Guestrin, and Anupam Gupta. Robust submodular observation selection. *Journal of Machine Learning Research*, 9(12), 2008.

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world, 2017. URL https://arxiv.org/abs/1607.02533.

Ping Li and Christian König. b-bit minwise hashing. In *Proceedings of the 19th International Conference on World Wide Web (WWW)*, pp. 671–680. ACM, 2010. doi: 10.1145/1772690.1772759.

Yongkang Li, Panagiotis Eustratiadis, Simon Lupart, and Evangelos Kanoulas. Unsupervised corpus poisoning attacks in continuous space for dense retrieval. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 2452–2462, 2025.

Yujia Li, Chenjie Gu, Thomas Dullien, Oriol Vinyals, and Pushmeet Kohli. Graph matching networks for learning the similarity of graph structured objects. In *International conference on machine learning*, pp. 3835–3845. PMLR, 2019. URL https://arxiv.org/abs/1904.12787.

Bin Liang, Hongcheng Li, Miaoqiang Su, Pan Bian, Xirong Li, and Wenchang Shi. Deep text classification can be fooled. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, IJCAI-2018, pp. 4208–4215. International Joint Conferences on Artificial Intelligence Organization, 2018. doi: 10.24963/ijcai.2018/585.

Zilong Lin, Zhengyi Li, Xiaojing Liao, XiaoFeng Wang, and Xiaozhong Liu. Mawseo: Adversarial wiki search poisoning for illicit online promotion. In *2024 IEEE Symposium on Security and Privacy (SP)*, pp. 388–406. IEEE, May 2024. doi: 10.1109/sp54263.2024.00049.

Ting Liu, Andrew Moore, Ke Yang, and Alexander Gray. An investigation of practical approximate nearest neighbor algorithms. In L. Saul, Y. Weiss, and L. Bottou (eds.), *Advances in Neural Information Processing Systems*, volume 17. MIT Press, 2004.

Yu-An Liu, Ruqing Zhang, Jiafeng Guo, Maarten de Rijke, Wei Chen, Yixing Fan, and Xueqi Cheng. Black-box adversarial attacks against dense retrieval models: A multi-view contrastive learning method, 2023a. URL https://arxiv.org/abs/2308.09861.

Yu-An Liu, Ruqing Zhang, Jiafeng Guo, Maarten de Rijke, Wei Chen, Yixing Fan, and Xueqi Cheng. Topic-oriented adversarial attacks against black-box neural ranking models, 2023b. URL https://arxiv.org/abs/2304.14867.

Baharan Mirzasoleiman, Ashwinkumar Badanidiyuru, Amin Karbasi, Jan Vondrák, and Andreas Krause. Lazier than lazy greedy. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

Jason Mohoney, Devesh Sarda, Mengze Tang, Shihabur Rahman Chowdhury, Anil Pacaci, Ihab F Ilyas, Theodoros Rekatsinas, and Shivaram Venkataraman. Quake: adaptive indexing for vector search. In *19th USENIX Symposium on Operating Systems Design and Implementation (OSDI 25)*, pp. 153–169, 2025.

Christopher Morris, Nils M Kriege, Franka Bause, Kristian Kersting, Petra Mutzel, and Marion Neumann. Tudataset: A collection of benchmark datasets for learning with graphs. *arXiv preprint arXiv:2007.08663*, 2020.

Kartik Nair, Pritish Chakraborty, Atharva Tambat, Indradyumna Roy, Soumen Chakrabarti, Anirban Dasgupta, and Abir De. A dense subset index for collective query coverage. In *International Conference on Learning Representations*, volume 2026, pp. 67280–67319, 2026.

George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functions—i. *Mathematical programming*, 14(1):265–294, 1978.

Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 427–436, 2015.

Luca Piccolboni, Alessandro Menon, and Graziano Pravadelli. Efficient control-flow subgraph matching for detecting hardware trojans in rtl models. *ACM Transactions on Embedded Computing Systems (TECS)*, 16(5s):1–19, 2017.

Vaibhav Raj, Indradyumna Roy, Ashwin Ramachandran, Soumen Chakrabarti, and Abir De. Charting the design space of neural graph representations for subgraph matching. *arXiv preprint arXiv:2510.22897*, 2025.

Ashwin Ramachandran, Vaibhav Raj, Indradyumna Roy, Soumen Chakrabarti, and Abir De. Iteratively refined early interaction alignment for subgraph matching based graph retrieval. *Advances in Neural Information Processing Systems*, 37:77593–77629, 2024.

Rishabh Ranjan, Siddharth Grover, Sourav Medya, Venkatesan Chakaravarthy, Yogish Sabharwal, and Sayan Ranu. Greed: A neural framework for learning graph distance functions. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, November 29-December 1, 2022*, 2022.

Nisarg Raval and Manisha Verma. One word at a time: adversarial attacks on retrieval models, 2020. URL https://arxiv.org/abs/2008.02197.

Rex, Ying, Zhaoyu Lou, Jiaxuan You, Chengtao Wen, Arquimedes Canedo, and Jure Leskovec. Neural subgraph matching, 2020. URL https://arxiv.org/abs/2007.03092.

Daniel Rose, Oliver Wieder, Thomas Seidel, and Thierry Langer. Pharmacomatch: Efficient 3d pharmacophore screening via neural subgraph matching. *arXiv preprint arXiv:2409.06316*, 2024.

Indradyumna Roy, Venkata Sai Baba Reddy Velugoti, Soumen Chakrabarti, and Abir De. Interpretable neural subgraph matching for graph retrieval. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pp. 8115–8123, 2022.

Indradyumna Roy, Rishi Agarwal, Soumen Chakrabarti, Anirban Dasgupta, and Abir De. Locality sensitive hashing in fourier frequency domain for soft set containment search. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, 2023.

Vaishaal Shankar, Jordan Zhang, Jerry Chen, Christopher Dinh, Mattthew Clements, and Avideh Zakhor. Approximate subgraph isomorphism for image localization. *Electronic Imaging*, 28:1–9, 2016.

Anshumali Shrivastava and Ping Li. Asymmetric LSH (ALSH) for sublinear time maximum inner product search (MIPS). In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 27, pp. 2321–2329, 2014.

Harsha Vardhan Simhadri, Ravishankar Krishnaswamy, Gopal Srinivasa, Suhas Jayaram Subramanya, Andrija Antonijevic, Dax Pryce, David Kaczynski, Shane Williams, Siddarth Gollapudi, Varun Sivashankar, Neel Karia, Aditi Singh, Shikhar Jaiswal, Neelam Mahapatro, Philip Adams, Bryan Tower, and Yash Patel. DiskANN: Graph-structured indices for scalable, fast, fresh and filtered approximate nearest neighbor search, 2023.

Jinyan Su, Preslav Nakov, and Claire Cardie. Corpus poisoning via approximate greedy gradient descent. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 4274–4294, 2025.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Florian Tramèr, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. The space of transferable adversarial examples. *arXiv preprint arXiv:1704.03453*, 2017.

Binghui Wang and Neil Zhenqiang Gong. Attacking graph-based classification via manipulating the graph structure. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, CCS '19, pp. 2023–2040, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450367479. doi: 10.1145/3319535.3354206.

Jingdong Wang, Ting Zhang, Jingkuan Song, Nicu Sebe, and Heng Tao Shen. A survey on learning to hash. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):769–790, 2018. doi: 10.1109/TPAMI.2017.2699960.

Yair Weiss, Antonio Torralba, and Rob Fergus. Spectral hashing. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 21, pp. 1753–1760, 2008.

Peter Willett, John M Barnard, and Geoffrey M Downs. Chemical similarity searching. *Journal of chemical information and computer sciences*, 38(6):983–996, 1998.

Kaidi Xu, Sijia Liu, Pin-Yu Chen, Mengshu Sun, Caiwen Ding, Bhavya Kailkhura, and Xue Lin. Towards an efficient and general framework of robust training for graph neural networks. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8479–8483, 2020. doi: 10.1109/ICASSP40776.2020.9054465.

Hengtong Zhang, Tianhang Zheng, Jing Gao, Chenglin Miao, Lu Su, Yaliang Li, and Kui Ren. Data poisoning attack against knowledge graph embedding. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pp. 4853–4859. International Joint Conferences on Artificial Intelligence Organization, 7 2019. doi: 10.24963/ijcai.2019/674.

Xiang Zhang and Marinka Zitnik. Gnnguard: defending graph neural networks against adversarial attacks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020a. Curran Associates Inc. ISBN 9781713829546.

Zijie Zhang, Zeru Zhang, Yang Zhou, Yelong Shen, Ruoming Jin, and Dejing Dou. Adversarial attacks on deep graph matching. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 20834–20851. Curran Associates, Inc., 2020b.

Zexuan Zhong, Ziqing Huang, Alexander Wettig, and Danqi Chen. Poisoning retrieval corpora by injecting adversarial passages. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 13764–13775, 2023.

Dingyuan Zhu, Ziwei Zhang, Peng Cui, and Wenwu Zhu. Robust graph convolutional networks against adversarial attacks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, pp. 1399–1407, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450362016. doi: 10.1145/3292500.3330851.

Wei Zou, Runpeng Geng, Binghui Wang, and Jinyuan Jia. {PoisonedRAG}: Knowledge corruption attacks to {Retrieval-Augmented} generation of large language models. In *34th USENIX Security Symposium (USENIX Security 25)*, pp. 3827–3844, 2025.

Daniel Zügner and Stephan Günnemann. Certifiable robustness of graph convolutional networks under structure perturbations. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, pp. 1656–1665, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450379984. doi: 10.1145/3394486.3403217.

Daniel Zügner, Amir Akbarnejad, and Stephan Günnemann. Adversarial attacks on neural networks for graph data. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '18, pp. 2847–2856, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450355520. doi: 10.1145/3219819.3220078.

Daniel Zügner and Stephan Günnemann. Adversarial attacks on graph neural networks via meta learning. In *International Conference on Learning Representations*, 2019.

A LIMITATIONS

We find the following limitations of our work:

1. **Access to solver for ground-truth label computations:** For the query workload, the adversary uses a solver to compute the ground truth labels of the perturbed graph. These labels allow it to construct Γ_{qc} and Γ_c. However, this restriction can be mitigated as follows. In addition to the prediction, one can also learn the model prediction error given by an NSR model. Now, one can use this prediction error or model confidence to obtain the difficult-to-predict query corpus graph pairs. The adversary can use solver only for these hard pairs to predict their labels, whereas it would trust the model output otherwise.
2. **Evolving Corpus:** Our setting assumes a fixed corpus that can be indexed ahead of the attack based on the visible query workload. However, in cases of corpora that evolve with time, our attack will not be able to account for the new corpus items, since the index will now be stale. The subset of corpus items $S \subseteq C$ and optimal perturbations $\{\delta_c(S) : c \in S\}$ might be sub-optimal for the updated corpus. However, this problem can be mitigated using recent works that perform indexing on evolving corpora (Mohoney et al., 2025).

B BROADER IMPACT

QUPAC assumes a scenario where an adversary is aware of the nature of all future test queries via access to a finite query workload. This would allow the adversary to attack the corpus in a way that the retrieval quality is maximally degraded for the expected queries, which is far more damaging than a query-agnostic attack on the corpus. We depict both white box and black box attacks, which shows considerable damage to a NSR's retrieval quality. Hence, even when the adversary's notion of the query workload is not exact, such an attack poses significant risks to a victim NSR.

QUPAC proposes an attack method that can analyze the potential vulnerability of a candidate NSR prior to its deployment, so that it can be made robust against such attacks.

C RELATED WORK

Adversarial ML in computer vision. Pioneering work on adversarial attacks and defenses started in the domain of computer vision with Szegedy et al. (2013) showing that neural networks learn fairly discontinuous functions resulting in the existence of adversarial examples. Several works following this (Nguyen et al., 2015; Engstrom et al., 2019) show simple algorithms for generating imperceptible input transformations to fool neural models. Tramèr et al. (2017) characterized the space of adversarial examples as a high dimensional subspace of the input space establishing the transferability of adversarial attacks; attack on one model architecture is effective against another. Several subsequent works (Szegedy et al., 2013; Goodfellow et al., 2014; Kurakin et al., 2017; Carlini et al., 2017; Ilyas et al., 2018) devise adversarial attacks on images in various settings ranging from restricted access to the victim model to the scope of the attack being restricted to a single example (targeted) or the whole dataset (global).

Adversarial attacks and defenses in graph ML. Many existing works tackle adversarial attacks in the domain of graph based learning. Pioneering works such as Nettack (Zügner et al., 2018) and RL-Attack (Dai et al., 2018) develop adversarial attacks for targeted node classification. Several works (Bojchevski et al., 2019b; Zügner et al., 2019; Geisler et al., 2021) perform data poisoning attacks during training that results in performance degradation during test. Certifiability works (Bojchevski et al., 2019a; Zügner et al., 2020) devise methods for evaluating robustness of models at certain points in the input space where a targeted adversarial attack is guaranteed to fail. Several subsequent works (Zhu et al., 2019; Wang et al., 2019; Zhang et al., 2019; 2020b) improve upon the state of the art attack methods. Following the adversarial attacks, there have been many works developing adversarial defenses (Feng et al., 2021; Zhang et al., 2020a; Xu et al., 2020) for various graph learning problems including node classification and graph classification. Our work tackles a different space: adversarial attacks on *subgraph matching* based graph retrievers.

