Scalable Network Motif Detection: Python-based RAND-ESU with Parallel Execution

Social Network Analysis for Computer Scientists — Course Paper

Ekansh Khanulia
e.khanulia@umail.leidenuniv.nl
LIACS, Leiden University
Leiden, Netherlands

Nazrin Amirova
n.i.amirova@umail.leidenuniv.nl
LIACS, Leiden University
Leiden, Netherlands

Abstract

Network motif detection identifies small, recurring subgraphs that reveal structural principles in complex networks. This report presents complete $k = 3$, $k = 4$, and $k = 5$ results from a revised Python implementation of RAND-ESU aligned with Wernicke (2005) [18] and the mfinder C implementation for ESA [4]. We evaluate four SNAP datasets: Wiki-Vote, Amazon0302, CA-AstroPh, and roadNet-CA. Across three seeds and four datasets (sum over all runs), we sampled 11.0M triads ($k = 3$, $q = 0.1$), 109.4M 4-node subgraphs ($k = 4$, $q = 0.01$), and 191.8M 5-node subgraphs ($k = 5$, $q = 0.0001$). Directed networks exhibit all 13 triad classes with strong star dominance, while undirected networks show only two triad classes at $k = 3$; $k = 5$ reaches up to **6,468** classes in directed networks (mean across seeds). We discuss ESA's inherent sampling bias on directed graphs; our ESA baselines apply Equation (**??**) probability correction so the estimator is unbiased in expectation, and our directed handling uses weak connectivity, which recovers in-star motifs (021U). Significance analysis using degree-preserving random graphs generated by edge swaps reports concentration ratios $C_k^i(G)/\hat{C}_k^i(G)$ (observed concentration divided by the mean concentration in random graphs) as in Wernicke (2005) [18], identifying over-represented motifs, while z-scores can reach extreme values when random-graph variance is small and should be interpreted as qualitative indicators only.

Keywords

Network Motifs, RAND-ESU, ESA, Subgraph Sampling, SNAP Datasets, Unbiased Sampling, Parallel Implementation, Graph Mining

ACM Reference Format:
Ekansh Khanulia and Nazrin Amirova. 2025. Scalable Network Motif Detection: Python-based RAND-ESU with Parallel Execution: Social Network Analysis for Computer Scientists — Course Paper. In *Proceedings of Social Network Analysis for Computer Scientists Course 2025 (SNACS '25)*. ACM, New York, NY, USA, 9 pages.

1 Introduction

Network motifs are small subgraph patterns that occur more often in real networks than in suitable randomized baselines. Introduced by Milo et al. [12], they act as "building blocks" that reflect how different systems are organized. They can be found in regulatory circuits in biology, triadic closure and information flow in social networks, and design hierarchies in technological networks. The core idea is that if a pattern appears far more frequently in a complex network than in randomized networks, it is likely to capture a meaningful organizing principle, and is thus called a network motif.

Milo et al. [12] showed that networks from different domains have distinctive "motif signatures," meaning certain subgraph patterns appear more frequently in some types of networks than others, which can be used to characterize and compare network types based on local connectivity patterns.

Network motif analysis is really helpful. In protein-protein interaction networks, conserved motifs help predict functional relationships between proteins. In the transcriptional network of *Escherichia coli*, specific motifs, such as feed-forward loops, are overrepresented and are known to support temporal gene expression patterns and encode responses to environmental signals.Motif signatures are not limited to biological systems. Networks from different domains can be grouped together into superfamilies based on the motifs they contain, like information-processing networks such as the World Wide Web or social networks like LinkedIn. These differences highlight the importance of building a scalable algorithm capable of analyzing motif patterns in very large networks.

However, finding motifs is challenging because the number of size-k subgraphs grows combinatorially with the size of the network. A graph with n nodes can contain up to $\binom{n}{k}$ distinct subgraphs of size k, making exhaustive enumeration infeasible .Kashtan et al. [3] Edge Sampling Algorithm (ESA) samples subgraphs by picking a random edge and then expanding it by iteratively adding adjacent nodes until k nodes are reached. However, Wernicke [18] identified that ESA has sampling bias because edge expansion is not uniform and over-samples dense, triangle-rich regions while under-sampling patterns. On the other hand, fixing that bias costs even more . Moreover, ESA does not provide a coverage estimate, which means it does not indicate what fraction of all subgraphs has been explored. Additionally, duplicate samples can occur, where the same subgraph is sampled repeatedly, wasting computation.

