Efficient Multi-Criteria Important Node Identification via Ego-Network Approximation

Andreas Kosmatopoulos*, Taxiarchis Skouras†, Theodora Tsikrika‡,
Stefanos Vrochidis§ and Ioannis Kompatsiaris¶
Information Technologies Institute
Centre for Research and Technology Hellas
Thessaloniki, Greece
Email: *akosmato@iti.gr, †tax_skouras@iti.gr, ‡theodora.tsikrika@iti.gr, §stefanos@iti.gr, ¶ikom@iti.gr

***Abstract*—Graphs are fundamental tools for modeling complex relationships across a wide range of domains, where identifying the most influential or structurally important nodes, known as key actors, is a central problem. Single centrality metrics often fail to capture universally influential nodes, highlighting the necessity for multi-criteria approaches. In this paper, we propose an efficient and effective approximation scheme that identifies key actors by aggregating multiple centrality measures through an ego-network-based graph-convolutional network (GCN). We introduce two novel heuristics: (i) size-aware ego-network retrieval, which constrains computational overhead by limiting ego-network sizes, and (ii) selective feature computation, which reduces runtime by computing detailed local features only for structurally significant nodes. Extensive experimental validation on both synthetic and real-world datasets demonstrates that our approximation achieves substantial computational savings, while closely matching the accuracy and ranking quality of baseline methods. Last but not least, robustness analyses underscore the practical utility of the identified key nodes, showcasing their effectiveness in network dismantling and disruption scenarios.**

***Index Terms*—graph, key actor, influential node, approximate query, centrality measure.**

I. Introduction

Graphs and networks are ubiquitous in modern applications and serve as an intuitive abstraction for modeling relationships across diverse domains, such as biology, medicine, sociology, and technology. Graph-based representations model real-world entities as graph nodes, and their interactions or relationships as edges, and this enables their use for several application domains such as influence maximization on social networks [1], drug discovery [2], network security [3], epidemic control [4], misinformation management [5], and others.

Considerable attention has been devoted to developing methods and processes that identify key actor nodes in a graph [6]–[8], namely those that exhert influence over other nodes or hold structural or topological importance within the network. Consequently, accurate identification of such key nodes is an important task that is able to aid in maximizing information spread, halting epidemics, hardening network security, and disrupting illicit networks. Typically, the identification of key actor nodes relies on single centrality measures that capture an intuitive aspect of node importance, such as its degree [9], betweenness [10], closeness [11], eigenvector [12] or pagerank [13] centrality. However, single centrality measures have been shown [14], [15] to be insufficient as the best predictors of influence across different networks. For example, a node with many direct connections (high degree) may still be peripheral in terms of long-range impact, while a node that bridges communities (high betweenness) might lack strong local reach. As a result, the need arises for methods that combine multiple centrality measures to more effectively identify the most influential nodes in a network.

Another practical limitation of single-metric methods arises under resource constraints (such as human, operational, or budgetary constraints), where only a limited number of nodes can be selected or acted upon in real-world settings. Basu and Sen [16], [17] highlight the inefficiency of relying solely on centrality rankings when resources are limited, showing that optimized node selection can achieve broader impact with fewer monitored nodes. Under such constraints, it is sensible to focus on nodes that are simultaneously important in multiple respects, in other words those that rank highly according to multiple centrality criteria.

In this paper, we build upon the work of Kosmatopoulos et al. [18] who introduced a graph neural network-based framework for aggregating multiple centrality rankings into a unified key actor score, and we extend their approach to improve scalability and applicability to larger or denser networks of different topologies. In their method, a Graph Convolutional Network (GCN) is trained to learn the Borda ranking of nodes based on a fixed set of classical centrality metrics, namely Degree, Betweenness, Closeness, Eigenvector, and PageRank. While effective, this approach can still be computationally expensive in large-scale or dense graphs; to address this, our work introduces a lightweight approximation scheme through two heuristic optimizations that significantly reduces execution time with only a marginal decrease in effectiveness when identifying "universally" influential nodes.

Contributions. In summary, our contributions are as follows:

- We extend the GCN-based framework of Kosmatopoulos et al. [18] for identifying "universally influential" nodes through Borda aggregation, by introducing two novel

This work has received funding from the European Union's H2020 research and innovation programme as part of the STARLIGHT (GA No 101021797) project.

heuristics that prune the total number of computations and yield substantial speedups with minimal accuracy loss.
- We demonstrate that our approximation variant achieves significant speed-ups with only marginal reductions in effectiveness, as measured by Accuracy and NDCG.
- We conduct an ablation study on key heuristic parameters to assess their impact on runtime and ranking quality.
- We evaluate the robustness and structural impact of the influential nodes identified by our method through measuring network degradation and showing that our approach outperforms state-of-the-art techniques in terms of structural disruption.

The rest of the paper is structured as follows. Section II reviews related work, while Section III details our method. Section IV presents the experimental evaluation of the proposed method and Section V concludes our work and presents future research directions.

II. Related Work

In this section, we review previous work performed on two domain areas that are closely linked to the proposed method before moving on to the description of our framework.

A. Key Actor Identification in Graphs

A natural and intuitive approach for identifying influential nodes, key actors, or critical nodes in a network is through the use of classical centrality measures [9]–[13] that aim to quantify a node's importance in the network through a single aspect of influence. Similarly, k-shell decomposition [19] provides an alternative way to identify influential nodes, with several refinements and extensions [20], [21] of the original method proposed in subsequent works.

Other approaches include voting-based methods, such as VoteRank [22] and VoteRank++ [23], which iteratively select influential nodes by aggregating neighborhood "votes", and entropy-based or information-based techniques, such as EnRenew [24], that promote nodes that can reach disparate parts of the network. Multi-criteria and hybrid measures, such as MCDE [25] and ECRM [26], have also shown promise in identifying nodes of importance; MCDE integrates multiple centrality measures using an ensemble decision-making framework, while ECRM computes influence through local structural entropy and neighborhood correlation.