Hashing and indexing. Locality-sensitive hashing (LSH) was introduced by Indyk et al. (1998) to solve approximate nearest-neighbor search in sub-linear time by hashing items so that nearby items collide more often than far ones. Andoni et al. (2018) later gave near-optimal hash families for Euclidean and angular distances, tightening the query-time/space trade-offs of the original

construction. Li et al. extended LSH beyond symmetric similarity. Specifically, they developed minwise hashing (Li et al., 2010) for Jaccard-similarity. Shrivastava et al. (2014) showed that maximum inner-product search (MIPS) is not directly amenable to standard LSH. But they can be solved via asymmetric transforms before hashing, a trick our own ANN-compatible reformulation of the attack score (Eqs. (7), (13)) also rely on. Weiss et al. (2008) propose spectral hashing, which learns the underlying hyperplane rather than drawing from Normal distribution. Wang et al. (2018) survey this broader learning-to-hash literature.

Neural subgraph retrieval. Our work is adjacent to neural subgraph retrieval. Neuromatch (Rex et al., 2020) developed loss objective to induce structure in the graph embeddings that naturally incorporated subgraph relation in the embeddings. GEN and GMN (Li et al., 2019) are single vector late and early interaction graph matching models and GMN uses attention mechanism for early interaction resulting in improved performance compared to GEN. Subsequent works such as IsoNet (Roy et al., 2022) and IsoNet++ (Ramachandran et al., 2024) use multi-vector graph representations. IsoNet is a late interaction model while IsoNet++ is early interaction and both methods use Gumbel-Sinkhorn network based graph alignment used downstream to rank the corpus of graphs. Our work evaluates the robustness of these models.

More related to our setting, FourierHashNet (Roy et al., 2023) builds locality-sensitive hash functions in the Fourier frequency domain for soft set-containment search, applied to accelerate subgraph-containment-style retrieval over large graph collections. CORGII (Chakraborty et al., 2025) instead learns a sparse, vocabulary-based binary encoding of dense graph representations. Both assume uncorrupted corpus and focus on efficient retrieval. In the current work, we instead repurpose ANN/LSH as an *adversary* to prune the ground set in a pre-indexing attack.

Adversarial attacks in information retrieval. Retrieval pipeline operates in two stages: **(1) retrieval:** retrieve items from the corpus and **(2) ranking:** rank the retrieved items and present the top-k items to the user. Adversarial attacks on IR systems target one or both stages. Retrieval-stage attacks inject items to or perturb the corpus items so that non-relevant items are retrieved Lin et al. (2024); Liu et al. (2023a); Zhong et al. (2023), while ranking stage attacks perturb already retrieved items to boost its ranking (Liang et al., 2018; Liu et al., 2023b; Raval et al., 2020). IAP (Zhong et al., 2023) and AGGD (Su et al., 2025) add spurious items to the corpus and evaluate the success of their attack in terms of the number of times the injected items are retrieved by the retrieval system representative of retrieval-stage attacks. Majority of the literature operates on text corpora, whereas we perform attacks on subgraph retrieval.

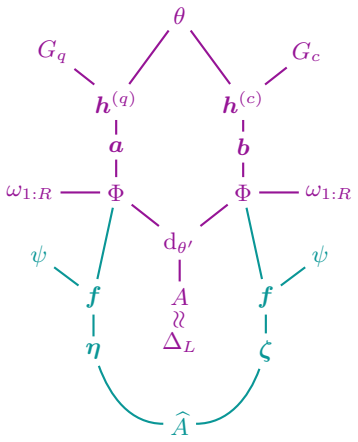


Figure 6: Overview of QUPAC network, showing optimized variables, sampled variables, and various neural functions.

D OVERVIEW OF QUPAC NETWORK

Figure 6 shows the key variables and functions of QUPAC. In the purple part, $\boldsymbol{h}^{(q)}$ and $\boldsymbol{h}^{(c)}$ are produced using a GNN parameterized by θ, acting on graphs G_q and G_c. $\boldsymbol{a}$ and $\boldsymbol{b}$ are reshaping and padding functions that enable us to describe relevance distance d_θ as a hinge of an inner product. Randomly sampled $\omega_{1:R}$ are used to augment the embeddings $\boldsymbol{a}$ and $\boldsymbol{b}$. d_θ is used to estimate the marginal gain A of the attacker. The teal part pre-adapts the attack to drifts from the historic query workload to the workload faced during deployment. Additional parameters ψ are introduced, and a surrogate distance $\widetilde{\mathrm{d}}_{\theta'}$ computed in terms of functions $\boldsymbol{f}, \boldsymbol{\eta}_{\psi,j}(G_q, \theta), \boldsymbol{\zeta}_{\psi,j}(G_c, \theta)$. Here θ' is allowed to vary in an ϵ-ball around θ while optimizing ψ. $\boldsymbol{\eta}_{\psi,j}(G_q, \theta), \boldsymbol{\zeta}_{\psi,j}(G_c, \theta)$ allow the attacker to estimate a drift-robust surrogate marginal gain $\widehat{A}$.

E PROOFS OF THE TECHNICAL RESULTS

E.1 NOTATION

$\mathcal{G} := \{G_c : c \in C\}$ is the set of corpus graphs. For $S \subseteq C$, let $S, \{\delta_c : c \in S\}$ define a perturbation of $\mathcal{G}$ to yield $\mathcal{G}_{S,\{\delta_c : c \in S\}}$ as follows:

$$\overline{G}_c = \begin{cases} G_c \oplus \delta_c & \text{if } c \in S, \\ G_c & \text{if } c \notin S. \end{cases} \quad \mathcal{G}_{S,\{\delta_c : c \in S\}} = \{\overline{G}_c : c \in C\} = \{G'_c : c \in S\} \cup \{G_c : c \notin S\} \tag{14}$$

Here, $\delta_c \in \Gamma_c$ ensures that $y(G_q, G_c) = y(G_q, G_c \oplus \delta_c)\ \forall q$. Hence, under any choice of $S, \{\delta_c : c \in S\}$, we can partition $\mathcal{G}_{S,\{\delta_c : c \in S\}}$ into $C_q^+ =: \{c \in C : y(G_q, G_c) = 1\}$ and $C_q^- =: \{c \in C : y(G_q, G_c) = -1\}$.

Pairs involving G_c. We define a set containing all the pairs that will appear in the loss terms $\ell(\bullet, G_c)$ or $\ell(G_c, \bullet)$.

$$\mathcal{C}_q(c) := \{j \in C : y(G_q, G_j) \cdot y(G_q, G_c) = -1\}$$

Deficit for each $\overline{G}_{c_\ominus}, \overline{G}_{c_\oplus}$. Consider

$$x_j(S) = \begin{cases} \gamma + \mathrm{d}_\theta(G_q, \overline{G}_c) - \mathrm{d}_\theta(G_q, \overline{G}_j), & y(G_q, G_c) = +1 \\ \gamma + \mathrm{d}_\theta(G_q, \overline{G}_j) - \mathrm{d}_\theta(G_q, \overline{G}_c), & y(G_q, G_c) = -1 \end{cases} \tag{15}$$

We quantify the amount by which $x_j(S)$ falls below zero as

$$D_j(S) := [-x_j(S)]_+ \geq 0. \tag{16}$$

E.2 ASSUMPTIONS

Assumption 1 (Adversarially meaningful perturbations)**.** For every $c \in C, \delta_c \in \Gamma_c$ and $q \in Q$,

$$A_q(c, \delta_c) = y(G_q, G_c)\Big(\mathrm{d}_\theta(G_q, G_c \oplus \delta_c) - \mathrm{d}_\theta(G_q, G_c)\Big) \geq 0 \tag{17}$$

Assumption 2. For every $q \in Q, c_1 \in C$, the relevance distance $\mathrm{d}_\theta(G_q, \overline{G}_{c_1})$ is a function of G_{c_1}, δ_{c_1} alone, it does not depend on $\{\delta_{c_2} : c_2 \in S, c_1 \neq c_2\}$.

Assumption 3. $\exists \tau \geq 0, \kappa > 1, \beta_0 \in (0, 1]$, such that for every $q \in Q, c \in C$:

(1) At least β_0 fraction of pairs involving G_c are $\geq \tau$: $\big|\{j \in \mathcal{C}_q(c) : D_j(\phi) \leq \tau\}\big| \geq \beta_0 |\mathcal{C}_q(c)|$.

(2) For every admissible perturbation, either $A_q(c, \delta_c) = 0$ or it is larger than the threshold τ by atleast a factor of κ: $A_q(c, \delta_c) \geq \kappa\tau$.

Assumption 4 (Non-trivial standalone gain)**.** $\Delta_F(c|\emptyset, Q) > 0$ for every $c \in C$.

E.3 PROOF OF THEOREM 1

Theorem 1. Let the set function $F(S, Q)$ be defined as in Eq. (5). Let Q denote the query workload, C the index for the set of corpus graphs, and γ be the tunable margin. Let Γ_c be as defined in Eq (2). Assume that $A_q(c, \delta_c) \geq 0$ for all $\delta_c \in \Gamma_c$. Then, we have the following properties. **(1)** The function $F(S, Q)$ is a monotone set function. **(2)** If there exists $\beta_0 \in (0, 1]$, such that, at least β_0 fraction of the positive-negative graph pairs $(\overline{G}_{c_\oplus}, \overline{G}_{c_\ominus})$ containing G_c satisfy $\mathrm{d}_\theta(G_q, \overline{G}_{c_\oplus}) - \mathrm{d}_\theta(G_q, \overline{G}_{c_\ominus}) + \gamma \geq 0$ then $F(\bullet, Q)$ is α-submodular with $\alpha \geq \beta_0$.

(1) Proof of monotonicity of $F(S, Q)$

Proof. Given query workload $Q, S \subseteq C$, and $c \in C \setminus S$. Let $\{\delta_w(S) : w \in S\}$ be an optimal perturbation set for S, so that $F(S, Q) = \mathcal{L}(S, \{\delta_w(S) : w \in S\}, Q)$.

We want to show that $\Delta_F(c : S, Q) = F(S \cup \{c\}, Q) - F(S, Q) \geq 0$.

Fix an arbitrary $\delta_c \in \Gamma_c$. Adding and subtracting $\mathcal{L}(S \cup \{c\}, \{\delta_w(S) : w \in S\} \cup \{\delta_c\}, Q)$ gives the identity,

$$\begin{aligned} \Delta_F(c : S, Q) = &[F(S \cup \{c\}, Q) - \mathcal{L}(S \cup \{c\}, \{\delta_w(S) : w \in S\} \cup \{\delta_c\}, Q)] \\ &+ [\mathcal{L}(S \cup \{c\}, \{\delta_w(S) : w \in S\} \cup \{\delta_c\}, Q) - \mathcal{L}(S, \{\delta_w(S) : w \in S\}, Q)] \end{aligned} \tag{18}$$

By Eq. (3), $\mathcal{L}(\bullet, Q) = \sum_{q \in Q} \mathcal{L}(\bullet, q)$. Notice that, $\forall \delta_c \in \Gamma_c$,

$$\mathcal{L}(S \cup \{c\}, \{\delta_w(S) : w \in S\} \cup \{\delta_c\}, Q) - \mathcal{L}(S, \{\delta_w(S) : w \in S\}, Q) = \sum_{q \in Q} \Delta_{\mathcal{L}}((c, \delta_c) : S, q)$$

This implies that $\Delta_F(c : S, Q) \geq \sum_{q \in Q} \Delta_{\mathcal{L}}((c, \delta_c) : S, q)$.

Fix a query $q \in Q$ and perturbation $\delta_c \in \Gamma_c$. By Assumption 2, we get

Case 1: $y(G_q, G_c) = +1$

$$\begin{aligned}\Delta_{\mathcal{L}}((c, \delta_c) : S, q) = \sum_{c_\ominus \in C_q^-} \Big(&\big[\gamma + \mathrm{d}_\theta(G_q, G_c \oplus \delta_c) - \mathrm{d}_\theta(G_q, \overline{G}_{c_\ominus})\big]_+ \\ &- \big[\gamma + \mathrm{d}_\theta(G_q, G_c) - \mathrm{d}_\theta(G_q, \overline{G}_{c_\ominus})\big]_+ \Big) \\ &\leq |\mathcal{C}_q(c)| A_q(c, \delta_c)\end{aligned} \tag{19}$$

Case 2: $y(G_q, G_c) = -1$

$$\begin{aligned}\Delta_{\mathcal{L}}((c, \delta_c) : S, q) = \sum_{c_\oplus \in C_q^+} \Big(&\big[\gamma + \mathrm{d}_\theta(G_q, \overline{G}_{c_\oplus}) - \mathrm{d}_\theta(G_q, G_c \oplus \delta_c)\big]_+ \\ &- \big[\gamma + \mathrm{d}_\theta(G_q, \overline{G}_{c_\oplus}) - \mathrm{d}_\theta(G_q, G_c)\big]_+ \Big) \\ &\leq |\mathcal{C}_q(c)| A_q(c, \delta_c)\end{aligned} \tag{20}$$

By Assumption 1, we see that $\Delta_{\mathcal{L}}((c, \delta_c) : S, q) \geq 0$. Using this result, we can further conclude that

$$\Delta_F(c : S, Q) \geq \sum_{q \in Q} \Delta_{\mathcal{L}}((c, \delta_c) : S, q) \geq 0. \tag{21}$$

Thus, $F(S, Q)$ is monotone. □

(2) Proof of α-submodularity of $F(\bullet, Q)$.