Wernicke [18] addressed these limitations by introducing ESU (a duplicate-free enumeration algorithm) and RAND-ESU, a randomized variant designed for probabilistic sampling. An advantage of RAND-ESU is that it provides unbiased motif concentration estimates because each subgraph is sampled with equal probability controlled . The original C++ implementation was evaluated on several small biological networks, including E. coli, S. cerevisiae, C. elegans, and the Ythan estuary food web. It was reported to be

SNACS '25, Leiden, the Netherlands

"orders of magnitude faster" than ESA, particularly for subgraph sizes of $k \geq 5$.

Our contribution to the paper is a Python re-implementation of RAND-ESU with several extensions beyond Wernicke (2005) [18]: (i) we implement automated probability scheduling to improve sampling coverage and reduce variance; (ii) we introduce a parallel execution strategy to accelerate ESU expansion and improve load balancing; (iii) we provide a corrected ESA implementation that improves motif detection in directed networks; and (iv) we implement motif significance testing under two degree-preserving baselines: first an edge-swap random-graph ensemble, and second Wernicke's direct Bender-Canfield-based estimator for the expected motif concentration. For each motif class i, we compute and compare concentration ratios $C_k^i(G)/\hat{C}_k^i(G)$ across both baselines to identify over- and under-represented motifs. Finally, we perform cross-domain validation on four diverse SNAP datasets [10]: Wiki-Vote, Amazon0302, CA-AstroPh, and roadNet-CA.

The remainder of this paper is organized as follows. Section 2 reviews prior work on network motifs, sampling algorithms, and significance testing, with emphasis on ESA, ESU, and RAND-ESU. Section 3 introduces the graph notation, motif definitions, and concentration measures used throughout the paper, along with background on motif class enumeration and directed triad classification. Section 4 presents our methodology, including the RAND-ESU implementation, automated probability scheduling, parallel execution strategy, and significance testing framework. Section 5 describes the SNAP datasets used for evaluation, covering their structure and domain characteristics. Section 6 reports the experimental setup and results, including sampling statistics, motif diversity, significance analysis, and runtime performance.FInally Section 7 conclude the paper.

2 Related work

Network motifs were introduced by Milo et al. [12] as small subgraph patterns that occur more frequently in real networks than in suitable randomized counterparts. Subsequent work showed that motif frequencies can also be influenced by global structural properties and null-model selection, emphasizing the importance of appropriate baselines while interpreting significant motifs [1, 17]. Motif analysis has since been applied across domains, including biological and social networks, to characterize local connectivity patterns.

To estimate motif concentrations without exhaustive enumeration, Kashtan et al. [3] proposed the Edge Sampling Algorithm (ESA), which samples subgraphs through edge-based expansion. However, ESA is inherently non-uniform and requires bias correction, which can become computationally expensive when we are dealing with larger subgraph sizes. Additionally, its edge-expansion strategy must treat directed graphs as weakly connected during sampling to avoid systematically missing certain motif classes [18].

Wernicke [18] introduced ESU, an algorithm that enumerates each size-k subgraph exactly once using lexicographic constraints, and RAND-ESU, a randomized variant that applies depth-wise probabilistic pruning. RAND-ESU visits each subgraph with equal probability $\prod_{d=1}^{k} p_d$, yielding an unbiased estimator for motif concentrations:

$$\hat{C}_k^i(R,G) = \frac{|\{G' \in R : G' \in S_k^i(G)\}|}{|R|}. \tag{1}$$

This approach was shown to be substantially faster than ESA for $k \geq 5$ on biological networks [18].

Motif significance is commonly assessed using degree preserving null models. Wernicke [18] describes both edge-swap random graphs, which are widely used but computationally costly, and approaches such as the Bender Canfield method, which avoid random graph generation at the cost of more involved mathematics. He further recommends understanding significance via concentration ratios $C_k^i(G)/\hat{C}_k^i(G)$ rather than z-scores, since z-scores can become unstable when variance across random graphs is very small. We follow this convention in our analysis.

We refer to two C implementations for our own implementation. The mfinder software [4] from the Alon Lab provides a reference implementation of ESA and degree-preserving edge-swap randomization. FANMODPlus [5] provides a C++ implementation of RAND-ESU with the ESU-tree structure. Key implementation parameters including directed neighbor handling, mutual-edge preservation during swaps, and switch factors were aligned to these codebases by code inspection to ensure methodological consistency.

