ROKSANA: An Open-Source Toolkit for Robust Graph-Based Keyword Search

Radin Hamidi Rad
radin.rad@utoronto.ca
University of Toronto
ON, Canada

Amir Khosrojerdi
amir.khosrojerdi@queensu.ca
Queen's University
ON, Canada

Ebrahim Bagheri
ebrahim.bagheri@utoronto.ca
University of Toronto
ON, Canada

Abstract

We introduce ROKSANA, an open-source Python toolkit designed to support research in graph-based keyword search under adversarial settings. ROKSANA provides a modular environment for dataset handling, graph neural network (GNN)-based retrieval, and adversarial attack modeling, enabling systematic evaluation of search robustness. The framework integrates built-in retrieval and attack methods while allowing seamless customization of search algorithms and perturbation strategies. Users can benchmark performance on a centralized leaderboard, generate reproducible evaluation reports, and explore ranking behaviors through an interactive web-based visualization interface. By centering around reproducibility, extensibility, and collaborative benchmarking, ROKSANA serves as a comprehensive platform for advancing robust and interpretable keyword search in graphs. This demonstration will showcase ROKSANA's capabilities in real-time, illustrating its impact on experimental workflows and adversarial robustness analysis in graph IR research.

CCS Concepts

• **Computing methodologies** → **Learning latent representations**; • **Mathematics of computing** → *Graph algorithms*; • **Information systems** → *Retrieval models and ranking*.

ACM Reference Format:
Radin Hamidi Rad, Amir Khosrojerdi, and Ebrahim Bagheri. 2025. ROKSANA: An Open-Source Toolkit for Robust Graph-Based Keyword Search. In *The 48th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'25), July 13–18, 2025*. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3511808.3557181

1 Introduction

Graph-based keyword search is a critical problem in information retrieval (IR) and network analysis, enabling the retrieval of semantically relevant nodes from large, structured graphs [5]. Despite extensive research, the field lacks a standardized, extensible framework that integrates dataset preprocessing, graph representation learning, search customization, adversarial robustness analysis, and systematic evaluation. This fragmentation hinders reproducibility, limits comparative assessments, and impedes the development of robust graph search techniques.

We present ROKSANA[1], an open-source Python-based framework that unifies these essential components into a cohesive, modular system. Available as a PyPI package (`pip install roksana`[2]), ROKSANA provides end-to-end support for graph-based keyword search, incorporating graph neural network (GNN)-driven retrieval methods (GCN[6], GAT [10], GraphSAGE [4]) and adversarial perturbation models (Random, Viking [3], PageRank [1], Degree-based attacks) to analyze search robustness under node demotion and promotion scenarios. The framework facilitates standardized dataset handling, supporting both built-in corpora (Cora [7], Citeseer [8], Pubmed [9]) and user-provided graphs, with automated preprocessing pipelines to ensure consistency across experiments.

Beyond retrieval and attack modeling, ROKSANA offers a structured evaluation module that generates downloadable performance reports, a benchmarking leaderboard to compare retrieval models against state-of-the-art baselines, and an interactive visualization tool[3] for in-depth analysis of graph search behavior. These capabilities establish ROKSANA as a comprehensive, extensible platform for advancing research in graph-based keyword search and adversarial robustness.

The demo intends to present to the audience detail ofs the architectural design and functional capabilities of ROKSANA, demonstrate its application through representative use cases, and discuss its impact on the development of more resilient and interpretable graph retrieval systems. As a demo, we intend to offer an interactive, hands-on experience that allows researchers and practitioners to engage directly with ROKSANA's functionalities. Attendees will observe how the framework streamlines dataset preparation, keyword-based retrieval using graph neural networks, and adversarial attack simulations, providing real-time insights into the robustness and effectiveness of different retrieval strategies. The demo will showcase key features, including dataset ingestion, query execution, attack simulations, benchmarking via the leaderboard, and interpretability tools for visualizing search perturbations.