B. Aggregation of Centrality Measures

Several works have explored aggregation approaches that combine multiple centrality indices to enable the identification of important nodes in networks. Mo et al. [27] proposed an evidential method based on Dempster–Shafer theory to combine evidence from several centrality measures that aimed to yield a consensus ranking. Liu et al. [28] presented a multi-attribute ranking where various centralities are weighted and combined into an overall importance score. Bian et al. [29] used the Analytic Hierarchy Process (AHP) to rank influential nodes by performing pairwise comparisons to assess their relative importance, and then computing a weighted sum score for each node. Ibnoulouafi et al. [30] introduced M-Centrality which ranks nodes by combining global position (core number from a K-shell decomposition) with neighborhood degree variation to identify key actors using complementary aspects of centrality. Zeng et al. [31] introduce a multi-criteria framework ("C^3-TOPSIS-Pareto") that evaluates nodes along three dimensions (structural cohesion, communication connectivity, and role uniqueness) and then ranks nodes via a TOPSIS [32] multi-attribute decision method combined with a Pareto-dominance filter. Li et al. [33] propose a composite centrality called Degree–K-shell–Betweenness Centrality (DKBC), which integrates degree centrality, k-shell position, and an intermediate influence factor under a gravity-model principle to identify key actor nodes across networks. Similarly, the MCDE model [25] (see Section II-A) employs an ensemble learning approach to integrate multiple centrality measures.

An intuitive aggregation technique is the Borda count [34], a consensus voting method where each centrality measure acts as an "expert" and "votes" by ranking nodes. These rankings are then combined to produce an overall score. Tian and Chen [35] applied a Borda count based multi-centrality model to maritime transport networks, to identify vulnerable ports and improve network robustness. Madotto and Liu [36] aggregated multiple centrality rankings into a single one using the Borda count, after pruning the set of centralities based on correlation and entropy-based information. Lastly, Kosmatopoulos et al. [18] introduced a GCN-based model aiming to find universally influential nodes by approximating the Borda count of a node's multiple centrality rankings. While previous research has focused on aggregating centrality metrics or algorithmic strategies for identifying influential nodes, our work builds on the method proposed in [18], extending it with two heuristics that significantly reduce computational overhead while maintaining high agreement with full Borda rankings across multiple centrality measures, making the method practical for dense or large graphs and addressing both effectiveness and efficiency.

III. Approximate Multi-Centrality Node Ranking

Given a graph $\mathcal{G}$ and a set of centrality measures $\mathcal{C}$ (e.g., degree, beetweenness, closeness, eigenvector, and pagerank centrality), the goal is to determine nodes that are consistently ranked highly across multiple centrality criteria. For each node $v \in \mathcal{V}$, where $\mathcal{V}$ is the set of nodes in $\mathcal{G}$, its rank $\mathcal{R}_v^c$ under centrality measure $c \in \mathcal{C}$ is defined as its position in the descending order of centrality scores (i.e., lower rank indicates higher importance). These ranks are then transformed into scores using the Borda Count method [34]: the node receives a score of $|\mathcal{V}| - \mathcal{R}_v^c$ for each ranking $\mathcal{R}^c$. The overall Borda score of v is obtained by summing these values across all centrality measures, and the k nodes with the highest aggregate scores are selected as the top multi-criteria influential nodes.

This aggregated definition of centrality naturally gives rise to a simple exhaustive evaluation method, denoted as the *Baseline* algorithm (Algorithm 1), which computes all centrality

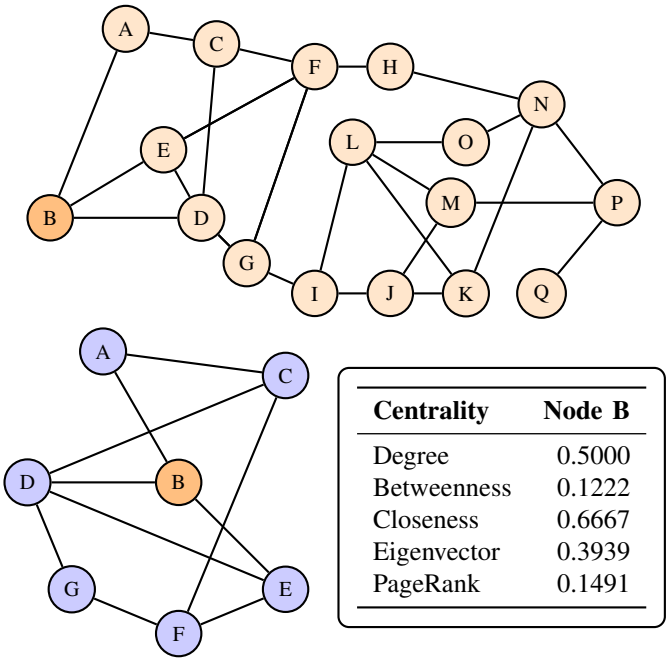
measures on the full graph, ranks each node accordingly, assigns Borda scores based on these ranks, and returns the top k nodes with the highest scores. Given the nature of the Baseline algorithm and the global centrality measure computations it performs, certain graph topologies and configurations may result in a prohibitive time cost for the retrieval of the top multi-criteria influential nodes.

Algorithm 1: Baseline Multi-Criteria Node Ranking

Input: Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, set of centrality measures $\mathcal{C}$, number of nodes k
Output: Set $\mathcal{D}$ of top-k multi-criteria influential nodes

for all $c \in \mathcal{C}$ **do**
    Compute centrality scores for all $v \in \mathcal{V}$ under c
    Generate ranking $\mathcal{R}^c$ in descending score order
for all $v \in \mathcal{V}$ **do**
    $S_v \leftarrow \sum_{c \in \mathcal{C}} (|\mathcal{V}| - \mathcal{R}^c_v)$
Sort all nodes by S_v in descending order
return the top k nodes as $\mathcal{D}$

Fig. 1. Baseline algorithm using Borda Count aggregation across multiple centrality rankings.



Centrality	Node B
Degree	0.5000
Betweenness	0.1222
Closeness	0.6667
Eigenvector	0.3939
PageRank	0.1491

Fig. 2. Top: Input Graph. Bottom left: The 2-hop ego-net of node B. Bottom right: Centralities of node B in its 2-hop ego-net (i.e., its feature vector).