3 Preliminaries

In this section, we introduce the notation and core definitions used throughout the paper. We first describe the graph model and neighborhood operators, including the exclusive-neighborhood construction . We then define size-k induced subgraphs, motif classes based on isomorphism, and motif concentration.We will also define the algorithm used (RAND-ESU and ESU) in the entire process, and also talk about motif significance.Finally, we summarize how the number of motif classes grows with increasing k in our datasets and review the directed triad census (M A N notation) used to label and interpret $k = 3$ motifs.

3.1 Graph Notation and Definitions

Let $G = (V, E)$ be a graph with $n = |V|$ vertices and $m = |E|$ edges. For directed graphs, $E \subseteq V \times V$, while for undirected graphs E consists of unordered vertex pairs. The neighborhood of a vertex v is defined as

$$N(v) = \{u \in V : (v,u) \in E \text{ or } (u,v) \in E\}. \tag{2}$$

For a vertex set $V' \subseteq V$, we define its neighborhood as

$$N(V') = \bigcup_{v \in V'} N(v) \setminus V'. \tag{3}$$

Following Wernicke [18], we assume that vertices are uniquely labeled $1, \ldots, n$, and we write '$u > v$" as shorthand for 'label$(u) >$ label(v)." The *exclusive neighborhood* of a vertex v with respect to a vertex set V' is defined as

$$N_{\text{excl}}(v, V') := \{u \in N(v) : u \notin N(V') \cup V'\}. \tag{4}$$

This construction is very essential for duplicate-free enumeration, as it guarantees that each subgraph is visited exactly once.

Definition 3.1 (Size-k Induced Subgraph). A size-k induced subgraph of G is defined by a vertex set $V' \subseteq V$ with $|V'| = k$, together with all edges in E whose endpoints both lie in V'.

Definition 3.2 (Motif Class). Two subgraphs belong to the same motif class if they are *isomorphic*, meaning there exists a bijection between their vertex sets that preserves adjacency relations. For a graph G, we denote by $S_k^i(G)$ the set of all size-k subgraphs that belong to motif class i.

Definition 3.3 (Concentration). The concentration of motif class i in a graph G is defined as

$$C_k^i(G) = \frac{|S_k^i(G)|}{\sum_j |S_k^j(G)|}, \tag{5}$$

that is, the fraction of all size-k subgraphs in G that belong to class i.

3.2 Problem definition

Our objective is to make large-scale network motif analysis practically feasible. Given a graph G and a subgraph size k, we aim to find the distribution of motif classes in G without exhaustively enumerating all size-k induced subgraphs. Because the number of such subgraphs grows combinatorially with network size, direct enumeration becomes computationally impossible even for moderate k in real-world graphs.We therefore focus on sampling-based estimation of motif frequencies. The estimates should accurately reflect the structural patterns present in the network and should not bias the results toward particular motif types. At the same time, the estimation process must scale to large graphs: computation should be controlled by sampling parameters rather, and memory usage should remain modest beyond storing the graph and aggregate motif statistics.

In addition, we also determine whether observed motifs are structurally significant. This requires comparing the motif distribution in G against distributions obtained from degree-preserving randomized graphs, providing a baseline to identify motifs that are over or under represented.

3.3 ESU and RAND-ESU

We now introduce the two core procedures ESU and RAND-ESU . These methods are introduced as alternatives to ESA that retain coverage guarantees while scaling better to large graphs.Wernicke [18] replaces ESA with a randomized enumeration. He first stated ESU, an algorithm that traverses the connected size-k subgraphs in a duplicate-free manner using the label ordering ($u > v$) and the exclusive-neighborhood N_{excl} which restricts how subgraphs may be constructed during enumeration. These constraints define an ESU-tree structure in which each leaf corresponds to a unique size-k subgraph. As a result, ESU outputs every connected size-k subgraph of G exactly once.