ROKSANA represents a timely and significant contribution to the information retrieval (IR) community. As graph-based retrieval continues to grow in relevance, particularly in domains such as scientific literature search, knowledge graphs, and recommendation systems, there is a strong need for standardized, reproducible methodologies that support both *retrieval effectiveness* and *robustness analysis*. By integrating graph neural network-based search with adversarial attack modeling, ROKSANA enables researchers

SIGIR'25, July 13–18, 2025, Padua, Italy

ACM ISBN 978-1-4503-9236-5/22/10
https://doi.org/10.1145/3511808.3557181

[1] http://roksana.ls3.rnet.torontomu.ca/
[2] https://pypi.org/project/ROKSANA
[3] https://tools.ls3.rnet.torontomu.ca/adversarialgraphs

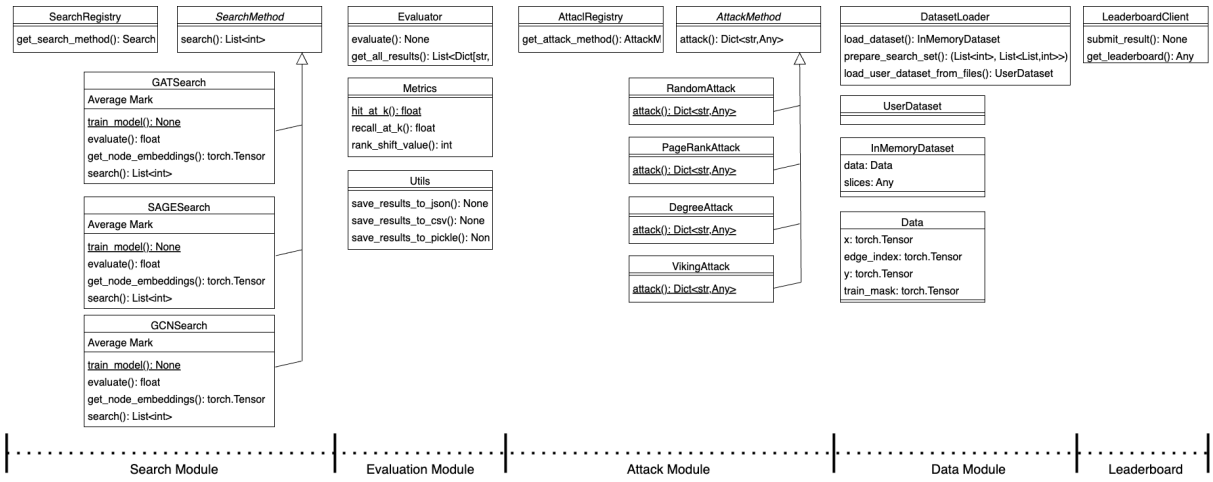

Figure 1: Inheritance Hierarchy for classes in ROKSANA python package.

to systematically evaluate vulnerabilities in retrieval systems, a crucial aspect as real-world applications increasingly encounter adversarial manipulations.

Furthermore, the framework lowers the barrier to entry for new researchers by providing preprocessed datasets, built-in retrieval models, and structured evaluation tools, supporting systematic experimentation and enabling reproducibility. Its extensible design encourages contributions from the IR community, allowing for the integration of novel retrieval architectures, adversarial strategies, and evaluation metrics. Through this demo, we aim to provide the IR community with a practical, scalable, and extensible platform that not only facilitates cutting-edge research but also establishes a standardized framework for evaluating and improving graph retrieval systems.

2 Toolkit Overview

ROKSANA is a modular and extensible toolkit designed to support research in graph-based keyword search and adversarial robustness analysis. It is accessible via `pip install roksana` and provides an integrated environment for dataset handling, graph-based retrieval, adversarial attack simulations, evaluation, benchmarking, and visualization. Each module operates independently yet seamlessly integrates into the overall framework, ensuring flexibility for researchers with diverse experimental needs. The inheritance hierarchy for ROKSANA's core classes is depicted in Figure 1.

At the core of ROKSANA, the **data module** facilitates structured data ingestion and preprocessing. It supports built-in datasets such as Cora [7], Citeseer [8], and Pubmed [9], along with user-defined graphs in a format compatible with Torch Geometric [2]. Researchers can transition from raw data to structured graph representations with minimal effort, enabling them to focus on higher-level retrieval and analysis tasks rather than extensive data wrangling. ROKSANA further allows users to partition datasets into training, testing, and search subsets, ensuring consistency in experimentation and evaluation.

The **search module** enables efficient keyword-based retrieval using graph neural network (GNN)-generated embeddings. ROKSANA provides implementations of GCN [6], GAT [10], and GraphSAGE [4], allowing nodes to be represented based on structural and semantic properties. Keyword queries are processed by computing similarity scores between query embeddings and node representations, facilitating ranking and retrieval. Users can also define custom search methods by extending the `SearchMethod` interface, enabling integration of domain-specific heuristics. Additionally, custom search methods can be registered within the framework for reuse, supporting modular experimentation across projects.