Proof. Fix $c \in C, S \subseteq T \subseteq C$ with $c \notin T$. We need to show that $\Delta_F(c : S, Q) \geq \alpha \Delta_F(c : T, Q)$ for some $\alpha > 0$.

We fix $q \in Q$, a corpus subset $S \subseteq C \setminus \{c\}$, set of feasible perturbations $\{\delta_w : w \in S, \delta_w \in \Gamma_w\}$. Then the marginal gain can be expressed as

$$\Delta_{\mathcal{L}}((c, \delta_c) : S, q) = \mathcal{L}(S \cup \{c\}, \{\delta_w : w \in S\} \cup \{\delta_c\}, q) - \mathcal{L}(S, \{\delta_w : w \in S\}, q) \tag{22}$$

Since by Assumption 1, $A_q(c, \delta_c) \geq 0$, we can conclude

$$\Delta_{\mathcal{L}}((c, \delta_c) : S, q) \leq |\mathcal{C}_q(c)| A_q(c, \delta_c) \tag{23}$$

By the theorem's assumption we have that

$$\begin{aligned}|\{c_\ominus \in C_q^- : \gamma + \mathrm{d}_\theta(G_q, \overline{G}_c) - \mathrm{d}_\theta(G_q, \overline{G}_{c_\ominus}) \geq 0\}| \geq \beta_0 |C_q^-| \quad \text{if } c \in C_q^+ \\ |\{c_\oplus \in C_q^+ : \gamma + \mathrm{d}_\theta(G_q, \overline{G}_{c_\oplus}) - \mathrm{d}_\theta(G_q, \overline{G}_c) \geq 0\}| \geq \beta_0 |C_q^+| \quad \text{if } c \in C_q^-\end{aligned} \tag{24}$$

Then

$$\Delta_{\mathcal{L}}((c, \delta_c) : S, q) \geq \beta_0 |\mathcal{C}_q(c)| A_q(c, \delta_c) \tag{25}$$

Eq.(3) gives us that $\mathcal{L}(\bullet, Q) = \sum_{q \in Q} \mathcal{L}(\bullet, q)$. This allows us to obtain the following upper bound:

$$\begin{aligned}\beta_0 \sum_{q \in Q} |\mathcal{C}_q(c)| A_q(c, \delta_c) &\leq \mathcal{L}(S \cup \{c\}, \{\delta_w : w \in S\} \cup \{\delta_c\}, Q) - \mathcal{L}(S, \{\delta_w : w \in S\}, Q) \\ &\leq \sum_{q \in Q} |\mathcal{C}_q(c)| A_q(c, \delta_c)\end{aligned} \tag{26}$$

Now, we attempt to get a lower bound on $\Delta_F(c : S, Q)$.

$$\begin{aligned}F(S \cup \{c\}, Q) &\geq \mathcal{L}(S \cup \{c\}, \{\delta_w(S) : w \in S\} \cup \{\delta_c\}, Q) \\ &\geq \mathcal{L}(S, \{\delta_w(S) : w \in S\}, Q) + \beta_0 \sum_{q \in Q} |\mathcal{C}_q(c)| A_q(c, \delta_c) \\ &= F(S, Q) + \beta_0 \sum_{q \in Q} |\mathcal{C}_q(c)| A_q(c, \delta_c)\end{aligned} \tag{27}$$

Maximizing the RHS over $\delta_c \in \Gamma_c$ gives us

$$\Delta_F(c : S, Q) \geq \beta_0 \max_{\delta_c \in \Gamma_c} \sum_{q \in Q} |\mathcal{C}_q(c)| A_q(c, \delta_c) \tag{28}$$

Similarly, we apply the same set of operations on T: considering the set of perturbations $\{\delta_w(T \cup \{c\}) : w \in T \cup \{c\}\}$, consisting of non-critical node-pairs, for $T \cup \{c\}$. Accordingly,

$$F(T, Q) \geq \mathcal{L}(T, \{\delta_w(T \cup \{c\}) : w \in T\}, Q) \tag{29}$$

Using Eq.(26) on the set T, we get,

$$\begin{aligned} F(T \cup \{c\}, Q) &= \mathcal{L}(T \cup \{c\}, \{\delta_w(T \cup \{c\}) : w \in T\} \cup \{\delta_c(T \cup \{c\})\}, Q) \\ &\leq \mathcal{L}(T, \{\delta_w(T \cup \{c\}) : w \in T\}, Q) + \sum_{q \in Q} |\mathcal{C}_q(c)| A_q(c, \delta_c(T \cup \{c\})) \\ &\leq F(T, Q) + \max_{\delta_c \in \Gamma_c} \sum_{q \in Q} |\mathcal{C}_q(c)| A_q(c, \delta_c) \end{aligned} \tag{30}$$

Thus, we obtain,

$$\Delta_F(c : T, Q) \leq \max_{\delta_c \in \Gamma_c} \sum_{q \in Q} |\mathcal{C}_q(c)| A_q(c, \delta_c) \tag{31}$$

Combining Eq.(28) and (31), we get,

$$\Delta_F(c : S, Q) \geq \beta_0 \max_{\delta_c \in \Gamma_c} \sum_{q \in Q} |\mathcal{C}_q(c)| A_q(c, \delta_c) \geq \beta_0 \Delta_F(c : T, Q) \tag{32}$$

Since, $\beta_0 \in (0, 1]$, and $S \subseteq T \subseteq C, c \notin T$ were chosen arbitrarily, $F(\bullet, Q)$ is α-submodular with $\alpha \geq \beta_0$. □

E.4 Proof of Theorem 2

Theorem 2. Given a corpus $\mathcal{G} = \{G_c : c \in C\}$ and a query G_q with $q \in Q$, let $\mathcal{L}$ be the set function defined in Eq. (3). Then, under mild assumptions, there $\exists \beta_1, \beta_2 > 0$, such that for every perturbation $\delta_c \in \Gamma_{qc}$ defined in Eq. (2), we have: $\beta_1 A_q(c, \delta_c) \leq \Delta_{\mathcal{L}}((c, \delta_c) : S, q) \leq \beta_2 A_q(c, \delta_c)$.

First, we establish two results that make the rest of the proof easier to resolve.

Lemma 4 (Shift due to perturbations)**.** By Assumption 2, fix a query $G_q : q \in Q$. Consider corpus items $c \in C, j \in \mathcal{C}_q(c)$. Then, replacing G_c by $G_c \oplus \delta_c$ for some $\delta_c \in \Gamma_{qc}$, increments $x_j(S)$ by $A_q(c, \delta_c)$.

Proof of Lemma 4. Suppose that $y(G_q, G_c) = +1$. Then, $\mathrm{d}_\theta(G_q, G_j)$ is unchanged when G_c is perturbed. So,

$$x_j(S) = \gamma + \mathrm{d}_\theta(G_q, \overline{G}_c) - \mathrm{d}_\theta(G_q, \overline{G}_j). \tag{33}$$

Replacing $G_k \leftrightarrow G_k \oplus \delta_k$, changes $x_j(S) \leftrightarrow x_j(S \cup \{k\})$,

$$\begin{aligned} x_j(S \cup \{k\}) - x_j(S) &= \mathrm{d}_\theta(G_q, G_k \oplus \delta_k) - \mathrm{d}_\theta(G_q, G_k) \\ &= y(G_q, G_k)\Big(\mathrm{d}_\theta(G_q, G_k \oplus \delta_k) - \mathrm{d}_\theta(G_q, G_k)\Big) \\ &= A_q(k, \delta_k) \end{aligned} \tag{34}$$

Eq. (34) holds similarly for $y(G_q, G_c) = -1$, hence establishing the result. □

Lemma 5 (Marginal gain in terms of deficits.)**.** By Assumptions 1 and 2, for every $S \subseteq C$ with $c \notin S$, every assignment $\{\delta_{c'} : c' \in S\}$ and every $\delta_c \in \Gamma_{qc}$,

$$\Delta_{\mathcal{L}}((c, \delta_c) : S, q) = \sum_{j \in \mathcal{C}_q(c)} [A_q(c, \delta_c) - D_j(S)]_+ = |\mathcal{C}_q(c)| A_q(c, \delta_c) - \sum_{j \in \mathcal{C}_q(c)} \min\{A_q(c, \delta_c), D_j(S)\} \tag{35}$$

Furthermore,

$$D_j(S) = [D_j(\phi) - \mathbf{1}_{\{j \in S\}} A_q(j, \delta_j)]_+ \leq D_j(\phi) \qquad \text{for every } j \in \mathcal{C}_q(c), \tag{36}$$

and consequently, $\Delta_{\mathcal{L}}((c, \delta_c) : S, q) \geq \Delta_{\mathcal{L}}((c, \delta_c) : \phi, q) \geq 0$.

Proof of Lemma 5. Notice that, for $u \in \mathbb{R}, s \geq 0$,

$$0 \leq [u + s]_+ - [u]_+ = [s - [-u]_+]_+ \leq s \tag{37}$$

By Lemma 4, the expression $\Delta_{\mathcal{L}}((c, \delta_c) : S, q)$ will only contain terms containing c, and will shift by $A_q(c, \delta_c)$.

$$\Delta_{\mathcal{L}}((c, \delta_c) : S, q) = \sum_{j \in \mathcal{C}_q(c)} \Big([x_j(S) + A_q(c, \delta_c)]_+ - [x_j(S)]_+ \Big) \tag{38}$$

Using Eq. (37) and Assumption 1,

$$[x_j(S) + A_q(c, \delta_c)]_+ - [x_j(S)]_+ = [A_q(c, \delta_c) - D_j(S)]_+ = A_q(c, \delta_c) - \min\{A_q(c, \delta_c), D_j(S)\} \tag{39}$$

Eq (37) asserts that each summand ≥ 0. Then, substituting into Eq (38), we get,

$$\Delta_{\mathcal{L}}((c, \delta_c) : S, q) = \sum_{j \in \mathcal{C}_q(c)} [A_q(c, \delta_c) - D_j(S)]_+ = |\mathcal{C}_q(c)| A_q(c, \delta_c) - \sum_{j \in \mathcal{C}_q(c)} \min\{A_q(c, \delta_c), D_j(S)\}$$

as needed.

By Assumption 2, $x_j(S) = x_j(\phi) + \mathbf{1}_{\{j \in S\}} A_q(j, \delta_j)$. Using Assumption 1, we have,

$$\begin{aligned} D_j(S) &= \big[-x_j(\phi) - \mathbf{1}_{\{j \in S\}} A_q(j, \delta_j) \big]_+ \\ &= \Big[[-x_j(\phi)]_+ - \mathbf{1}_{\{j \in S\}} A_q(j, \delta_j) \Big]_+ \\ &= \big[D_j(\phi) - \mathbf{1}_{\{j \in S\}} A_q(j, \delta_j) \big]_+ \\ D_j(S) &\leq D_j(\phi) \end{aligned} \tag{40}$$

From the above, it follows that $\Delta_{\mathcal{L}}((c, \delta_c) : S, q) \geq \Delta_{\mathcal{L}}((c, \delta_c) : \phi, q) \geq 0$. □

Now, we show the proof for Theorem 2 using the above results.

Proof. Fix $q \in Q, c \in C, S \subseteq C : c \notin S$, and its corresponding perturbations set $\{\delta_{c'} : c' \in S\}$, and $\delta_c \in \Gamma_{qc}$.

If $A_q(c, \delta_c) = 0$, then $\Delta_{\mathcal{L}}((c, \delta_c) : S, q) = 0$, so the bounds hold. We now consider the case of $A_q(c, \delta_c) > 0$.