But ESU is a bit too slow on large graph, therefore Wernicke [18] stated RAND-ESU ,which is obtained by probabilistically pruning the ESU-tree.Probabilistic pruning means that at each step of a tree traversal , you randomly decide to stop exploring a branch with some random probability. Specifically, for each $d \in \{1, \ldots, k\}$ we choose $p_d \in (0, 1]$, and an expansion step at depth d is executed with probability p_d . This depth-wise probabilistic pruning implies that any particular leaf of the ESU-tree is reached with probability

$$\Pr[\text{a given size-}k\text{ subgraph is sampled}] = \prod_{d=1}^{k} p_d,$$

so all size-k subgraphs have the same sampling probability and the resulting samples are unbiased [18].These unbiased samples can then be used to estimate motif-class frequencies (and hence concentrations) by counting how often each class appears in the sample.In section 4 ,we will go in depth about how ESU and RAND-ESU works.

3.4 Motif significance

Beyond estimating motif concentration in graph G,motif analysis typically asks whether a motif class is over or under represented relative to degree-preserving null model. Following Wernicke [18], motif significance can assessed by comparing the observed concentration $C_k^i(G)$ of motif class i to its expected concentration under a degree-preserving null model.We report the concentration ratio as

$$\rho_i = \frac{C_k^i(G)}{\widehat{C}_k^i(G)},$$

where $\widehat{C}_k^i(G)$ denotes the expected (null) concentration. Values $\rho_i > 1$ indicate over-representation and $\rho_i < 1$ indicate under-representation.

We can get this expected concentration in two ways.One by using Edge-swap ensemble baseline , which approximates $\widehat{C}_k^i(G)$ by generating an numerous degree-preserving randomized graphs via repeated edge swaps and averaging the resulting motif concentrations. However Wernicke [18] notes that this can be computationally expensive and requires many swaps per graph.So he uses Direct degree-sequence baseline (Bender and Canfield) to avoid generating randomized graphs. Wernicke also considers a direct computation of expected motif frequencies under the space of graphs with the same degree sequence, using counting results due to Bender and Canfield. This provides a degree-preserving reference without constructing an explicit random-graph ensemble.

3.5 Triad Census for Directed Networks

For $k = 3$ directed graphs, the 13 possible triads are classified using the M-A-N notation (Mutual, Asymmetric, Null) [2]. Table 1 shows the relation between canonical signatures and triad labels used throughout this paper.

4 Approach

4.1 Original Method: ESU,RAND-ESU and Siginificane Testing

ESU (Algorithm 1) enumerates all connected size-k subgraphs by combining a fixed root order with an exclusive-extension rule. For each root vertex v, ESU grows a partial vertex set V_{sub} using only candidates u with $u > v$, so the same vertex set cannot be regenerated from a smaller-labeled root. During recursion it maintains a candidate set V_{ext}; when a vertex $w \in V_{\text{ext}}$ is added, the candidates are updated by adding only those new neighbors of w that satisfy the root constraint and lie in the exclusive neighborhood (via N_{excl}),

Table 1: Triad census for connected $k = 3$ directed networks. Signatures are 9-bit adjacency matrix encodings in canonical form.

Signature	Label	Description
000100100	021U	In-star: two nodes point to third
000000110	021D	Out-star: one node points to two
000001100	021C	Chain: $0 \longrightarrow 1 \longrightarrow 2$
001001010	111D	Mutual pair + outgoing edge
000001110	111U	Mutual pair + incoming edge
001101100	120D	Feed-forward loop (down)
000101110	120U	Feed-forward loop (up)
001100110	120C	Cycle with mutual edge
000100110	030T	Transitive triad
001100010	030C	Directed 3-cycle
001001110	201	Two mutual dyads sharing a node
001101110	210	Mutual pair + two asymmetric edges
011101110	300	Complete (all 6 directed edges)

Algorithm 1 ENUMERATESUBGRAPHS(G, k) (ESU)

```
Require: Graph $G = (V, E)$, integer $1 \le k \le |V|$
Ensure: All size-$k$ subgraph in $G$
for all $v \in V$ do
    $V_{\text{ext}} \leftarrow \{ u \in N(\{v\}) \mid u > v \}$
    ExtendSubgraph$(\{v\}, V_{\text{ext}}, v, k)$
end for
procedure ExtendSubgraph$(V_{\text{sub}}, V_{\text{ext}}, v, k)$
    if $|V_{\text{sub}}| = k$ then
        output $G[V_{\text{sub}}]$
        return
    end if
    while $V_{\text{ext}} \neq \emptyset$ do
        choose and remove some vertex $w$ from $V_{\text{ext}}$
        $V'\text{ext} \leftarrow V\text{ext} \cup \{ u \in N_{\text{excl}}(w, V_{\text{sub}}) \mid u > v \}$
        ExtendSubgraph$(V_{\text{sub}} \cup \{w\}, V'_{\text{ext}}, v, k)$
    end while
end procedure
```

i.e., they are not already available through previously chosen vertices. When $|V_{\text{sub}}| = k$, ESU outputs $G[V_{\text{sub}}]$ and backtracks.Since this traversal is computationally exhaustive we use RAND-ESU.