To evaluate search robustness, the **attack module** implements adversarial perturbations that manipulate graph structures, simulating realistic threats to retrieval models. ROKSANA provides built-in attack strategies, including Random, Viking [3], PageRank-based [1], and Degree-based node modifications, which influence search rankings through node demotion and promotion. A key contribution of the toolkit is the ability to define and integrate custom attack strategies, allowing researchers to systematically benchmark their adversarial techniques against established baselines. This module is critical for analyzing vulnerabilities in graph search models and devising defense mechanisms.

ROKSANA incorporates a **comprehensive evaluation module** to ensure reproducible assessment of retrieval and adversarial impact. Standard ranking metrics such as hit@k, recall@k, and rank shift value are provided to quantify search effectiveness and robustness. The module automatically generates downloadable reports in CSV, JSON, and Pickle formats, enabling seamless documentation of experimental results. Built-in pipelines facilitate large-scale comparisons across datasets, search methods, and attack strategies, ensuring methodical and consistent evaluation.

A distinctive feature of ROKSANA is its **leaderboard and benchmarking system**, designed to favor transparency and collaborative progress. Researchers can submit results based on predefined dataset splits and compare their performance against baseline methods and state-of-the-art models integrated into the toolkit. This promotes a standardized framework for benchmarking, helping the community identify strengths and weaknesses in retrieval approaches while encouraging reproducibility and innovation.

To support interpretability, ROKSANA provides an **interactive visualization tool**, offering researchers an intuitive means of exploring graph structures, embeddings, and search results. Users can analyze retrieval behaviors under various attack scenarios, observe ranking changes, and gain qualitative insights into how different search methods respond to adversarial perturbations. By bridging numerical evaluation with visual exploration, this module aids in refining retrieval algorithms and enhancing the robustness of graph-based search models.

Overall, ROKSANA serves as an end-to-end research toolkit, addressing key challenges in graph keyword search while ensuring extensibility, reproducibility, and ease of experimentation. Through this demo, attendees will gain hands-on experience with its modular capabilities, from data processing and search execution to adversarial evaluation and visualization, showcasing its potential to advance research in information retrieval and adversarial graph learning.

3 Demonstrating Use Cases

To illustrate ROKSANA's capabilities, we present three key use cases: baseline retrieval experiments, adversarial robustness evaluation, and the integration of custom search and attack methods.

A fundamental use case involves executing a *baseline keyword search* experiment on the Cora [7] dataset. With a single command, users can load the dataset and partition it into training and testing subsets. A GCN [6] model is trained to generate node embeddings, which serve as the foundation for keyword-based retrieval. The trained model enables ranking nodes based on their similarity to a given query embedding, allowing users to execute retrieval with minimal setup. The following code demonstrates the streamlined execution of this process:

Listing 1: Baseline Search with GCN

```
from roksana.datasets import load_dataset,
    prepare_test_set
from roksana.search_methods import
    get_search_method
from roksana.evaluation import Evaluator

data=load_dataset('cora','data/')[0]
queries,golds=prepare_test_set(data,0.1,123)
gcn=get_search_method('gcn',data=data,epochs=10,lr
    =0.01)
res=gcn.search(data.x[queries[0]],top_k=5)
print("Retrieved:",res)
ev=Evaluator(gcn,gcn,[5])
ev.evaluate(queries,golds,'eval','gcn.csv')
```

Beyond basic retrieval, ROKSANA facilitates *adversarial robustness* studies by enabling attack simulations on keyword search results. Users can assess how retrieval models respond to perturbations in the underlying graph structure. For instance, applying a Viking attack [3] to the Citeseer [8] dataset modifies node importance, influencing search rankings. By running the evaluation module before and after the attack, researchers can quantify the impact using metrics such as hit@k, recall@k, and rank shift values. The following example demonstrates how an attack is applied and how retrieval effectiveness is reassessed post-perturbation:

Listing 2: Applying Viking Attack

```
from roksana.attack_methods import
    get_attack_method
viking=get_attack_method('viking',data=data,
    perturbations=2)
for q in queries:
    viking.attack(q,2)
gcn_after=get_search_method('gcn',data=data,epochs
    =10,lr=0.01)
ev=Evaluator(gcn,gcn_after,[5])
ev.evaluate(queries,golds,'eval','gcn_viking.csv')
```