We first provide a brief overview of the approach introduced in [18] for the identification of universally influential nodes that forms the basis of the method proposed in this work (Section III-A). We then present two heuristics: one based on ego-network size thresholding (Section III-B), and another that utilizes selective feature computation by performing centrality calculations only on a sampled subset of nodes (Section III-C). These heuristics significantly reduce the total number of computations, thereby accelerating multi-criteria key actor identification while preserving a high level of agreement with the rankings produced by the baseline method. A dedicated section (Section III-D) outlines how these two heuristics are combined and integrated into the original GCN-based pipeline to operate effectively as a unified approximation scheme.

A. Ego-net Based Multi-Centrality Node Ranking Evaluation

To alleviate this computational cost, the authors in [18] proposed a GCN-based method that utilizes a node's ego-network[1] to estimate the top multi-criteria influential nodes. More specifically, given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, the algorithm (hereafter referred to as the *Ego-net Based* approach) computes local centralities for each node $v \in \mathcal{V}$ within the subgraph induced by its 2-hop ego-network $\mathcal{G}_v^{(2)}$, and constructs a feature vector from the resulting scores. Figure 2 presents an example graph alongside the ego-network of a selected node and its corresponding feature vector.

The feature vectors derived from the ego-net local centralities of all nodes are subsequently used as input to a GCN model. This model is trained on a set of graphs varying in size and topology, where each node is annotated with its global centrality values and its target label corresponds to its true Borda count score. In effect, the model learns to approximate a node's Borda score using only local ego-net information. Since the absolute magnitude of the predicted scores is not important, but rather their relative order, training and evaluation are performed using a margin ranking loss [37], guiding the model to preserve the correct ordering of nodes by multi-criteria influence.

The ego-net based approach often outperforms the baseline in several configurations, by relying solely on local information to approximate multi-criteria influence. However, in densely connected graphs or those with overlapping communities, ego-networks tend to grow rapidly and may encompass a substantial portion of the graph. As a result, the approach becomes impractical for larger graphs due to excessive computational overhead (see Figure 3). This limitation creates the need for a method that keeps ego-net sizes manageable in particular in dense or large-scale networks, offering a speedup over both the Baseline and the Ego-net Based method, while maintaining a tolerable decrease in accuracy.

B. Size-aware Ego-network Retrieval

Ego-net based computation may become prohibitively slow for two main reasons. First, in densely connected graphs, the ego-net of a high-degree node can grow excessively: a 2-hop neighborhood may include a significant fraction of the entire graph, drastically increasing computational overhead. Second, even when ego-nets remain relatively small, computing local centralities for all nodes across all ego-nets can still result in a high cumulative cost.

[1]An ego-network (or ego-net) $\mathcal{G}_v^{(r)}$ of a node v in graph $\mathcal{G}$ consists of all nodes within r hops from v, along with the edges among them.

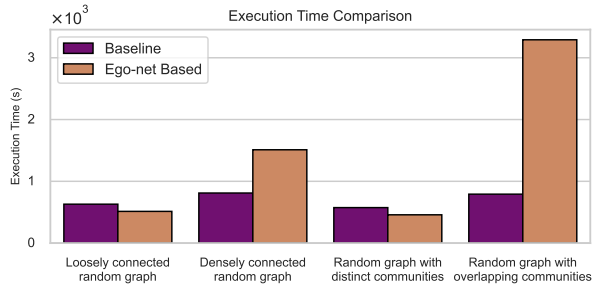


Fig. 3. Execution time comparison between the Baseline and Ego-net Based approaches across different graph types and densities.

To address the former issue, we propose a heuristic that limits the size of each ego-net by introducing a parameter α, which restricts the retrieved ego-network to at most α of its original size. In particular, starting from the central node, the ego-net is traversed in a breadth-first manner until the number of retrieved nodes reaches $\alpha\%$ of the original ego-net size. This results in a more manageable ego-net that still encodes meaningful local structural information around the node.

C. Selective Feature Computation

While limiting ego-net size reduces per-node cost, computing local centralities for all nodes can still result in a substantial cumulative overhead. To address this, we propose a second heuristic that limits the full feature vector construction to only the top $\beta\%$ of nodes ranked by degree centrality. This choice is motivated by its consistent performance, as further examined in RQ5 of our experimental analysis (Section IV-F). For the remaining nodes, we skip the expensive local computations and instead assign a sparse feature vector where only the normalized degree centrality is retained. This allows the model to retain minimal yet structurally relevant information for lower-ranked nodes without incurring significant computational cost.

D. Heuristic Approximation of Multi-Criteria Influence

Bringing together the two heuristics, the proposed approach operates by selectively constructing local feature vectors on size-limited ego-networks and applying a trained GCN to rank multi-criteria node influence (Algorithm 4). Specifically, only the top-$\beta\%$ highest-degree nodes are selected for local analysis, with their 2-hop ego-networks truncated to $\alpha\%$ of their original size to control computation. Local centralities are computed on these induced subgraphs, and used as input features. The remaining nodes are assigned sparse vectors containing only their normalized degree centrality. These features are then processed by the GCN-based model defined in [18] which estimates each node's multi-criteria Borda count score and identifies the top-k most influential nodes accordingly.