Upper bound. Each summand satisfies $\min\{A_q(c, \delta_c), D_j(S)\} \geq 0$, so

$$\Delta_{\mathcal{L}}((c, \delta_c) : S, q) \leq |\mathcal{C}_q(c)| A_q(c, \delta_c) \tag{41}$$

Lower bound. We will use the fact that $\Delta_{\mathcal{L}}((c, \delta_c) : S, q) \geq \Delta_{\mathcal{L}}((c, \delta_c) : \phi, q) \geq 0$ from Lemma 5. Using Assumption 3, we define,

$$N := \{j \in \mathcal{C}_q(c) : D_j(\phi) \leq \tau\}, \qquad \text{so that } |N| \geq \beta_0 |\mathcal{C}_q(c)| \tag{42}$$

We evaluate Eq.(35) at $S = \phi$. Using the monotonicity of $t \mapsto \min\{A_q(c, \delta_c), t\}$ with $D_j(\phi) \leq \tau$ on N and Assumption 3,

$$\Delta_{\mathcal{L}}((c, \delta_c) : \phi, q) \geq \sum_{j \in N} (A_q(c, \delta_c) - \min\{A_q(c, \delta_c), \tau\}) = |N|(A_q(c, \delta_c) - \tau) \tag{43}$$

Since, $\kappa > 1$, we also get $\kappa\tau \leq A_q(c, \delta_c)$,

$$A_q(c, \delta_c) - \tau \geq (1 - \frac{1}{\kappa}) A_q(c, \delta_c) \geq 0 \tag{44}$$

Combining Eq. (43) and Eq. (44), we get,

$$\Delta_{\mathcal{L}}((c, \delta_c) : S, q) \geq \Delta_{\mathcal{L}}((c, \delta_c) : \phi, q) \geq \beta_0 (1 - \frac{1}{\kappa}) |\mathcal{C}_q(c)| A_q(c, \delta_c) = \beta |\mathcal{C}_q(c)| A_q(c, \delta_c) \tag{45}$$

Now, setting $\beta_1 = \beta_0(1 - \frac{1}{\kappa})|\mathcal{C}_q(c)|$, and $\beta_2 = |\mathcal{C}_q(c)|$, we can combine equations (45) and (41) to get,

$$\beta_1 A_q(c, \delta_c) \leq \Delta_{\mathcal{L}}((c, \delta_c) : S, q) \leq \beta_2 A_q(c, \delta_c) \tag{46}$$

Since the corpus item $c \in C$, the subset $S \subseteq C$ and the query $q \in Q$ were chosen arbitrarily, the result shows that such β_1, β_2 exist always, which was the needed result. □

E.5 Proof of Proposition 3

To prove Proposition 3, we use an intermediate results given as Lemma 6. Both Proposition 3 and Lemma 6 and their proofs are obtained from Nair et al. (2026) and reproduced/quoted here to make

the analysis self contained. Proposition 3 has been adapted from and is slightly different from Nair et al. (2026) which takes sum over query tokens in text, whereas we sum over dimensions.

Lemma 6. Let a, b be arbitrary vectors, and let $\boldsymbol{\omega} \sim \mathcal{N}(0, \boldsymbol{I})$, where $\boldsymbol{I}$ is identity matrix of compatible dimension. Define the random feature map $\Phi_{\boldsymbol{\omega}}$ as given in Eq. (8). Then we have,

$$\Phi_{\boldsymbol{\omega}}(a)^\top \Phi_{\boldsymbol{\omega}}(b) \;\le\; [a^\top b]_+ \text{ always, and } \Phi_{\boldsymbol{\omega}}(a)^\top \Phi_{\boldsymbol{\omega}}(b) \;=\; [a^\top b]_+ \text{ w. p. } \ge \tfrac{1}{2}. \tag{47}$$

Proof. Based on the definition of $\Phi_{\boldsymbol{\omega}}$, it expands as

$$s := \Phi_{\boldsymbol{\omega}}(a)^\top \Phi_{\boldsymbol{\omega}}(b) \tag{48}$$

$$= \frac{1}{2}\Big(a^\top b + \operatorname{sign}(\boldsymbol{\omega}^\top a)\operatorname{sign}(\boldsymbol{\omega}^\top b)\, a^\top b\Big) \tag{49}$$

$$= \frac{1}{2}\Big(1 + \operatorname{sign}(\boldsymbol{\omega}^\top a)\operatorname{sign}(\boldsymbol{\omega}^\top b)\Big)\, a^\top b \tag{50}$$

Since $\operatorname{sign}(\boldsymbol{\omega}^\top a)\cdot\operatorname{sign}(\boldsymbol{\omega}^\top b) \in \{-1, +1\}$, s takes only two possible values: $s = a^\top b$ when the two hyperplane signs agree, and $s = 0$ otherwise. In particular, s never exceeds $[a^\top b]_+$, which gives the first claim.

For the probability bound, we use the following identity (Charikar, 2002) for $\boldsymbol{\omega}$ drawn from a spherically symmetric distribution,

$$\Pr\left[\operatorname{sign}(\boldsymbol{\omega}^\top a) = \operatorname{sign}(\boldsymbol{\omega}^\top b)\right] = 1 - \frac{\theta}{\pi}, \qquad \theta := \cos^{-1}\left(\frac{a^\top b}{\|a\|\,\|b\|}\right). \tag{51}$$

We split on the sign of $a^\top b$.

Case $a^\top b > 0$ (so $\theta < \pi/2$). Here $[a^\top b]_+ = a^\top b$, and s equals this value exactly when the two signs agree:

$$\Pr\left[s = [a^\top b]_+\right] = 1 - \frac{\theta}{\pi} \;>\; 1 - \frac{1}{2} = \frac{1}{2}. \tag{52}$$

Case $a^\top b < 0$ (so $\theta > \pi/2$). Here $[a^\top b]_+ = 0$, and s equals this value exactly when the two signs *disagree*:

$$\Pr\left[s = [a^\top b]_+\right] = \Pr\left[\operatorname{sign}(\boldsymbol{\omega}^\top a) \neq \operatorname{sign}(\boldsymbol{\omega}^\top b)\right] = \frac{\theta}{\pi} \;>\; \frac{1}{2}. \tag{53}$$

Case $a^\top b = 0$**.** Here $[a^\top b]_+ = 0 = s$ regardless of the realized signs, so the equality holds with probability 1.

In every case $\Pr[s = [a^\top b]_+] \ge 1/2$, proving the claim.

□

Proposition 3. Given R random hyperplanes $\boldsymbol{\omega}_r$ drawn i.i.d. from $\boldsymbol{\omega}_r \sim \mathcal{N}(0, \boldsymbol{I}_2)$ and the relevance distance $\mathrm{d}_\theta(G_q, G_c)$ computed in Eq. (7), let $\Phi_{\boldsymbol{\omega}}$ be the feature map defined in Eq. (8). Then, with probability at least $1 - \dim_h / 2^R$, we have:

$$\mathrm{d}_\theta(G_q, G_c) = \sum_{j \in [\dim_h]} \max_{r \in [R]} \Phi_{\boldsymbol{\omega}_r}(\boldsymbol{a}_j(G_q, \theta))^\top \Phi_{\boldsymbol{\omega}_r}(\boldsymbol{b}_j(G_c, \theta)) \tag{54}$$

Proof. Based on Eq.(7), the relevance distance decomposes across the $\dim_h$ embedding coordinates as

$$\mathrm{d}_\theta(G_q, G_c) = \sum_{j \in [\dim_h]} \left[\boldsymbol{a}_j(G_q, \theta)^\top \boldsymbol{b}_j(G_c, \theta)\right]_+ \tag{55}$$

Fix a coordinate $j \in [\dim_h]$ and abbreviate $\boldsymbol{a}_j := \boldsymbol{a}_j(G_q, \theta)$, $\boldsymbol{b}_j := \boldsymbol{b}_j(G_c, \theta)$.

By Lemma 6, for each hyperplane $\boldsymbol{\omega}_r \sim \mathcal{N}(0, \boldsymbol{I}_2)$.

$$\Phi_{\boldsymbol{\omega}_r}(\boldsymbol{a}_j)^\top \Phi_{\boldsymbol{\omega}_r}(\boldsymbol{b}_j) \;\le\; [\boldsymbol{a}_j^\top \boldsymbol{b}_j]_+, \qquad \Pr\left[\Phi_{\boldsymbol{\omega}_r}(\boldsymbol{a}_j)^\top \Phi_{\boldsymbol{\omega}_r}(\boldsymbol{b}_j) = [\boldsymbol{a}_j^\top \boldsymbol{b}_j]_+\right] \;\ge\; \frac{1}{2}. \tag{56}$$

Since $\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_R$ are drawn i.i.d., these R events are mutually independent for the fixed pair $(\boldsymbol{a}_j, \boldsymbol{b}_j)$. Because no individidual dot product term can exceed $[\boldsymbol{a}_j^\top \boldsymbol{b}_j]_+$, the maximum over $r \in [R]$

becomes different from $[\boldsymbol{a}_j^\top \boldsymbol{b}_j]_+$ only if *every* trial fails, which happens with probability at most $(1/2)^R$.

Applying a union bound over all $\dim_h$ coordinates,

$$\Pr\left(\max_{r\in[R]} \Phi_{\boldsymbol{\omega}_r}(\boldsymbol{a}_j)^\top \Phi_{\boldsymbol{\omega}_r}(\boldsymbol{b}_j) \;\neq\; [\boldsymbol{a}_j^\top \boldsymbol{b}_j]_+ \;\text{ for some } j \in [\dim_h]\right) \;\leq \sum_{j\in[\dim_h]} \Pr[E_j] \;\leq\; \dim_h \cdot 2^{-R} \tag{57}$$

The complementary event occurs with probability $1 - \dim_h /2^R$, which immediately gives the required result.

□

F Details about experimental setup

In this section, we present the details of our experimental setup which includes a discussion of dataset statistics, victim and substitute model architectures, details of the metrics we used and the process of adaptation of baselines to our setting.

F.1 Dataset details

We use five molecular graph datasets from the TUdatasets (Morris et al., 2020) repository for our experiments PTC-FM, PTC-FR, PTC-MM, PTC-MR and AIDS. For each dataset, we have a corpus of size 800 graphs and queries are split into train, validation and test splits in the ratio $60:15:25$.

	$\lvert Q\rvert$	Mean $\lvert V_q\rvert$	Min $\lvert V_q\rvert$	Max $\lvert V_q\rvert$	Mean $\lvert E_q\rvert$
AIDS	300	11.51	7	15	11.38
FM	300	12.14	6	15	11.93
FR	300	12.36	6	15	12.19
MM	300	11.99	6	15	11.75
MR	300	12.34	6	15	12.08

Table 7: Query graph statistics: We report the average, minimum and max node count *Mean* $|V_q|$, *Min* $|V_q|$, and *Max* $|V_q|$, respectively. Also, we report average edges across all the queries *Mean* $|E_q|$.

In Table 7, we report the statistics of the query graphs averaged across all the splits.

$\mathcal{G} = \{G_c : c \in C\}$					
	$\lvert C\rvert$	Mean $\lvert V_c\rvert$	Min $\lvert V_c\rvert$	Max $\lvert V_c\rvert$	Mean $\lvert E_c\rvert$
AIDS	800	20.77	17	25	21.45
FM	800	19.59	17	25	20.38
FR	800	19.89	17	25	20.79
MM	800	19.70	17	25	20.47
MR	800	19.83	17	25	20.62

$\{G_c \oplus \delta_c : \delta_c \in \Pi_c, \forall c \in C\}$					
	$\lvert C\rvert$	Mean $\lvert V_c\rvert$	Min $\lvert V_c\rvert$	Max $\lvert V_c\rvert$	Mean $\lvert E_c\rvert$
AIDS	167483	21.38	17	25	22.90
FM	148521	20.15	17	25	21.77
FR	153148	20.43	17	25	22.15
MM	150408	20.29	17	25	21.87
MR	152388	20.42	17	25	22.03

Table 8: Corpus graph statistics: The left table reports the graph statistics for the dataset corpus having 800 graphs. The right table reports the statistics of the corpus $\{G_c \oplus \delta_c : \delta_c \in \Pi_c, \forall c \in C\}$ used for ANN indexing.

Table 8 (left) reports the clean corpus statistics containing 800 graphs. For QuPAC indexing, we index for each pair (c, δ_c) the perturbed corpus graph $G_c \oplus \delta_c$. Table 8 (right) presents the graph statistics for the perturbed corpus $\{G_c \oplus \delta_c : \delta_c \in \Pi_c, \forall c \in C\}$.

	pairs(1)	pairs(0)	$\frac{\text{pairs(1)}}{\text{pairs(0)}}$
AIDS	44507	195493	0.2277
FM	42790	197210	0.2170
FR	41208	198792	0.2073
MM	42305	197695	0.2140
MR	42063	197937	0.2125

	pairs(1)	pairs(0)	$\frac{\text{pairs(1)}}{\text{pairs(0)}}$
AIDS	15709693	34535207	0.4549
FM	13612083	30944217	0.4399
FR	13500789	32443611	0.4161
MM	14493571	30628829	0.4732
MR	14297944	31418456	0.4551

Table 9: Relevance statistics: corpus relevance statistics (left) and perturbed corpus relevance statistics (right)

In Table 9 (left) we report the statistics $y(G_q, G_c)$ for the clean corpus. There are a total of 240,000 query-corpus graph pairs for the clean corpus. Table 9 (right) reports the relevance statistics $y(G_q, G_c \oplus \delta_c)$ for the perturbed corpus. We see that the number of query-corpus graph pairs is more than 40 million.