To support extensibility, ROKSANA allows researchers to define *custom search and attack methods*, integrating them into the evaluation and benchmarking pipeline. A researcher may implement a novel retrieval strategy by extending the `SearchMethod` interface, incorporating domain-specific heuristics or specialized learning techniques. Similarly, custom attack strategies can be implemented to study new forms of adversarial influence on retrieval models. These user-defined methods seamlessly integrate with ROKSANA's evaluation framework, enabling comparative assessment against built-in search and attack strategies. The example below illustrates the creation of a custom search and attack method:

Listing 3: Custom Search & Attack

```
from roksana.search_methods.base_search import
    SearchMethod
from roksana.attack_methods.base_attack import
    AttackMethod

class MySearch(SearchMethod):
    def search(self,qf,top_k=5):return range(top_k
        )
my_srch=MySearch(data=data)
print("Custom:",my_srch.search(data.x[queries
    [0]],5))

class MyAttack(AttackMethod):
    def attack(self,node,p=1):return {"changed":
        node}
my_atk=MyAttack(data=data)
print("Attack:",my_atk.attack(queries[0],3))
```

Through these use cases, ROKSANA demonstrates its flexibility as a research toolkit for graph-based keyword search. The framework streamlines experimental workflows, provides robust evaluation tools, and facilitates adversarial robustness studies, making it an essential resource for researchers working in information retrieval and graph learning.

3.1 Leaderboard and Benchmarking

The `leaderboard` feature in ROKSANA provides a structured benchmarking environment for keyword search and adversarial attack methods, ensuring transparency, reproducibility, and comparability across different approaches. By standardizing evaluation settings and datasets, the leaderboard enables researchers to systematically assess retrieval effectiveness and adversarial robustness under consistent conditions.

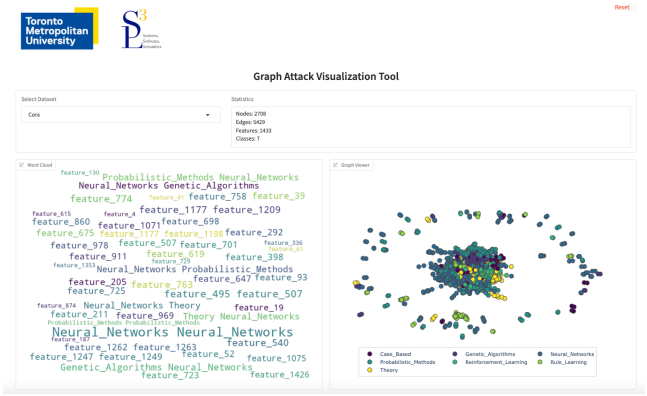

Figure 2: Screenshot of the user interface of the visualization module for graph search and attack tasks.

Users can execute search and attack pipelines locally, compute evaluation metrics using ROKSANA's evaluation module, and submit their results to a centralized leaderboard. Each submission includes key performance indicators such as hit@k, recall@k, and rank shift values, which quantify both retrieval quality and susceptibility to perturbations. Once uploaded, the results are integrated into a publicly accessible leaderboard that ranks methods based on their performance, allowing users to directly compare their approaches against existing baselines and state-of-the-art techniques.

Seamlessly integrated with the evaluation module, the leaderboard simplifies the submission process, requiring only a few lines of code to upload results. The following example demonstrates how a user interacts with the leaderboard:

Listing 4: Leaderboard Submission

```
from roksana.leaderboard import LeaderboardClient
res=ev.get_all_results()
cli=LeaderboardClient('https://tools.ls3.rnet.
    torontomu.ca/adversarialgraphs/leaderboard','
    API_KEY')
cli.submit_result('user123',{'score':res[0]['
    Hit@k_before_attack']})
lb=cli.get_leaderboard()
print("Leaderboard:",lb)
```

Once submitted, the leaderboard's web interface displays ranked methods, corresponding evaluation metrics, and user contributions, offering a centralized reference point for measuring progress in graph-based keyword search and adversarial retrieval resilience. Through this framework, ROKSANA establishes a community-driven benchmarking ecosystem that advances the state of research in robust and interpretable information retrieval.

3.2 Visualization

ROKSANA provides an interactive `web-based application`[4] designed to facilitate the exploration and analysis of graph-based keyword search and adversarial perturbations. Through an intuitive web interface (Figure 2), users can interactively examine how different datasets, retrieval models, and attack strategies influence search results.

The platform allows users to select from built-in datasets, including Cora [7], Citeseer [8], and Pubmed [9], or upload their own graph data in a compatible format. Once a dataset is chosen, users can apply various retrieval models such as GCN [6], GAT [10], and GraphSAGE [4] to generate node embeddings and execute keyword-based search queries. The system supports both default retrieval methods and user-defined search algorithms, enabling flexible experimentation with novel approaches.