IV. Experimental Evaluation

A. Experimental Setup

All experiments were conducted on a workstation equipped with one NVIDIA RTX A6000 GPU (with 48 GB of memory),

Algorithm 2: Approximate Multi-Criteria Node Ranking

Input: $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, centralities $\mathcal{C}$, k, α, β
Output: Top-k multi-criteria influential nodes $\mathcal{D}$

Compute degree d_v for all $v \in \mathcal{V}$
Let $\mathcal{S} \subset \mathcal{V}$ be top $\beta\%$ nodes by d_v
for all $v \in \mathcal{V}$ **do**
  if $v \in \mathcal{S}$ **then**
    Extract $\mathcal{G}_v^{(2)}$ truncated up to $\alpha\%$ nodes via BFS
    Compute local centralities $\mathcal{C}$ on $\mathcal{G}_v^{(2)}$
    $X[v] \leftarrow$ normalized vector of $\mathcal{C}(v)$
  else
    $X[v] \leftarrow [\hat{d}_v, 0, \ldots, 0]$ ▷ $\hat{d}_v$ = Normalized d_v
Predict scores $\hat{s}_v \leftarrow \text{GCN}(X)$ ▷ As defined in [18]
$\mathcal{D} \leftarrow$ top-k nodes by $\hat{s}_v$
return $\mathcal{D}$

Fig. 4. Inference procedure using size-aware and selective ego-net feature construction.

an Intel Xeon Silver 4216 CPU running at 2.10 GHz with 64 physical cores, and 256 GB of system RAM. The system operated under Ubuntu Linux with NVIDIA driver version 535.230.02 and CUDA version 12.2.

B. Implementation details

Regarding the implementation, we keep the training recipe of [18]: the GCN model is trained for 100 epochs with an Adam optimizer and a learning rate of 1×10^{-3} under a pairwise margin ranking loss [37]. The network consists of two GCN layers followed by two fully–connected layers that output the estimated Borda score.

C. Metrics

Accuracy: Accuracy measures the proportion of nodes for which the method correctly predicts key actor status:

$$\text{Accuracy} = \frac{1}{|\mathcal{V}|} \sum_{v \in \mathcal{V}} \mathbf{1}\left[\hat{y}_v = y_v\right], \tag{1}$$

where $y_v \in \{0, 1\}$ indicates whether node v is a ground-truth key actor, $\hat{y}_v \in \{0, 1\}$ is the predicted label.

Normalized Discounted Cumulative Gain (NDCG).: To assess rank quality with position sensitivity, we use NDCG [38]. Given a ranked list, the Discounted Cumulative Gain at cut-off p is

$$\text{DCG}_p = \sum_{i=1}^{p} \frac{2^{rel_i} - 1}{\log_2(i+1)}, \tag{2}$$

where rel_i is the relevance at position i. Normalization by the ideal DCG yields

$$\text{NDCG}_p = \frac{\text{DCG}_p}{\text{IDCG}_p}. \tag{3}$$

Reporting convention. Throughout the experiments we report both measures at an evaluation cut-off, denoted by the "@k" suffix (i.e., Acc@k, NDCG@k), where k equals the number of requested key actors for each α_β configuration.

D. Datasets

We evaluate our approach on both synthetic and real-world datasets. Erdős–Rényi (ER) graphs randomly place edges with probability p, generating random structures that frequently serve as baselines [39]. Networks with power-law degree and community-size distributions are generated by Lancichinetti–Fortunato–Radicchi (LFR) graphs, which control inter-community connectivity via the mixing parameter μ [40].

To complement the synthetic benchmarks, we further evaluate on two real-world social network datasets showcased in Table I: "Deezer-EU", a follower graph representing European users of the Deezer music-streaming platform, containing $28,281$ nodes and $92,752$ edges [41]; and "Soc-douban," a friendship graph from the Douban media platform with approximately $327,162$ nodes and $154,908$ edges [42]. These datasets enable us to assess the practical effectiveness of the proposed approximation on real-world networks in addition to controlled synthetic settings.

TABLE I
DATASET CONFIGURATIONS.

Graph	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{E}\rvert$	avg. deg.
ER (p=1.5×10^{-4})	1.00×10^{5}	$\approx 7.50\times10^{5}$	$\approx 1.50\times10^{1}$
ER (p=2.2×10^{-4})	1.00×10^{5}	$\approx 1.10\times10^{6}$	$\approx 2.20\times10^{1}$
ER (p=2.5×10^{-4})	1.00×10^{5}	$\approx 1.25\times10^{6}$	$\approx 2.50\times10^{1}$
LFR (μ=0.30)	1.00×10^{5}	1.00×10^{6}	2.00×10^{1}
LFR (μ=0.35)	1.00×10^{5}	1.00×10^{6}	2.00×10^{1}
LFR (μ=0.40)	1.00×10^{5}	1.00×10^{6}	2.00×10^{1}
Deezer–EU	2.83×10^{4}	9.72×10^{4}	7.4×10^{0}
Soc–douban	1.54×10^{5}	3.27×10^{5}	2.04×10^{2}

E. Research Questions

We address four key research questions in our experimental analysis: (**RQ1**) How efficient and scalable is the proposed Approximation method? (**RQ2**) How do the two parameters α and β affect the efficiency of the Approximation method? (**RQ3**) How effective is the Approximation method when evaluated on synthetic datasets and how is this effectiveness affected by the ynthetic graph parameters, such as the edge probability p in ER graphs and the mixing parameter μ in LFR graphs? (**RQ4**) How effective is the Approximation method when evaluated on real-world datasets? (**RQ5**) What is the impact of the node–sampling policy used to select the top-β% of nodes for ego-net feature computation on the Approximation method's effectiveness and cost? (**RQ6**) How practically consequential are the key actors identified by the Approximation method?

F. Experimental Results & Discussion

Table II provides the results for the three evaluation metrics for all settings and will be used to answer research questions RQ1 to RQ4. In Table II the Baseline method acts as a point of reference and it defines the ground-truth ranking, so both Accuracy@k and NDCG@k are equal to 1 by definition.

RQ1 Efficiency and scalability of the Approximate method. The execution-time rows of Table II reveal a consistent reduction of *two to three orders of magnitude* relative to the Baseline, on both synthetic and real graphs. This observation holds even for the most dense ER instance and the most strongly mixed LFR instance, indicating that the cost of the proposed method scales more favorably in practice, while the Baseline becomes rapidly more expensive as graphs grow. Hence, with regard to computational efficiency the Approximation method may be considered more preferable.