F.2 Victim and substitute NSR retrievers

GEN (Li et al., 2019). GEN uses a GNN encoder to generates single vector graph representations of input graphs. Given query corpus graph pairs, GEN generates the embeddings $\boldsymbol{h}^{(q)}$ and $\boldsymbol{h}^{(c)}$ independently (late interaction) after R propagation steps. These embeddings are used to compute the relevance distance between G_q and G_c as the hinge distance $d_\theta(G_q, G_c) = \|[\boldsymbol{h}^{(q)} - \boldsymbol{h}^{(c)}]_+\|_1$.

GMN (Li et al., 2019). GMN is an early interaction variant of GEN. GMN performs cross graph interaction between the query and corpus graphs through attention mechanism between $\boldsymbol{h}_r^{(q)}$ and $\boldsymbol{h}_r^{(c)}$ at each propagation step $r \in [R]$. The final relevance distance is identical to GEN, i.e. $d_\theta(G_q, G_c) = \|[\boldsymbol{h}^{(q)} - \boldsymbol{h}^{(c)}]_+\|_1$, where $\boldsymbol{h}^{(q)}$ and $\boldsymbol{h}^{(c)}$ are GNN embeddings after the R propagation steps.

IsoNet(Roy et al., 2022). IsoNet series of models are late interaction multi-vector graph embedders. The IsoNet (Node) variant performs GNN message passing to generate node representations for each node in the graph. The query and corpus node embeddings, $\boldsymbol{H}^{(q)}$ and $\boldsymbol{H}^{(c)}$, after M rounds of message passing are passed through a gumbel-sinkhorn network to ex-

tract a soft node alignment matrix R which is used to compute the hinge based relevance distance $d_\theta(G_q, G_c) = \|[\boldsymbol{H}^{(q)} - \boldsymbol{R}\boldsymbol{H}^{(c)}]_+\|_1$. The IsoNet (Edge) variant differs from IsoNet (Node) in that operates on the edge embeddings as opposed to node embeddings. After GNN propagation, the edge embeddings form the multi-vector graph embeddings. The edge based representation $\boldsymbol{X}^{(q)}$ and $\boldsymbol{X}^{(c)}$ are passed through a gumbel-sinkhorn layer to extract the edge alignment matrix $\boldsymbol{R}$. The relevance distance between query-corpus graphs is computed as $d_\theta(G_q, G_c) = \|[\boldsymbol{X}^{(q)} - \boldsymbol{R}\boldsymbol{X}^{(c)}]_+\|_1$.

IsoNet++ (Ramachandran et al., 2024). IsoNet++ is an early-interaction extension of IsoNet. IsoNet++ (Node) performs multiple rounds of GNN propagation. After round t, the graph representations are passed through gumbel-sinkhorn aligner to yield the node alignment $\boldsymbol{R}(t)$. The GNN propagation in subsequent round are contextualized through the node alignment $\boldsymbol{R}(t)$. After T rounds, the query and corpus embeddings ($\boldsymbol{H}^{(q)}(T)$ and $\boldsymbol{H}^{(c)}(T)$) are used to compute the relevance distance $d_\theta(G_q, G_c) = \|[\boldsymbol{H}^{(q)}(T) - \boldsymbol{R}(T)\boldsymbol{H}^{(c)}(T)]_+\|_1$. IsoNet++ (Edge) operates on edge embeddings. After round t, the graph query and corpus graph representations, $\boldsymbol{X}^{(q)}(t)$ and $\boldsymbol{X}^{(c)}(t)$ are passed through the Gumbel-Sinkhorn network to yield edge alignment matrix $\boldsymbol{R}(t)$ and the GNN propagations in the subsequent round $(t+1)$ are contextualized using $\boldsymbol{R}(t)$. After T rounds, the relevance distance is computed as $d_\theta(G_q, G_c) = \|[\boldsymbol{X}^{(q)}(T) - \boldsymbol{R}(T)\boldsymbol{X}^{(c)}(T)]_+\|_1$.

F.3 Adaptation of baselines

Nettack (Zügner et al., 2018). Nettack is an adversarial attack strategy for targeted node-classification. The method uses a GCN surrogate model for gradient-free score prediction to compute the attack. We adapt Nettack to our setting as follows. The method computes impact score for each node pair $(u, v) \in \Pi_c$ and then computes the per-node score as the max impact score over all incident non-critical edges. The top-2 highest impact nodes are given label 1 and rest are given label 0. Next, a surrogate 2-layer GCN is trained across whole corpus to predict the impact node labels. We keep the surrogate architecture is kept same as that in the original implementation. At inference time, Nettack uses the surrogate to predict the impact score of the nodes. The top-2 nodes ranked by impact score are chosen as the attacked node-pair.

GMA (Zhang et al., 2020b). GMA is an attack model that uses a trained MLP to learn a weighted distribution over all candidate edges in a corpus graph G_c, and propose the worst-case edge perturbation δ_c for a (G_q, G_c) pair. The method runs a K-step projected gradient descent (PGD): at each step, an edge index δ_w is sampled, and the substitute model's score is computed on $(G_q, G_w \oplus \delta_w)$. The reward for this perturbation is computed as

$$(\mathrm{d}_\theta(G_q, G_w \oplus \delta_w) - \mathrm{d}_\theta(G_q, G_w))(2y(G_q, G_w) - 1) \tag{58}$$

Via a REINFORCE step, the weighted distribution is updated based on the reward. The trained MLP, called a Starting-Point Network (SPN) learns a warm-start for the K-step PGD. At inference time, the SPN is frozen, and it outputs a logit vector corresponding to the predicted worst-case edge perturbation. This is then refined via the K-step PGD. For our budget-constrained, query workload-based setting, we made the following adaptations to the attack method: **(1)** At each step of the PGD, the rewards are aggregated over the entire query workload Q, and an averaged reward (over Q) is used in the REINFORCE step. **(2)** Instead of just the top-scoring edge for each corpus graph, the attack is run for each corpus graph. Then, the B perturbations corresponding to the top-B rewards are selected across all the corpus graphs.

RL-Attack (Dai et al., 2018). RL-Attack trains a DQN on a Q-learning policy over graph embeddings to select the worst-case edge perturbation for any graph. It further optimizes the $\mathcal{O}(N^2)$ search space over all edges by independently scoring each node as a potential start/end point for a perturbed edge, and selects the maximally scored node-pair over two passes in each episode. A trained mean-field GNN encoder is used to generate the graph embeddings. Candidate node-pairs that shift the victim model's score favourably are assigned a Q-reward of $+1$, and -1 otherwise. We make the following adaptations to this baseline to use it for a budget-constrained attack on a fixed corpus, with access to a query workload Q: **(1)** To make the choice of worst-case node-pair perturbation conditioned on the query workload, a representative query $G_q : q \in Q$ is chosen. The DQN's ε-greedy exploration is then carried out on the concatenated embedding $[h_v||h_{G_c}||h_{G_q}]$, where, h_v is the node embedding for the candidate node, and, h_{G_q}, h_{G_c} are the graph embeddings of the query and corpus graphs via the mean-field GNN encoder. **(2)** Once the worst-case edge perturbations are chosen for each corpus graph, the corpus graphs are then ranked by distance shifts induced by their respective worst-case perturbations, and the top-B graphs are chosen.

IAP (Zhong et al., 2023). IAP is a graph injection-based adversarial attack that generates adversarial corpus items to degrade the retrieval quality of the victim model. It assumes that queries are clustered in the encoder's query-embedding space, so it is possible to greedily attack larger clusters to maximize the effect of adding an adversarial corpus item by measuring its effect across multiple queries. The original method is built for text corpora, and we made the following adaptations to extend its setting to our budget-constrained attack on graphs. **(1)** The WL-hash of the query graphs is used for constructing the feature vectors of the query graphs, and then they are clustered using standard K-Means. **(2)** The perturbation budget B is defined over the number of edges that can be exhausted in building adversarial graphs over all the query clusters. **(3)** One adversarial graph is constructed per-cluster, by greedily iterating over the clusters.

AGGD (Su et al., 2025). AGGD also employs a corpus injection-based attack strategy. Here, the worst-case edge perturbation is chosen by the node-pair flip that causes the maximal gradient shift in the corpus embedding. The Top-K such candidate node-pairs are selected, and then the perturbed corpus items are explicitly constructed, and the node-pair that degrades the ranking quality the most is chosen. To adapt this baseline to our query workload-restricted setting, the model scoring for worst-case edge perturbation is restricted to the belief model, and the query workload is made visible to the attack. The attack iterates over each corpus item, selecting the worst-case edge perturbation for each until the budget B has been exhausted.

F.4 Evaluation Protocol

Adjustment of ground truth relevance labels. The adversary computes $(S, \{\delta_c\})$ based on the historic query workload Q, and applies the same set of perturbations $\{\delta_c\}$ to the corpus subset S for all the unseen queries Q^*. The computed $\delta_j \in \{\delta_c\}$ are chosen such that $\delta_j \in \cap_{q \in Q} \Gamma_{qj}$, that is, relevance labels are preserved for perturbed corpus items. However, since Q^* is not accessible during the computation of $(S, \{\delta_c\})$, we expect to see this label preservation to be violated for some unseen queries, that is, $\exists q \in Q^*, \delta_j \notin \Gamma_{qj}$. To ensure that the victim NSR's ranking quality is assessed fairly, we enforce that the evaluation be correct with respect to the perturbed corpus. For this purpose, we recompute the true relevance labels for all corpus items. The updated set of relevance labels is computed as $\{y(G_q, \overline{G}_c) : \overline{G}_c \in \mathcal{G}_{S,\{\delta_c : c \in S\}}\}$, and the victim model's ranking quality is evaluated using this set as the ground truth.

Metrics. All our evaluations report mean average precision (MAP), the mean of average precision (AP) across all unseen queries Q^*. Then AP for a query $q \in Q^*$ is computed as

$$AP(q) = \frac{1}{C_q^+} \sum_{k=1}^{N} P(k) \mathbf{1}_{\{\text{rel@}k=1\}}$$

Here C_q^+ is the corpus subset relevant to q, $P(k)$ is precision at rank k, $\mathbf{1}_{\{\bullet\}}$ is the indicator function, and rel@k is the relevance label of item at rank k. Given $AP(q)$, mean average precision is computed as:

$$MAP = \frac{1}{|Q^*|} \sum_{q \in Q^*} AP(q)$$

Evaluation for IAP and AGGD. IAP and AGGD inject items into the corpus, while all other methods edit graphs already present in the corpus. For evaluation of these baselines, we first compute the relevance label of the new corpus graphs with the test queries and then report MAP on the corpus appended with the new graphs.

F.5 Hardware

All our experiments are conducted on NVIDIA A6000, A6000 (Ada) and RTX 6000 Blackwell architecture GPUs.

G ADDITIONAL EXPERIMENTS

In this section we report several additional experiments that extend the results in the main paper. We start with a full sweep of main results on all five datasets across the complete set of victim models in the white box and black box settings. We also show a sweep of results on cross-architecture transfer across different substitute and victim architecture combinations. We also provide time and space complexity of our method.

G.1 EXTENDED WHITE BOX AND BLACK BOX RESULTS

In Figure 1 we show the results of comparison of QUPAC with baselines in the white box (top row) and black box (bottom row) settings for IsoNet++ (Edge) on FM, FR, MM and MR datasets on test workload. We now present the extended results for the same. We compare the adversarial performance of QUPAC with baselines for FM, FR, MM, MR and AIDS datasets across six victim NSR models, i.e. GEN, GMN, IsoNet (Edge), IsoNet (Node), IsoNet++ (Edge) and IsoNet++ (Node). For the test workload, the attack is computed on a 20% subset of the test set and the evaluation is carried out on the 80% held-out set. While for the validation workload, the attack is computed on the val set and the evaluation is carried out on the whole test set.

G.1.1 WHITE BOX AND BLACK BOX RESULTS ON TEST WORKLOAD

White box setting on workload consisting of 20% test queries Figure 10 shows the comparison of QUPAC against baselines on the white box setting for workload consisting of 20% test queries. We observe that **(1)** QUPAC significantly outperforms baselines in all the figures often beating baselines by a huge margin. **(2)** GEN is the most affected victim model by the attack while IsoNet++ (Edge) is the least affected by the adversarial attacks. **(3)** RL-Attack is the most competitive baseline in all cases.