As discussed in section 3.3 RAND-ESU samples connected size-k subgraphs by probabilistically pruning the ESU-tree , but we should also discuss how to choose $\{p_d\}$ to achieve a desired sampling efforts. A common constraint is to target an expected sampling fraction $q \in (0, 1)$ by setting $\prod_{d=1}^{k} p_d = q$.

As a general rule, the parameters p_d should be chosen larger for smaller depths and decrease as d increases, as long as the amortized runtime per sample remains acceptable.This tends to reduce sampling variance, encourages exploration across many regions of the graph, and keeps the number of visited leaves manageable [18].Compared to edge-sampling (ESA), RAND-ESU requires choosing sampling parameters and controls only the expected number of samples. In return ,it samples size k subgraphs without bais and avoids the costly bias-correction step required by ESA;setting $p_d = 1$ for all depths recovers deterministic ESU.

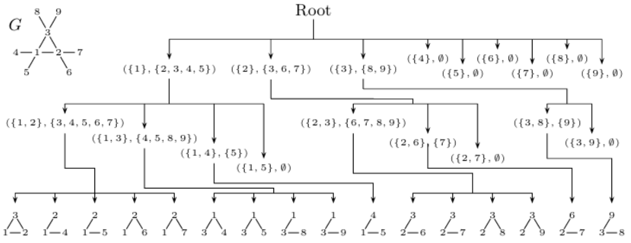

Figure 1: ESU-tree for enumerating size-3 subgraphs in graph G . Each internal node contains $(V_{\text{sub}}, V_{\text{ext}})$ pairs. The 16 leaves at depth 3 correspond to the 16 unique size-3 subgraphs in G. RAND-ESU samples branches with probability p_d at each depth d. Figure from Wernicke [18].

Having estimated motif concentrations $C_k^i(G)$, we next evaluate whether these motifs are structurally significant under a degree-preserving null model. As already discussed about concentration ratio ρ_i and the two methods used to estimating the expected concentration $\widehat{C}_k^i(G)$ in Section 3.4 , we now describe in detail how these two method works.This distinction is important in our evaluation ,where these two methods are compared on multiple real world networks. In Edge-swap ensemble methods we generate a collection $\{G^{(1)}, \ldots, G^{(B)}\}$ of degree-preserving randomized graphs using double-edge swaps. Each randomized graph is obtained by applying T valid swap moves starting from G . For each b, we estimate $C_k^i(G^{(b)})$ using the same subgraph sampling pipeline as for G. The expected concentration is then approximated by the sample mean

$$\widehat{\mu}k^i = \frac{1}{B}\sum b = 1^B C_k^i(G^{(b)}),$$

(and optionally the sample variance $\widehat{\sigma}_k^{i\,2}$ if we also report z-scores). We finally report $\rho_i = C_k^i(G)/\widehat{\mu}_k^i$.

In case of direct degree sequence estimator (Bender-Canfield) we approximate $\widehat{C}_k^i(G)$ via Monte Carlo sampling of vertex k-sets $\{v_1, \ldots, v_k\} \subseteq V$. For each sampled k-set, we compute the number of degree-sequence realizations in which the induced subgraph belongs to motif class i, using the counting formulas due to Bender and Canfield. Adding these counts over samples gives us an estimate of $\widehat{C}_k^i(G)$, which we use in computing concentration ratio ρ_i.

4.2 Automated Probability Scheduling

As already mentioned , choosing the depth probabilities p_d is crucial .They should be high for small depths and decrease as d grows to balance coverage and runtime. We use the "fine" schedule from Wernicke (2005) [18] that meets a target total sampling probability q: $p_d = 1.0$ for $d < k$ and $p_k = q$. This ensures all root nodes and early extensions are considered, with probabilistic pruning only at the final depth. Table 2 lists example values.