Despite the significant runtime reductions, the Approximation method preserves most of the Baseline's ranking quality. On synthetic graphs, *Accuracy@k* typically remains within a few percentage points of the Baseline (e.g., ≥ 0.89 already at 5_5 and rising above 0.95 for LFR at 10_20), while *NDCG@k* is consistently high (≥ 0.97 for ER and ≥ 0.98 for LFR across all settings). On real graphs, the drop is more pronounced but still modest: Accuracy stays above 0.81 (Soc–douban) and 0.85 (Deezer) at 10_20, with NDCG between 0.86 and 0.96. Accuracy@k and NDCG@k values generally improve for larger k in the same setting, as ranking errors near the top become less penalizing and broader agreement with the Baseline ranking is easier to achieve.

RQ2 Influence of the parameters α and β. As α_β increases from 5_5 to 10_20, runtime grows by less than one order of magnitude, while ranking quality improves or remains stable. On ER graphs, Accuracy@k either improves or varies slightly (e.g., 0.891–0.942 at *5_5* vs. 0.912–0.930 at 10_20), and NDCG@k increases monotonically (from 0.972–0.992 at 5_5 to 0.985–0.990 at 10_20). On LFR graphs, both metrics improve consistently: Accuracy@k rises from 0.932–0.941 (5_5) to 0.954–0.957 (10_20), while NDCG@k increases from ≈ 0.989–0.991 to ≈ 0.995.

Real-world graphs exhibit a stronger effect: on "Deezer", Accuracy@k increases from 0.678 (5_5) to 0.854 (10_20) and NDCG@k from 0.863 to 0.959, whereas on "Soc–douban" Accuracy@k grows moderately (from 0.818 to 0.849) with NDCG@k remaining high (from 0.948 to 0.958). In summary, 10_10 offers a balanced trade-off, capturing most of the Accuracy/NDCG gains of 10_20 at a fraction of the computational cost, making it suitable for resource-limited settings.

RQ3 Sensitivity to graph density (p) and mixing strength (μ). Across the ER and LFR benchmarks, the proposed method maintains high agreement with the Baseline (NDCG@k near 0.98 and Accuracy@k around 0.90 or higher across configurations) while running in low number of seconds, indicating strong effectiveness on controlled synthetic settings. Fixing the configuration to 10_10 (evaluated at k=10), the execution time varies by less than a factor of ≈ 2.3 across the three ER settings and by ≈ 2.2 across the three LFR settings. Despite these changes, ranking quality remains high: for ER graphs, Accuracy@10 stays within $[0.895, 0.942]$ and NDCG@10 in $[0.977, 0.991]$; for LFR graphs, Accuracy@10 lies in $[0.936, 0.955]$ and NDCG@10 in $[0.981, 0.994]$. Hence, both metrics degrade only marginally, if at all, as density (ER) or mixing (LFR) increases, indicating that the approximation is empirically robust to these structural variations.

RQ4 Effectiveness on real-world networks. The last two columns of Table II show that for the Approximation method

TABLE II
EXECUTION TIME (SECONDS), ACCURACY, AND NDCG FOR ALL SYNTHETIC AND REAL DATASETS.

α_β / **Metric**	**ER graphs**			**LFR graphs**			**Real**	
	p=1.5×10^{-4}	p=2.2×10^{-4}	p=2.5×10^{-4}	μ=3.0×10^{-1}	μ=3.5×10^{-1}	μ=4.0×10^{-1}	Deezer	Soc-doub
5_5 Time	18.5	20.9	22.2	21.7	21.9	22.0	4.6	34.2
5_5 Accuracy@k (k=5)	0.8908	0.9006	0.9418	0.9324	0.9338	0.9408	0.6784	0.8181
5_5 NDCG@k (k=5)	0.9718	0.9809	0.9918	0.9895	0.9894	0.9909	0.8629	0.9483
10_10 Time	19.7	40.2	44.4	22.4	50.1	50.3	5.4	69.8
10_10 Accuracy@k (k=5)	0.7950	0.7914	0.7806	0.9326	0.9378	0.9016	0.6862	0.6530
10_10 Accuracy@k (k=10)	0.8948	0.9154	0.9423	0.9356	0.9548	0.9522	0.8105	0.8220
10_10 NDCG@k (k=5)	0.9204	0.9116	0.9097	0.9856	0.9907	0.9770	0.8549	0.8817
10_10 NDCG@k (k=10)	0.9772	0.9894	0.9909	0.9810	0.9908	0.9944	0.9284	0.9552
10_20 Time	62.8	76.4	82.9	88.2	86.6	90.1	6.9	119.9
10_20 Accuracy@k (k=5)	0.7616	0.818	0.8197	0.9064	0.914	0.9118	0.7279	0.6638
10_20 Accuracy@k (k=10)	0.8655	0.8724	0.8692	0.9315	0.9385	0.9333	0.8013	0.8320
10_20 Accuracy@k (k=20)	0.9123	0.9265	0.9298	0.9543	0.9554	0.9569	0.8536	0.8492
10_20 NDCG@k (k=5)	0.8934	0.9347	0.9353	0.9814	0.9824	0.9817	0.8857	0.8731
10_20 NDCG@k (k=10)	0.9557	0.9599	0.9587	0.9895	0.9907	0.9899	0.9283	0.9581
10_20 NDCG@k (k=20)	0.9847	0.9869	0.9896	0.9951	0.9954	0.9955	0.9585	0.9582
100_100 Time	522.3	1513.8	2562.7	2988.7	3123.8	3217.8	316.2	13676.6
100_100 Accuracy@k (k=5)	0.8395	0.9134	0.925	0.9262	0.9312	0.9304	0.6742	0.8302
100_100 Accuracy@k (k=10)	0.8852	0.9245	0.9412	0.9452	0.9451	0.9472	0.7387	0.8572
100_100 Accuracy@k (k=20)	0.8929	0.944	0.9425	0.9508	0.9509	0.9520	0.8020	0.788
100_100 NDCG@k (k=5)	0.9604	0.9834	0.9892	0.9873	0.9887	0.98869	0.8405	0.9575
100_100 NDCG@k (k=10)	0.9633	0.9889	0.9921	0.9925	0.9927	0.9932	0.8818	0.9680
100_100 NDCG@k (k=20)	0.9838	0.9926	0.9933	0.9946	0.9949	0.9947	0.9226	0.9366
Baseline Time	1077.9	818.9	1370.1	792.3	771.6	807.8	253.4	9204.5
Baseline Accuracy@k	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Baseline NDCG@k	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000

Accuracy@k remains competitive: it increases markedly on Deezer (from 0.678 at 5_5 to 0.854 at 10_20) and stays consistently high on Soc–douban (0.818–0.849). In addition, NDCG@k values obtained are in the range 0.86 to 0.96 with a runtime cost of only a few seconds for Deezer (4.6–6.9 seconds) and a few tens to hundreds of seconds for Soc–douban (34.2–119.9 seconds), far below the corresponding Baseline times (253.4 seconds and 9204.5 seconds, respectively).