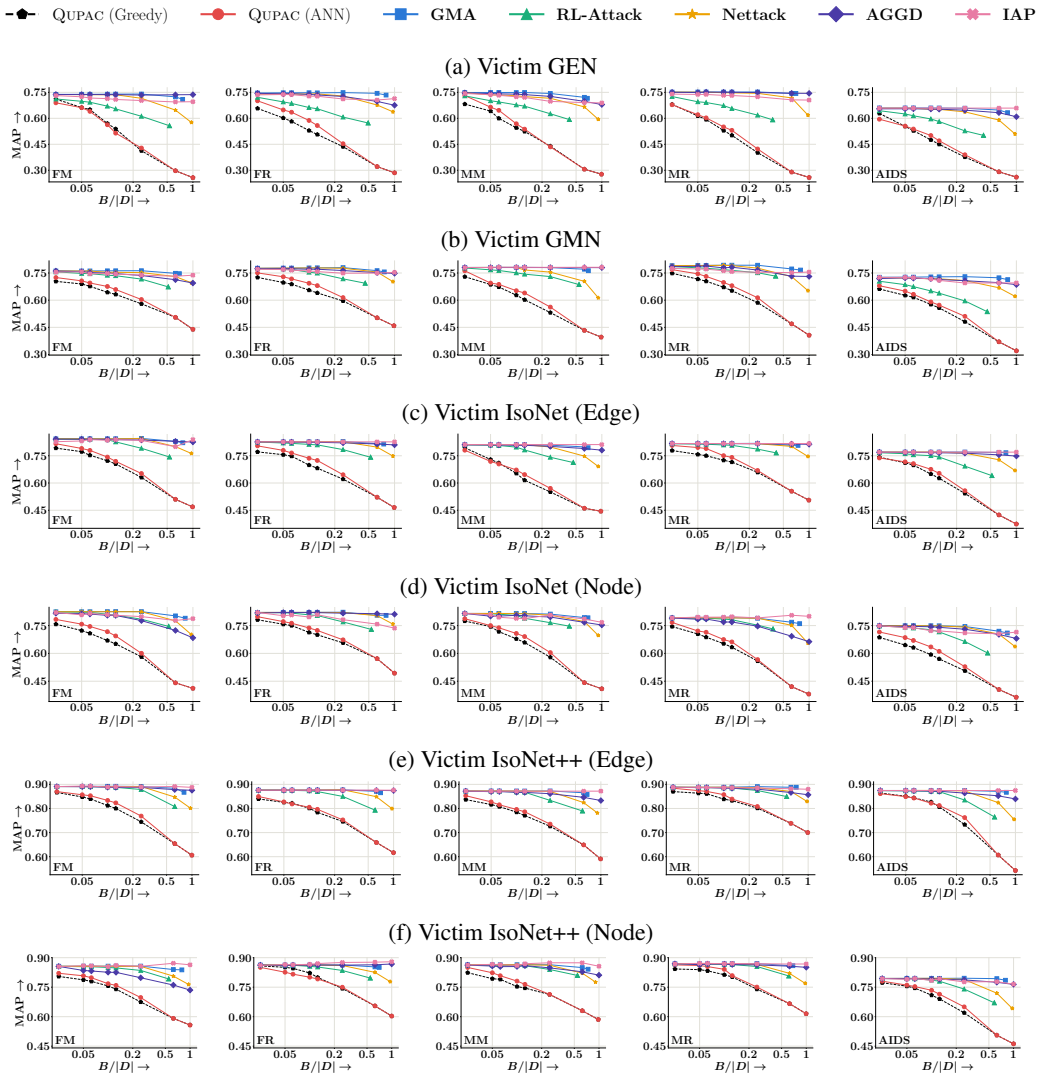


Figure 10: Performance measured in terms of MAP degradation (lower indicates a more successful attack) for QUPAC's variants and all baselines *viz.*, GMA, RL-Attack, Nettack, AGGD, and IAP. We consider white box settings. The past query workload Q consists of 20% of test queries, and we measure performance on the remaining 80%. QUPAC outperforms existing baselines by a significant margin.

Black box setting on workload consisting of 20% test queries In Figure 11 we report the extended comparison of QUPAC with the baselines on the black box setting. The past query workload Q consists of 20% of test queries. We make the following observations: **(1)** QUPAC outperforms baselines in all cases in the black box setting. This shows that our attack effectively transfers under imperfect rank correlation between the substitute and the victim model. **(2)** Consistent with the white box setting, GEN is the most affected victim model by the attack while IsoNet++ (Edge) is the least affected by the adversarial attacks. **(3)** RL-Attack is the most competitive baseline in all cases.

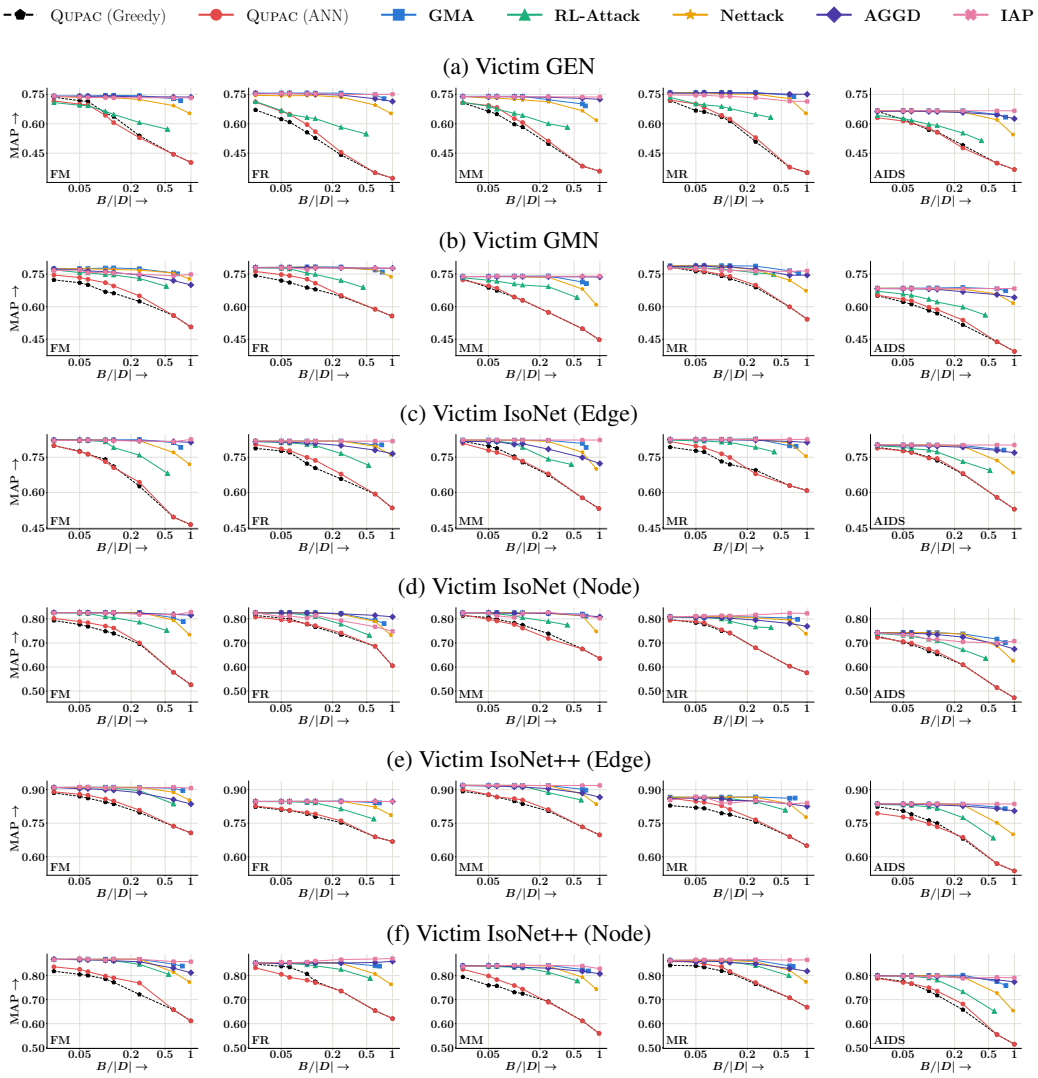


Figure 11: Performance measured in terms of MAP degradation (lower indicates a more successful attack) for QUPAC's variants and all baselines *viz.*, GMA, RL-Attack, Nettack, AGGD, and IAP. We consider black box settings. The past query workload Q consists of 20% of test queries, and we measure performance on the remaining 80%. QUPAC outperforms existing baselines by a significant margin.

G.1.2 WHITE BOX AND BLACK BOX RESULTS ON WORKLOAD CONSISTING OF VALIDATION SET QUERIES

White box setting In Figure 12, we extend the results of Figure 1 on the white box setting. The past query workload Q consists of queries from the validation set. We observe that: **(1)** QUPAC outperforms baselines in all cases when attack is computed on the validation workload. This shows that our method transfers well across different query workloads. **(2)** On the validation workload, we observe that GEN is the most affected victim model by the attack while IsoNet++ (Edge) is the least affected by the adversarial attacks. **(3)** We see that at lower budgets, QUPAC (Greedy) performs marginally better than QUPAC (ANN). **(4)** RL-Attack is the most competitive baseline in all cases.

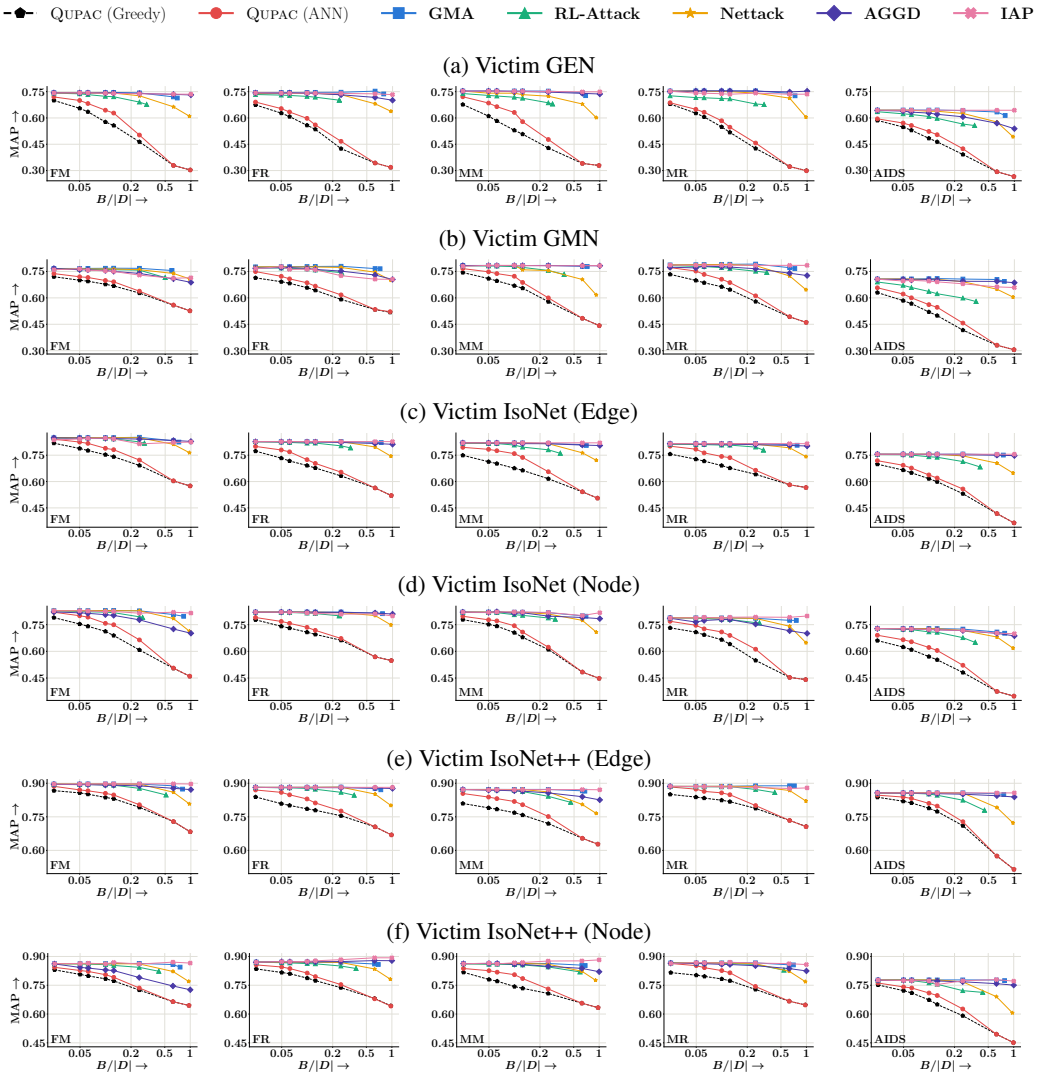


Figure 12: Performance measured in terms of MAP degradation (lower indicates a more successful attack) for QUPAC's variants and all baselines *viz.*, GMA, RL-Attack, Nettack, AGGD, and IAP. We consider white box settings. The past query workload Q consists of queries from the validation set. QUPAC outperforms existing baselines by a significant margin.

Black box settings Figure 13 shows the comparison of QUPAC with baselines in the black box setting. The past query workload Q consists of queries from the validation set. We make the following observations: **(1)** QUPAC outperforms baselines in all cases when attack is computed on the validation workload and evaluated in the black box setting. This shows that our method transfers well across different query workloads under imperfect rank correlation between the substitute and victim models. **(2)** We observe that GEN is the most affected victim model by the attack while IsoNet++ (Edge) is the least affected by the adversarial attacks. **(3)** RL-Attack is the most competitive baseline in all cases.