4.3 Algorithm Implementation

Algorithm 2 our implementation logic. We use a Bernoulli selection process where each candidate child is selected independently with

Table 2: Probability schedules for different (k, q) configurations using the fine schedule.

Config	q	$p_{\mathbf{depth}}$
$k = 3$	0.1	$[1.0, 1.0, 0.1]$
$k = 4$	0.01	$[1.0, 1.0, 1.0, 0.01]$
$k = 5$	0.0001	$[1.0, 1.0, 1.0, 1.0, 0.0001]$

Algorithm 2 RAND-ESU with Automated Scheduling

```
Require: Graph $G = (V, E)$, size $k$, sampling fraction $q$
Ensure: Sampled subgraphs with unbiased concentration estimates
Compute $p_1, \dots, p_k$ from $q$ using fine schedule
for each vertex $v \in V$ do
    if Random() $\leq p_1$ then
        $V_{\text{ext}} \leftarrow \{u \in N(v) : u > v\}$
        ExtendSubgraph($\{v\}, V_{\text{ext}}, v, 1$)
    end if
end for

procedure ExtendSubgraph($V_{\text{sub}}, V_{\text{ext}}, v_{\min}, d$)
    if $|V_{\text{sub}}| = k$ then
        yield $G[V_{\text{sub}}]$
        return
    end if
    while $V_{\text{ext}} \neq \emptyset$ do
        Remove arbitrary $w$ from $V_{\text{ext}}$
        if Random() $\leq p_{d+1}$ then
            $V'_{\text{ext}} \leftarrow V_{\text{ext}} \cup \{u \in N_{\text{excl}}(w, V_{\text{sub}}) : u > v_{\min}\}$
            ExtendSubgraph($V_{\text{sub}} \cup \{w\}, V'_{\text{ext}}, v_{\min}, d + 1$)
        end if
    end while
end procedure
```

probability $p_{|V_{\text{sub}}|+1}$ at the current recursion depth (i.e., the depth-wise schedule $\{p_d\}$).Implementation details mirror the reference RAND-ESU logic. For directed graphs, ESU/RAND-ESU expansion uses weak connectivity (predecessors ∪ successors) so that in-star motifs remain reachable, while motif classification preserves the original edge directions. The exclusive neighborhood follows the corrected definition from Section 3 to avoid duplicate enumeration paths. Each sampled subgraph is mapped to a canonical adjacency signature by enumerating all $k!$ node permutations and selecting the lexicographically smallest adjacency string; this is feasible for small k (typically $k \leq 5$) and matches the canonical labeling used in mfinder/FANMOD. For larger k, we optionally use a memory-lean counting path that aggregates signature counts per worker instead of storing all subgraphs.

4.4 Parallel Execution Strategy

We also implement parallelization to reduce RAND-ESU runtime on large graph.Parallelization of recursive graph algorithms is challenging due to the uneven computational cost of different subtrees. We implement a dynamic chunking strategy using Python's multiprocessing library. First, we determine the list of valid root nodes (optionally filtering/sampling them); then we divide this list into chunks, creating 4× more chunks than available CPU cores to improve load balancing as workers finish at different times. Each worker reconstructs the graph from a serialized representation (node/edge lists) to avoid the high overhead of pickling NetworkX objects. Finally, results are aggregated using Python's Counter for efficient motif signature counting to count signatures efficiently without storing all sampled subgraphs.

4.5 Significance Testing

To assess motif significance, we compare observed concentrations $C_k^i(G)$ against mean concentrations in degree-preserving random graphs $\hat{C}_k^i(G)$ generated by edge swaping [18]. Throughout this section, we report the concentration ratio $C_k^i(G)/\hat{C}_k^i(G)$.

In edge swap method ,our approach generates degree-preserving random graphs using mfinder-style edge swaps. For directed graphs, mutual edges are swapped separately to preserve reciprocity, matching mfinder defaults; for undirected graphs the switch factor is 10 (not used in our significance runs, which are directed only).We fully enumerate $k = 3$ triads in each randomized graph to remove sampling variance. We generate 100 randomized graphs via edge swaps (100 swaps per edge for directed graphs), run full enumeration of $k = 3$ triads on each randomized graph, and then compute concentration ratios as

$$\text{Ratio}i = \frac{C_k^i(G)}{\hat{C}_k^i(G)} = \frac{C\text{orig}^i}{\mu_{\text{rand}}^i} \tag{6}$$

We report p-values as the fraction of randomized graphs whose concentration exceeds the observed value. These values are descriptive and are included alongside ratios to indicate how often a motif appears above the original concentration under the degree-preserving null.It should be however noted that Wernicke (2005) [18] uses 10,000 random graphs for significance testing. Due to computational constraints, we use 100 random graphs. This deviation is explicitly documented; while 100 graphs provide stable mean estimates, variance estimates may be less precise.