RQ5 Impact of node sampling strategy As defined in Section III-C, the proposed approximation computes full ego–net features only for the top-$\beta\%$ of nodes. Here, we justify the choice of degree centrality over alternatives, based on its observed impact on effectiveness. We compare three strategies: (*i*) Degree-based sampling (top–$\beta\%$ by degree centrality) [43], (*ii*) k-core sampling (top–$\beta\%$ by core number) [44], and (*iii*) Random sampling (uniform over nodes). Figure 5 reports Accuracy@k with k=10 under an indicative 10_10 configuration on an LFR graph and two real networks.

Across all datasets in Figure 5, degree-guided sampling consistently attains the highest Accuracy@k, with core-guided sampling a close second on the larger real graph (Soc–douban), and random selection trailing markedly. These results align with the intuition that degree ranking selects structurally prominent vertices that concentrate local information useful to the GCN, while k-core emphasizes globally embedded vertices and thus remains competitive on dense graphs; on the other hand, uniform sampling provides no structural bias and degrades top-k agreement.

RQ6 Practical utility of identified key actors. Assessing the practical utility and structural importance of identified key actors is critical, particularly within the scope of network robustness and dismantling scenarios. Network robustness refers to the resilience of a network to the loss or removal of nodes/edges and has been widely surveyed in the robustness/resilience literature [45]. Network dismantling, in contrast, studies targeted strategies for fragmenting a network efficiently and has its own dedicated comparative surveys and frameworks [46]. These concepts are directly applicable in various real-world contexts, such as disrupting malicious or terrorist networks, impeding misinformation dissemination, or mitigating contagion within social communities.

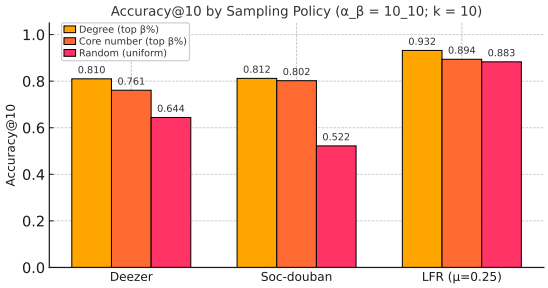


Fig. 5. Accuracy@k (k=10) for three node–sampling strategies (10_10 setting) on Deezer, Soc–douban, and an LFR graph.

To this end, we examine the impact of sequentially removing nodes identified as key actors by our Approximate Multi-Criteria method, compared to state-of-the-art approaches (VoteRank, VoteRank++, EnRenew, IKS, and ECRM) [22]–[26] in the two real datasets. In particular, we measure two key indicators: the size of the Largest Connected Component (LCC) and the number of disconnected components resulting from node removals.

Figure 6 illustrates the results on the Deezer dataset. The LCC size reflects how quickly a network becomes fragmented; rapid decreases indicate that removed nodes held significant structural importance. The Approximate Multi-Criteria method shows effective performance, achieving a notable reduction in LCC size after only a few node removals, thereby indicating that the identified key actors are structurally influential. Moreover, the number of components in the network are noticeably increased upon node removals performed by our approach, reinforcing the effectiveness of our identified nodes in fracturing the network into smaller, isolated subgraphs. This performance suggests the potential applicability of the Approximation method in practical scenarios, such as disrupting illicit networks or hindering coordinated malicious activities in social network contexts.

A similar pattern is observed on the larger *Soc–douban* network (Fig. 7). The initial LCC is about 1.55×10^5 nodes. After removing the first 30 key actors found by our approximate Multi-Criteria method, the LCC falls to roughly 1.538×10^5 , which is a decrease of $\approx 0.7\%$, while the competing methods exhibit only $\approx 0.3\%$ – $\approx 0.5\%$ on the same budget. After 50 removals, our curve reaches about 1.533×10^5 (general reduction subpercent), while the alternatives remain visibly higher (near or above 1.54×10^5). In parallel, the graph fragments increase more sharply under our removals: the number of connected components climbs to approximately 1.5×10^3 after 50 deletions, compared to a substantially smaller increase (roughly 2×10^2–8×10^2) for the baseline methods. Together, these trends indicate that the nodes selected by our approach are structurally more disruptive, yielding faster LCC decay and markedly stronger fragmentation for the same removal budget.

Summary. Collectively, these results show that our method offers significant computational savings while retaining most of Baseline's ranking quality not only on synthetic datasets, but also on real-world graphs. This makes it a fast and practical solution for multi-criteria key actor identification. Taken together, the evidence in Table II substantiates that the Approximation pipeline delivers substantial computational savings while preserving at a significant percentage of the Baseline's ranking quality in all scenarios, thereby satisfactorily answering RQ1–RQ4. Beyond these, **RQ5** indicates that the node–sampling policy matters: selecting the top–$\beta\%$ nodes by degree consistently yields the highest Accuracy@k across LFR and both real graphs (Fig. 5), with k-core a competitive second and uniform random sampling lagging; accordingly, we adopt degree-based sampling as the default. **RQ6** evaluates practical utility via robustness/dismantling tests on Deezer and Soc–douban: removing the key actors identified by our method produces a steeper decline of the largest connected component and a sharper increase in the number of connected components than competing approaches (Figs. 6–7), indicating that the selected nodes are structurally impactful and useful for network-disruption scenarios.