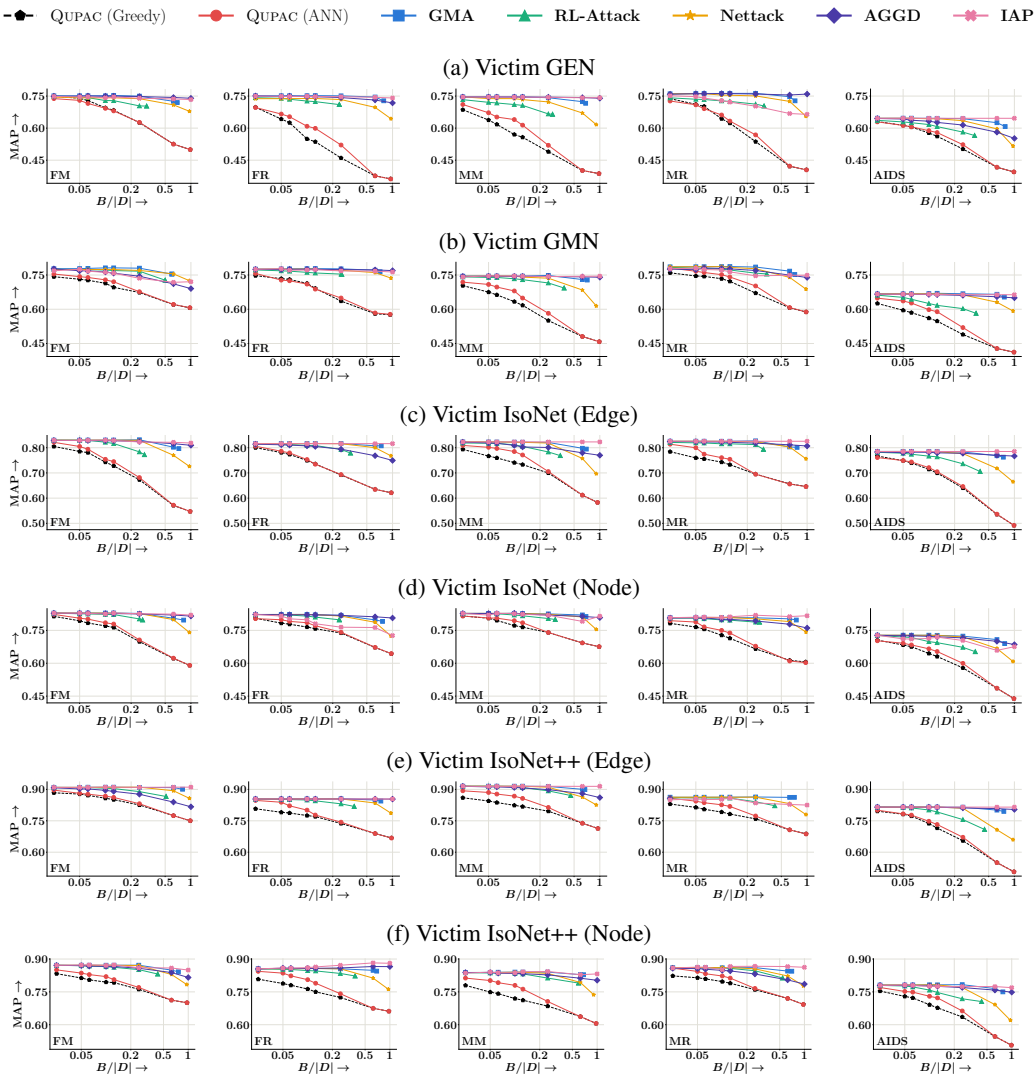


Figure 13: Performance measured in terms of MAP degradation (lower indicates a more successful attack) for QUPAC's variants and all baselines *viz.*, GMA, RL-Attack, Nettack, AGGD, and IAP. We consider black box settings. The past query workload Q consists of queries from the validation set. QUPAC outperforms existing baselines by a significant margin.

G.2 TRANSFER ATTACK: EXTENDED RESULTS

We now present a comprehensive sweep of experiments for transfer attacks. As described in Section 4.1, in transfer attack experiments the substitute model d_θ architecture differs from the victim model d_φ. The adversary computes $(S, \{\delta_c\})$ based on the substitute and the evaluation is performed on the victim model. Figure 14 and 15 extend the results of Figure 2 in the paper on six distinct combinations of d_θ-d_φ architectures, i.e. $\mathrm{d}_\theta, \mathrm{d}_\varphi \in \{$ IsoNet (Edge), GMN, IsoNet++ (Edge) $\}$, for test and validation workloads, respectively.

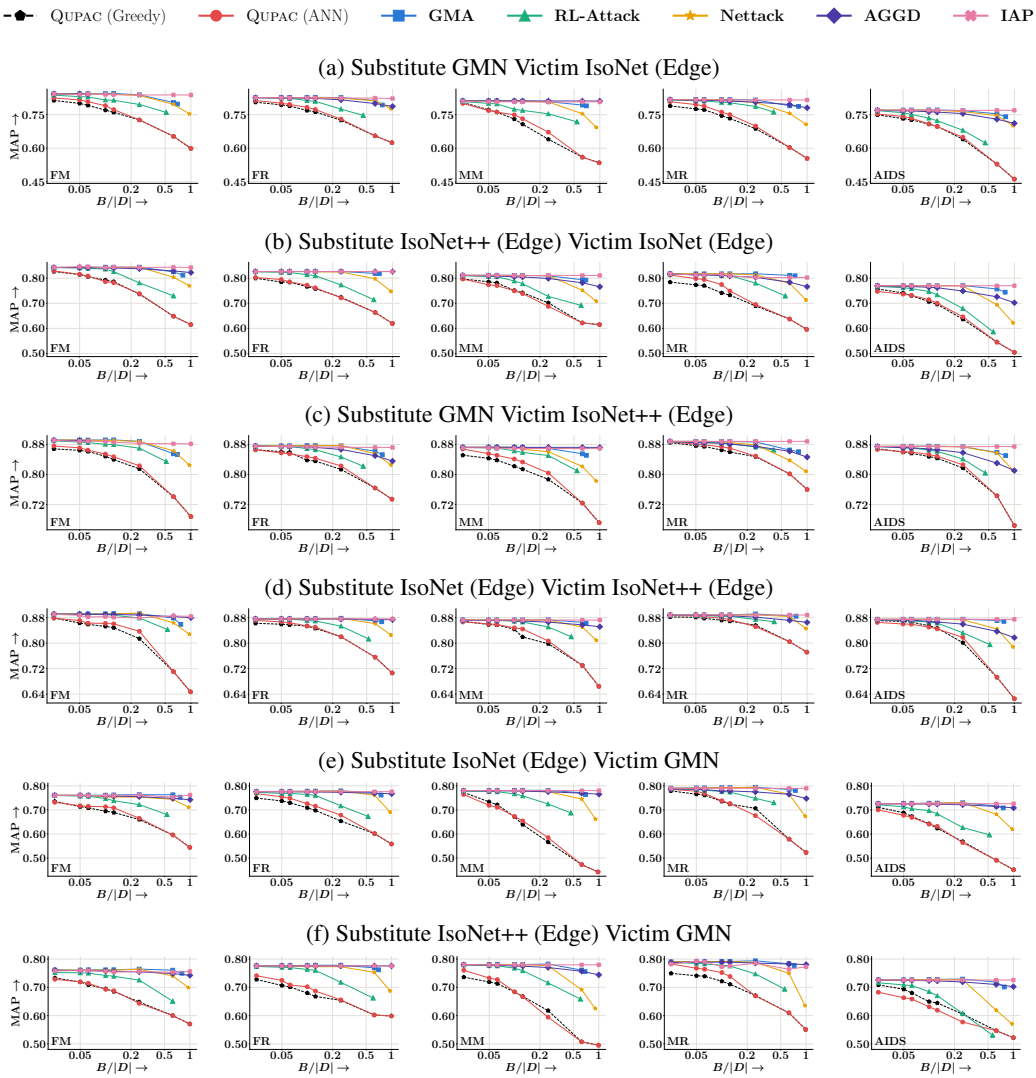


Figure 14: Performance measured in terms of MAP degradation (lower indicates a more successful attack) for QUPAC's variants and all baselines *viz.*, GMA, RL-Attack, Nettack, AGGD, and IAP. We consider transfer attack settings. The past query workload Q consists of 20% of test queries, and we measure performance on the remaining 80%. QUPAC outperforms the baselines by a significant margin.

Transfer attack results on workload consisting of 20% test queries. Figure 14 shows the transfer attack results, where the past query workload Q consists of 20% of test queries. Results are reported on the remaining 80%. We observe that: **(1)** QUPAC outperforms baselines in almost all cases. RL-Attack is competitive in the case of substitute IsoNet++ (Edge) on victim GMN for AIDS dataset. **(2)** GMN is a more potent substitute for IsoNet (Edge) victim than IsoNet++ (Edge). **(3)** IsoNet (Edge) as a substitute is more effective than GMN against IsoNet++ (Edge). IsoNet++ (Edge) is a more effective substitute against GMN than IsoNet (Edge). Both GMN and IsoNet++

(Edge) are early interaction architectures while IsoNet (Edge) is late interaction resulting in a poorly aligned ranking and exhibiting poor adversarial performance.

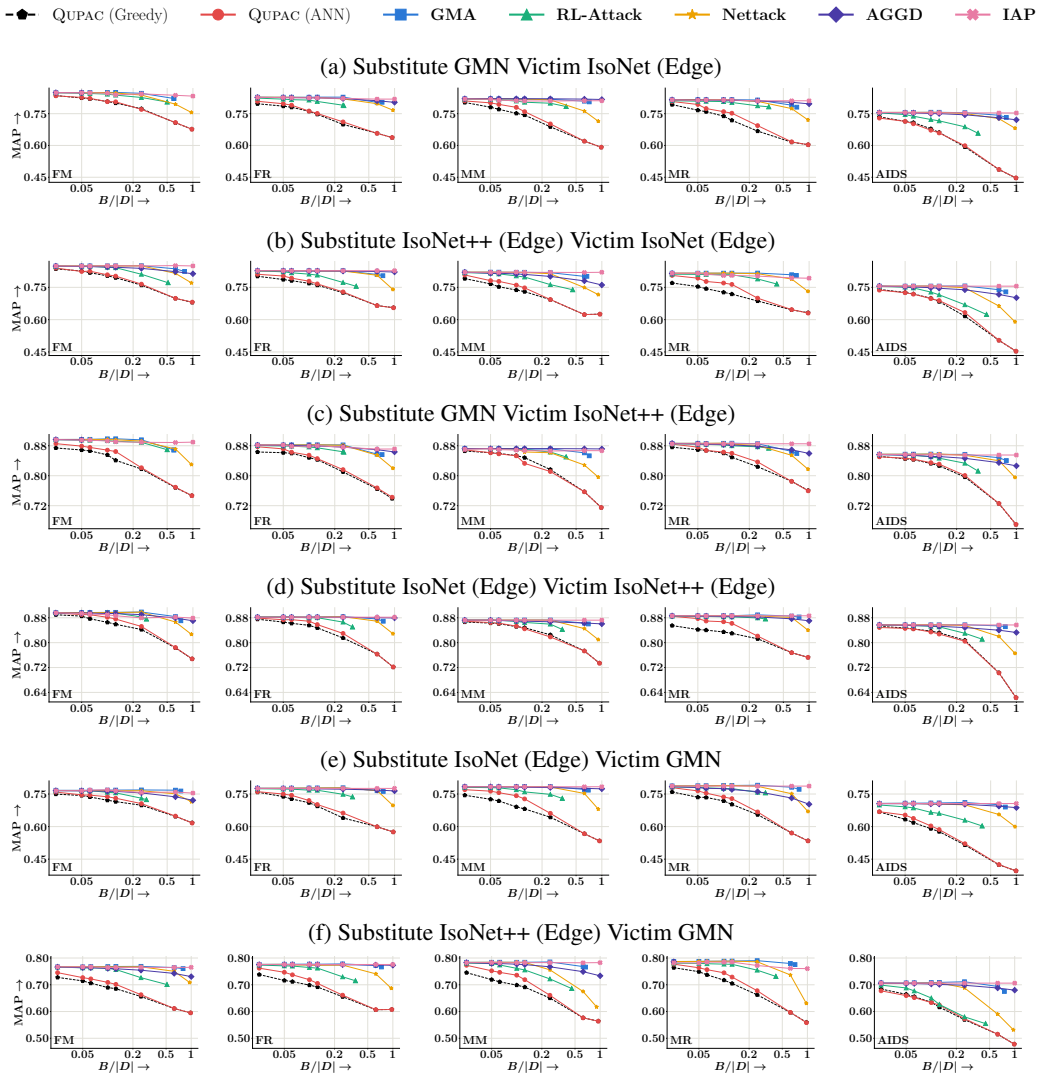


Figure 15: Performance measured in terms of MAP degradation (lower indicates a more successful attack) for QUPAC's variants and all baselines *viz.*, GMA, RL-Attack, Nettack, AGGD, and IAP. We consider transfer attack settings. The past query workload Q consists of queries from the validation set. QUPAC outperforms the baselines by a significant margin.