To assess the impact of adversarial interventions, users can apply built-in attack strategies, including Random, Viking [3], PageRank-based [1], and Degree-based attacks. Additionally, the framework allows the integration of custom attack models, providing a controlled environment for stress-testing retrieval performance. The visualization module enables real-time observation of ranking shifts induced by these perturbations, offering a direct way to evaluate search robustness.

Beyond retrieval and attack analysis, the interface displays evaluation metrics such as hit@k, recall@k, and rank shift values before and after attack application. Users can dynamically adjust query parameters, explore ranking distributions, and visually inspect how adversarial modifications alter node importance in the graph structure. Furthermore, the dashboard integrates with the ROKSANA leaderboard, allowing users to compare their method's performance against state-of-the-art baselines and community-submitted results in a reproducible and structured manner.

By providing an interactive and visually driven approach, the ROKSANA visualization module reduces reliance on static evaluation reports, allowing researchers to gain deeper, more intuitive insights into search behaviors. Instead of merely analyzing numerical results, users can experiment with different configurations in a real-time environment, offering a more exploratory and interpretable workflow.

4 Concluding Remarks

In this demo paper, we have introduced ROKSANA, an open-source and extensible toolkit designed to facilitate research in graph-based keyword search by integrating data handling, retrieval, adversarial attack modeling, evaluation, leaderboard benchmarking, and interactive visualization into a unified framework. By offering built-in retrieval and attack methods alongside extensible interfaces, ROKSANA enables researchers to conduct systematic, reproducible, and scalable experiments while lowering the barrier to exploring robustness in keyword search.

This demonstration will highlight how ROKSANA streamlines experimental workflows, facilitates comparative evaluation, and fosters collaboration through its leaderboard-driven benchmarking. By providing an interactive environment where users can execute search queries, apply adversarial perturbations, visualize ranking effects, and submit their results for community comparison, ROKSANA establishes itself as a practical and impactful tool for both academic research and real-world applications in network analysis. Through hands-on engagement, attendees will experience how ROKSANA enables systematic exploration of graph retrieval methods.

[4] https://tools.ls3.rnet.torontomu.ca/adversarialgraphs

References

[1] Sergey Brin and Lawrence Page. 2012. Reprint of: The anatomy of a large-scale hypertextual web search engine. *Comput. Networks* 56, 18 (2012), 3825–3833. https://doi.org/10.1016/J.COMNET.2012.10.007

[2] Matthias Fey and Jan E. Lenssen. 2019. Fast Graph Representation Learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*.

[3] Viresh Gupta and Tanmoy Chakraborty. 2021. VIKING: Adversarial Attack on Network Embeddings via Supervised Network Poisoning. In *Advances in Knowledge Discovery and Data Mining - 25th Pacific-Asia Conference, PAKDD 2021, Virtual Event, May 11-14, 2021, Proceedings, Part III (Lecture Notes in Computer Science, Vol. 12714)*, Kamal Karlapalem, Hong Cheng, Naren Ramakrishnan, R. K. Agrawal, P. Krishna Reddy, Jaideep Srivastava, and Tanmoy Chakraborty (Eds.). Springer, 103–115. https://doi.org/10.1007/978-3-030-75768-7_9

[4] William L. Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive Representation Learning on Large Graphs. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (Eds.). 1024–1034. https://proceedings.neurips.cc/paper/2017/hash/5dd9db5e033da9c6fb5ba83c7a7ebea9-Abstract.html

[5] Yu Hao, Xin Cao, Yufan Sheng, Yixiang Fang, and Wei Wang. 2021. KS-GNN: Keywords Search over Incomplete Graphs via Graphs Neural Network. *Advances in Neural Information Processing Systems* 34 (2021).

[6] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net. https://openreview.net/forum?id=SJU4ayYgl

[7] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net. https://openreview.net/forum?id=SJU4ayYgl

[8] Ryan A. Rossi and Nesreen K. Ahmed. 2015. The Network Data Repository with Interactive Graph Analytics and Visualization. In *AAAI*. https://networkrepository.com

[9] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Gallagher, and Tina Eliassi-Rad. 2008. Collective Classification in Network Data. *AI Mag.* 29, 3 (2008), 93–106. https://doi.org/10.1609/AIMAG.V29I3.2157

[10] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net. https://openreview.net/forum?id=rJXMpikCZ