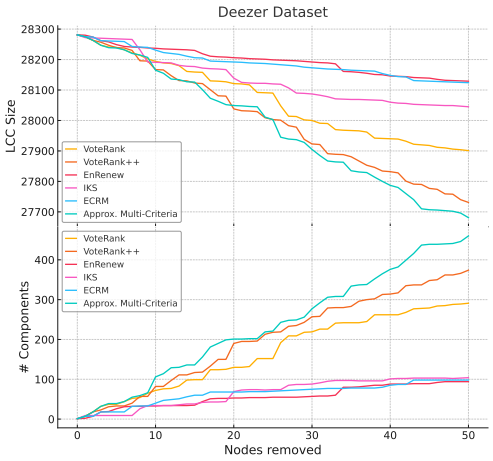


Fig. 6. LCC size and component count versus number of removed key actors for state-of-the-art methods in the Deezer dataset.

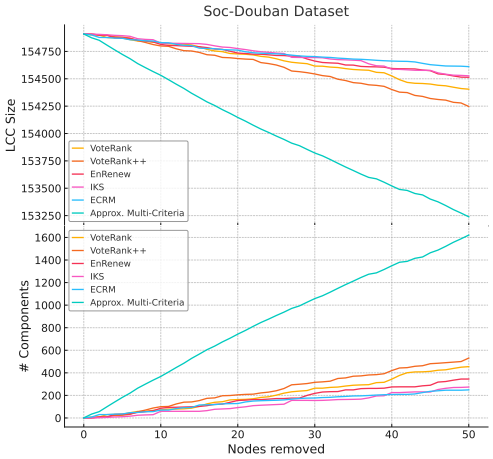


Fig. 7. LCC size and component count versus number of removed key actors for state-of-the-art methods in the soc-Douban dataset.

V. Conclusions and Future Work

In this paper, we have presented a computationally efficient and robust method for identifying influential nodes in complex networks through the approximation of multi-centrality rankings using ego-network information.Using our method, the computational burden inherent to multi-criteria centrality analysis is mitigated With our approach we reduced the computational overhead typically associated with multi-criteria centrality analyses, achieving scalability to large-scale and dense networks. Extensive experiments on both synthetic (ER and LFR) and real-world datasets (Deezer-EU and Soc-douban) validate our approach, demonstrating that our method consistently provides rankings close to the exhaustive baseline while substantially decreasing execution time. Moreover, we empirically validated the practical utility of our identified key actors by examining their impact on network robustness and dismantling. The experimental results indicate that the nodes identified by our approximate method tend to fragment the network more effectively than state-of-the-art approaches for

discovering influential nodes. This highlights our method's potential practical applicability in real-world scenarios, such as effectively disrupting illicit networks or hindering the propagation of misinformation or harmful behaviors. Future work includes further enhancing scalability, investigating adaptive parameter selection strategies, and exploring the method's applicability in dynamic and evolving networks.

REFERENCES

[1] S. Banerjee, M. Jenamani, and D. K. Pratihar, "A survey on influence maximization in a social network," *Knowledge and Information Systems*, vol. 62, no. 9, pp. 3417–3455, 2020.

[2] M. Ma, M. Huang, Y. He, J. Fang, J. Li, X. Li, M. Liu, M. Zhou, G. Cui, and Q. Fan, "Network medicine: A potential approach for virtual drug screening," *Pharmaceuticals*, vol. 17, no. 7, p. 899, 2024.

[3] S. Wandelt, W. Lin, X. Sun, and M. Zanin, "From random failures to targeted attacks in network dismantling," *Reliability Engineering & System Safety*, vol. 218, p. 108146, 2022.

[4] Z. Liu, G. Wan, B. A. Prakash, M. S. Lau, and W. Jin, "A review of graph neural networks in epidemic modeling," in *Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining*, 2024, pp. 6577–6587.

[5] H. Zhang, M. A. Alim, X. Li, M. T. Thai, and H. T. Nguyen, "Misinformation in online social networks: Detect them all with a limited budget," *ACM Transactions on Information Systems (TOIS)*, vol. 34, no. 3, pp. 1–24, 2016.

[6] S. Peng, Y. Zhou, L. Cao, S. Yu, J. Niu, and W. Jia, "Influence analysis in social networks: A survey," *Journal of Network and Computer Applications*, vol. 106, pp. 17–32, 2018.

[7] M. Lalou, M. A. Tahraoui, and H. Kheddouci, "The critical node detection problem in networks: A survey," *Computer Science Review*, vol. 28, pp. 92–117, 2018.

[8] E.-Y. Yu, Y.-P. Wang, Y. Fu, D.-B. Chen, and M. Xie, "Identifying critical nodes in complex networks via graph convolutional networks," *Knowledge-Based Systems*, vol. 198, p. 105893, 2020.

[9] P. Bonacich, "Factoring and weighting approaches to status scores and clique identification," *Journal of mathematical sociology*, vol. 2, no. 1, pp. 113–120, 1972.

[10] L. C. Freeman, "A set of measures of centrality based on betweenness," *Sociometry*, pp. 35–41, 1977.

[11] ——, "Centrality in social networks conceptual clarification," *Social networks*, vol. 1, no. 3, pp. 215–239, 1978.

[12] P. Bonacich, "Power and centrality: A family of measures," *American journal of sociology*, vol. 92, no. 5, pp. 1170–1182, 1987.

[13] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web." Stanford InfoLab, Tech. Rep., 1999.

[14] D. Bucur, "Top influencers can be identified universally by combining classical centralities," *Scientific reports*, vol. 10, no. 1, p. 20550, 2020.

[15] R. Pethes, E. Bodor-Eranus, K. Takács, and L. Kovács, "The core might change anyhow we define it: The instability of key actors in longitudinal social network data," *Complexity*, vol. 2024, no. 1, p. 3956877, 2024.

[16] K. Basu and A. Sen, "Identifying individuals associated with organized criminal networks: A social network analysis," *Social Networks*, vol. 64, pp. 42–54, 2021.

[17] ——, "Epidemiological model independent misinformation source identification." in *ICWSM Workshops*, 2021.

[18] A. Kosmatopoulos, K. Loumponias, O. Theodosiadou, T. Tsikrika, S. Vrochidis, and I. Kompatsiaris, "Identification of key actor nodes: A centrality measure ranking aggregation approach," in *Proceedings of the 2022 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, 2023, p. 125–128.