Transfer attack results on workload consisting of queries from the validation set. Figure 15 shows the transfer attack results when the attack is computed using queries from the validation set. We observe that: **(1)** QUPAC outperforms baselines in almost all cases. RL-Attack is competitive in the case of substitute IsoNet++ (Edge) on victim GMN for AIDS dataset. Although the margin of improvement for QUPAC compared to baselines decreases compared to test load. **(2)** Consistent to the test workload, we observe that GMN is a stronger substitute to IsoNet (Edge) victim than IsoNet++ (Edge). **(3)** We observe that, IsoNet (Edge) as a substitute is more effective than GMN against IsoNet++ (Edge).

G.3 Shift in Relevance Ratio due to Adversarial Perturbations

QuPAC computes $(S, \{\delta_c\})$ based on the query workload Q, and the same perturbation is applied to the corpus for all unseen queries Q^*. The adapted baselines also attack the corpus in a similar way, where the perturbation strategy is determined by the historic query workload. Hence, we expect that for some of the perturbations $\delta_j \in \{\delta_c\}, \exists q \in Q^*$ such that $\delta_j \notin \Gamma_{qj}$. For such queries, the perturbations on the corpus will shift the distribution of per-query positive relevance labels across the corpus. We quantify this shift as

$$\frac{1}{|Q^*|} \sum_{q \in Q^*} \frac{\Big| |\{G_{c_\oplus} \in \mathcal{G} : y(G_q, G_{c_\oplus}) = +1\}| - |\{G_{c_\oplus} \in \mathcal{G}_{S,\{\delta_c : c \in S\}} : y(G_q, G_{c_\oplus}) = +1\}| \Big|}{|\mathcal{G}|}$$

We report the average shift in distribution under different query workloads, and datasets, across the baselines, and QuPAC (Greedy) and QuPAC (ANN). We report the shifts due to perturbations in white box settings. Table 16 shows the shift in relevance ratio under white box attacks. **(1)** We see that across all datasets, the shift in relevance ratio is extremely small. Table 16 that the shift in distribution of positive corpus graphs across after perturbation across the unseen queries $q \in Q^*$ is $< 1\%$. **(2)** The results in Table 16 show that across all unseen queries, the distribution of positive-negative graph pairs $(G_{c_\oplus}, G_{c_\ominus})$ in the clean corpus $\mathcal{G}$ and the modified corpus $\mathcal{G}_{S,\{\delta_c : c \in S\}}$ remains almost the same (within 1%). This makes the attack more effective, since $\mathcal{G}_{S,\{\delta_c : c \in S\}}$ is almost indistinguishable from $\mathcal{G}$ in terms of distribution of relevance labels.

	FM		FR		MM		MR	
	B=40	B=80	B=40	B=80	B=40	B=80	B=40	B=80
Nettack	0.000 ± 0.000	0.001 ± 0.000	0.000 ± 0.000	0.001 ± 0.000	0.000 ± 0.000	0.001 ± 0.000	0.000 ± 0.000	0.001 ± 0.000
GMA	0.001 ± 0.000	0.002 ± 0.000	0.001 ± 0.000	0.002 ± 0.001	0.001 ± 0.000	0.003 ± 0.000	0.001 ± 0.000	0.002 ± 0.000
RL-Attack	0.001 ± 0.000	0.001 ± 0.001	0.000 ± 0.000	0.001 ± 0.000	0.000 ± 0.000	0.001 ± 0.001	0.002 ± 0.001	0.005 ± 0.002
AGGD	0.001 ± 0.000	0.003 ± 0.000	0.001 ± 0.000	0.002 ± 0.001	0.001 ± 0.000	0.002 ± 0.000	0.001 ± 0.000	0.002 ± 0.000
IAP	0.002 ± 0.001	0.002 ± 0.001	0.001 ± 0.000	0.003 ± 0.000	0.001 ± 0.000	0.001 ± 0.001	0.002 ± 0.001	0.002 ± 0.001
QuPAC (Greedy)	0.002 ± 0.001	0.005 ± 0.002	0.002 ± 0.001	0.005 ± 0.002	0.002 ± 0.001	0.006 ± 0.002	0.003 ± 0.002	0.007 ± 0.003
QuPAC (ANN)	0.003 ± 0.002	0.006 ± 0.004	0.003 ± 0.001	0.005 ± 0.002	0.002 ± 0.000	0.004 ± 0.001	0.003 ± 0.001	0.007 ± 0.002

Table 16: Shift in relevance ratio for different attack methods in white box setting: Nettack, GMA, RL-Attack, AGGD, IAP, and QuPAC (Greedy) and QuPAC (ANN). The victim model is IsoNet++ (Edge). We report the change in fraction of positive relevance query-corpus pairs due to perturbations, averaged across the unseen queries Q^*, across 4 datasets: FM, FR, MM and MR, and 2 budgets $B = \{40, 80\}$. The table shows the average shift in relevance ratio averaged across 5 different query workloads. We see that the shift in distribution of positive corpus graphs across queries due to perturbations is extremely small $(< 1\%)$.

G.4 Statistical Significance of QuPAC's results

We conduct one sided paired t-test comparing the performance of QuPAC (ANN) with the closest performing baseline (RL-Attack) on five randomly drawn workload of 20%-subset of test queries. Table 17 shows that QuPAC (ANN) outperforms RL-Attack by a statistically significant margin ($p < 0.012$, for white box setting; and $p < 0.006$, for black box setting). For white box setting, $p < 1e-3$ is most cases.

	FM	FR	MM	MR
$B = 20$	1.24×10^{-3}	1.00×10^{-4}	4.28×10^{-4}	1.16×10^{-2}
$B = 40$	3.29×10^{-5}	4.76×10^{-5}	5.14×10^{-5}	2.92×10^{-4}
$B = 50$	2.85×10^{-5}	2.61×10^{-5}	7.24×10^{-5}	3.32×10^{-4}
$B = 80$	7.03×10^{-5}	1.96×10^{-5}	5.55×10^{-5}	1.00×10^{-5}
$B = 100$	4.75×10^{-5}	8.52×10^{-6}	1.66×10^{-5}	2.51×10^{-5}
$B = 200$	1.75×10^{-4}	5.37×10^{-6}	1.50×10^{-6}	5.27×10^{-6}
$B = 500$	8.36×10^{-8}	1.87×10^{-5}	2.02×10^{-5}	3.74×10^{-5}
$B = 800$	7.74×10^{-7}	1.89×10^{-5}	4.91×10^{-6}	3.76×10^{-6}

	FM	FR	MM	MR
$B = 20$	5.21×10^{-4}	3.47×10^{-4}	4.78×10^{-3}	5.35×10^{-3}
$B = 40$	2.57×10^{-4}	1.79×10^{-5}	2.57×10^{-4}	2.52×10^{-5}
$B = 50$	1.37×10^{-4}	1.65×10^{-5}	3.52×10^{-4}	1.91×10^{-5}
$B = 80$	1.06×10^{-4}	8.22×10^{-6}	9.19×10^{-6}	2.07×10^{-6}
$B = 100$	1.24×10^{-4}	1.86×10^{-5}	2.60×10^{-6}	1.17×10^{-4}
$B = 200$	4.80×10^{-5}	1.04×10^{-4}	6.35×10^{-6}	3.26×10^{-5}
$B = 500$	7.59×10^{-6}	5.46×10^{-6}	7.31×10^{-6}	6.99×10^{-5}
$B = 800$	4.87×10^{-6}	8.47×10^{-5}	2.65×10^{-5}	4.02×10^{-6}

Table 17: We use a one-sided paired t-test across five historical query workloads, each comprising 20% of the test queries, to assess whether QuPAC (ANN) significantly outperforms RL-Attack. The top table reports the p-values for the white box setting, and the bottom table reports p-values for the black box setting on the victim model IsoNet++ (Edge). QuPAC (ANN) outperforms the baseline by a statistically significant margin ($p < 0.012$).

G.5 TIMING ANALYSIS: QUPAC (GREEDY) VS QUPAC (ANN)

We report timing comparison between QUPAC (Greedy) and QUPAC (ANN) in this section. The time for QUPAC (ANN) computes the end-to-end time, including retrieval plus greedy routine. For QUPAC (Greedy), the timing is reported for running the greedy algorithm exhaustively.

We compare the runtime of QUPAC (Greedy) with QUPAC (ANN) in Table 18 for $\mathcal{B} = 200B$ and $\mathcal{B} = 400B$, with $B = 100$. From the results, we observe that QUPAC (ANN) runs significantly faster than QUPAC (Greedy) a speedup roughly of the order of 1.5 to 2.0 times.

$B = 100$						
	IsoNet (Edge)			IsoNet (Node)		
Dataset	QUPAC (Greedy)	QUPAC (ANN) $\mathcal{B} = 200B$	QUPAC (ANN) $\mathcal{B} = 400B$	QUPAC (Greedy)	QUPAC (ANN) $\mathcal{B} = 200B$	QUPAC (ANN) $\mathcal{B} = 400B$
AIDS	103.6	56.9	61.8	107.3	56.8	59.8
FR	105.1	55.2	61.9	105.6	56.6	61.6
FM	107.0	57.5	66.3	106.9	56.4	64.3
MR	95.4	51.0	57.7	95.8	52.9	58.6
MM	90.9	52.2	57.7	91.5	53.4	62.7

	IsoNet++ (Edge)			IsoNet++ (Node)		
Dataset	QUPAC (Greedy)	QUPAC (ANN) $\mathcal{B} = 200B$	QUPAC (ANN) $\mathcal{B} = 400B$	QUPAC (Greedy)	QUPAC (ANN) $\mathcal{B} = 200B$	QUPAC (ANN) $\mathcal{B} = 400B$
AIDS	108.4	58.2	61.8	102.4	56.2	65.8
FR	106.3	58.4	58.9	113.7	55.7	61.7
FM	111.1	57.7	62.6	108.1	56.9	65.3
MR	95.2	56.7	61.2	97.3	50.2	60.2
MM	89.9	49.8	59.0	92.0	53.2	62.7

	GEN			GMN		
Dataset	QUPAC (Greedy)	QUPAC (ANN) $\mathcal{B} = 200B$	QUPAC (ANN) $\mathcal{B} = 400B$	QUPAC (Greedy)	QUPAC (ANN) $\mathcal{B} = 200B$	QUPAC (ANN) $\mathcal{B} = 400B$
AIDS	101.7	59.9	64.7	102.4	57.8	61.9
FR	112.6	56.7	60.7	107.4	62.0	61.6
FM	109.1	58.0	65.4	105.0	56.4	60.6
MR	94.3	57.0	57.9	94.4	47.9	59.8
MM	89.5	52.7	58.1	92.3	53.4	62.1

Table 18: End to end running time of QUPAC (Greedy) and QUPAC (ANN) for $\mathcal{B} \in \{200B, 400B\}$. In this table, $B = 100$ and all run times are reported in seconds. We observe that QUPAC (ANN) is significantly faster than QUPAC (Greedy) showing that the ground set pruning makes the attack computation very efficient.

G.6 Space Complexity of QuPAC (ANN)

In Table 19, we report the overall storage cost of implementing our method. We cache the substitute model embeddings d_θ, representations by $\boldsymbol{\zeta}_{\psi,j}$ neural network and the LSH Index. We see that the storage cost of our method is small, of the order of 100 MBs.

Dataset	d_θ (MB)	$\boldsymbol{\zeta}_{\psi,j}(\cdot,\theta)$ (MB)	LSH Index (MB)
AIDS	6.3942	103.8226	0.0040
FM	5.6709	92.0683	0.0040
FR	5.8474	94.9365	0.0040
MM	5.7429	93.2380	0.0040
MR	5.8184	94.4654	0.0040

Table 19: Space complexity for QuPAC (ANN). We cache d_θ embeddings, $\boldsymbol{\zeta}_{\psi,j}(\cdot,\theta)$ and the LSH index in memory. We see that the overall storage cost of our method is small.

G.7 Ablation: Parameter Perturbation Robustness

QuPAC trains a neural network $\boldsymbol{f}_\psi$ to learn worst-case perturbations in an ϵ-ball around the learned parameters of the substitute model d_θ. The training objective of $\boldsymbol{f}_\psi$ (12) makes the adversary's approximations of d_θ robust to the unseen queries Q^*. To illustrate the contribution of this robust training, we train an adversary using a non-robust objective, that is,

$$\min_{\psi} \sum_{c,q,j} \mathrm{BCE}(\mathrm{sigmoid}(-(\boldsymbol{\eta}_{\psi,j}(G_q,\theta))^\top \boldsymbol{\zeta}_{\psi,j}(G_c,\theta)), y(G_q,G_c)) \tag{59}$$

	FM		MM	
	$B=40$	$B=80$	$B=40$	$B=80$
QuPAC (ANN, non-robust)	0.826	0.800	0.825	0.804
QuPAC (ANN)	**0.808**	**0.768**	**0.823**	**0.781**

Table 20: Comparison of QuPAC's adversarial performance when the neural network $\boldsymbol{f}_\psi$ is trained on its robust training objective (12), versus a non-robust objective (59). We report the adversarial MAP of these two attack methods for FM and MM datasets on victim IsoNet++ (Node) for budgets $B = 40$ and $B = 80$. We see that QuPAC (ANN) performs better when trained on the robust objective.