[19] M. Kitsak, L. K. Gallos, S. Havlin, F. Liljeros, L. Muchnik, H. E. Stanley, and H. A. Makse, "Identification of influential spreaders in complex networks," *Nature physics*, vol. 6, no. 11, pp. 888–893, 2010.

[20] B. Wei, J. Liu, D. Wei, C. Gao, and Y. Deng, "Weighted k-shell decomposition for complex networks based on potential edge weights," *Physica A: Statistical Mechanics and its Applications*, vol. 420, pp. 277–283, 2015.

[21] M. Wang, W. Li, Y. Guo, X. Peng, and Y. Li, "Identifying influential spreaders in complex networks based on improved k-shell method," *Physica A: Statistical Mechanics and its Applications*, vol. 554, p. 124229, 2020.

[22] J.-X. Zhang, D.-B. Chen, Q. Dong, and Z.-D. Zhao, "Identifying a set of influential spreaders in complex networks," *Scientific reports*, vol. 6, no. 1, pp. 1–10, 2016.

[23] P. Liu, L. Li, S. Fang, and Y. Yao, "Identifying influential nodes in social networks: A voting approach," *Chaos, Solitons & Fractals*, vol. 152, p. 111309, 2021.

[24] C. Guo, L. Yang, X. Chen, D. Chen, H. Gao, and J. Ma, "Influential nodes identification in complex networks via information entropy," *Entropy*, vol. 22, no. 2, p. 242, 2020.

[25] A. Sheikhahmadi and M. A. Nematbakhsh, "Identification of multi-spreader users in social networks for viral marketing," *Journal of Information Science*, vol. 43, no. 3, pp. 412–423, 2017.

[26] A. Zareie, A. Sheikhahmadi, M. Jalili, and M. S. K. Fasaei, "Finding influential nodes in social networks based on neighborhood correlation coefficient," *Knowledge-based systems*, vol. 194, p. 105580, 2020.

[27] H. Mo, C. Gao, and Y. Deng, "Evidential method to identify influential nodes in complex networks," *Journal of Systems Engineering and Electronics*, vol. 26, no. 2, pp. 381–387, 2015.

[28] Z. Liu, C. Jiang, J. Wang, and H. Yu, "The node importance in actual complex networks based on a multi-attribute ranking method," *Knowledge-Based Systems*, vol. 84, pp. 56–66, 2015.

[29] T. Bian, J. Hu, and Y. Deng, "Identifying influential nodes in complex networks based on ahp," *Physica A: Statistical Mechanics and its Applications*, vol. 479, pp. 422–436, 2017.

[30] A. Ibnoulouafi, M. El Haziti, and H. Cherifi, "M-centrality: identifying key nodes based on global position and local degree variation," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2018, no. 7, p. 073407, 2018.

[31] Z. Zeng, W. Zhang, and H. Jin, "A "c3-topsis-pareto" based model for identifying critical nodes in complex networks," *Systems*, vol. 13, no. 2, p. 138, 2025.

[32] K. P. Yoon and C.-L. Hwang, *Multiple attribute decision making: an introduction*. Sage publications, 1995.

[33] S. Li, Y. Quan, X. Luo, and J. Wang, "Influential nodes identification for complex networks based on multi-feature fusion," *Scientific Reports*, vol. 15, no. 1, p. 11440, 2025.

[34] T. K. Ho, J. J. Hull, and S. N. Srihari, "Decision combination in multiple classifier systems," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 1, pp. 66–75, 1994.

[35] Z. Tian and L. Chen, "A multi-centrality model based on borda count method for identification of important ports in maritime networks," in *2018 2nd International Conference on Management, Education and Social Science (ICMESS 2018)*. Atlantis Press, 2018, pp. 1351–1357.

[36] A. Madotto and J. Liu, "Super-spreader identification using meta-centrality," *Scientific reports*, vol. 6, no. 1, p. 38994, 2016.

[37] S. K. Maurya, X. Liu, and T. Murata, "Graph neural networks for fast node ranking approximation," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 15, no. 5, pp. 1–32, 2021.

[38] K. Järvelin and J. Kekäläinen, "Cumulated gain-based evaluation of ir techniques," *ACM Transactions on Information Systems (TOIS)*, vol. 20, no. 4, pp. 422–446, 2002.

[39] P. Erdős and A. Rényi, "On random graphs i." *Publicationes Mathematicae*, vol. 6, pp. 290–297, 1959.

[40] A. Lancichinetti, S. Fortunato, and F. Radicchi, "Benchmark graphs for testing community detection algorithms," *Physical Review E*, vol. 78, no. 4, p. 046110, 2008.

[41] B. Rozemberczki and R. Sarkar, "Characteristic functions on graphs: Birds of a feather, from statistical descriptors to parametric models," in *Proceedings of CIKM*, 2020, pp. 1321–1330, dataset available at https://github.com/benedekrozemberczki/Deezer-dataset.

[42] K. Zhou, C. Li, and K. Chen, "Soc-douban: A large-scale chinese social network dataset," Online repository, 2019, https://snap.stanford.edu/data/soc-douban.html.

[43] L. C. Freeman, "Centrality in social networks: Conceptual clarification," *Social Networks*, vol. 1, no. 3, pp. 215–239, 1978.

[44] S. B. Seidman, "Network structure and minimum degree (k-core)"," *Social Networks*, vol. 5, no. 3, pp. 269–287, 1983.

[45] J. P. G. Sterbenz, D. Hutchison, E. K. Çetinkaya, A. Jabbar, J. P. Rohrer, M. Schöller, and P. Smith, "Resilience and survivability in communication networks: Strategies, principles, and survey of disciplines," *IEEE Communications Surveys & Tutorials*, vol. 12, no. 1, pp. 70–92, 2010.

[46] S. Wandelt, X. Shi, X. Sun, and M. Zanin, "From random failures to targeted attacks in network dismantling," *Reliability Engineering & System Safety*, vol. 218, p. 108146, 2022.