We also implement the direct Bender-Canfield expectations from Wernicke (2005) Section 3 as a cross-check for directed triads. For directed graphs with in- and out-degree sequences, the asymptotic count of graphs with the same degree sequence is

$$N(\mathbf{d}_{\text{in}}, \mathbf{d}_{\text{out}}) \approx \frac{m!}{\prod_i d_i^{\text{out}}! \prod_i d_i^{\text{in}}!} \exp(-\lambda), \quad \lambda = \frac{\sum_i d_i^{\text{out}} d_i^{\text{in}}}{m}. \tag{7}$$

Fixing a triad pattern on vertices (v_1, v_2, v_3) reduces degrees to $\mathbf{d}'$ and edges to $m' = m - |E_{\text{triad}}|$, so the expected weight of that pattern is proportional to $N(\mathbf{d}')/N(\mathbf{d})$. We compute this ratio incrementally from the local degree changes, sample $T = 100{,}000$ random vertex triples, and estimate the expected concentration as the ratio of total pattern weight to the total weight of connected triads (Eq. 3 in [18]). Observed concentrations are obtained by full enumeration of connected triads when possible. The reported runs include the λ correction and are used only for directed $k = 3$.

We report concentration ratios as the primary significance metric rather than using z-scores. The z-score is defined as:

$$Z_i = \frac{C_{\text{orig}}^i - \mu_{\text{rand}}^i}{\sigma_{\text{rand}}^i} \tag{8}$$

The problem here is when the standard deviation σ_{rand}^i is extremely small,which occurs when motifs are rare in degree-preserving

Table 3: Dataset characteristics (processed graph counts; independent of sampling seeds).

Dataset	Nodes	Edges	Dir.
Wiki-Vote	7,115	103,689	Yes
Amazon0302	262,111	1,234,877	Yes
CA-AstroPh	18,771	198,050	No
roadNet-CA	1,965,206	2,766,607	No

random graphs.Even modest concentration differences yield z-scores in the thousands or millions, which are unfathomable quantitatively. Concentration ratios, by contrast, directly measure how much more frequently a motif appears in the real network versus the random baseline. A ratio of 100 has clear meaning: the motif is 100 times more common than expected. We therefore report z-scores only as a secondary diagnostic, treating values above a threshold (e.g., $|Z| > 2$) as indicating significance without interpreting their precise magnitude.

5 Datasets

We use four SNAP datasets [10] from different domains, covering e-commerce, scientific collaboration, physical infrastructure, and social voting. Compared to Wernicke's [18] datasets, they are substantially larger and more diverse.Next we will provide more details about the dataset. The networks span from 7,115 to 1,965,206 nodes and represent fundamentally different domains. For CA-AstroPh and roadNet-CA, the SNAP files list each undirected edge twice.So we load them as undirected graphs and report the number of unique undirected edges .(Table 3).[1]

The Wiki-Vote dataset : This dataset [7, 8, 16] records Wikipedia admin election votes as a directed network. A directed edge from user A to user B indicates that A voted for B.

Amazon0302 - Product Co-Purchasing Network : This dataset [6, 13] is an Amazon co-purchasing network (March 2003), where an edge $itoj$ means "customers who bought i also bought j."

CA-AstroPh - Scientific Collaboration Network : This dataset [9, 15] is an arXiv Astro Physics co-authorship network (1993 -2003), where undirected edges link co-authors.

RoadNet-CA - California Road Network : This dataset [11, 14] models California roads. Nodes are intersections and edges are road segments. The low triangle count reflects how roads are segmented in the dataset: many potential topological triangles (e.g., three-way intersections) are broken into chains of intermediate nodes, creating open triads rather than closed triangles. This is a dataset representation artifact rather than a geometric constraint, as planar graphs can and do contain many triangles.

6 Experiments and Results

In this section, we run experiments to evaluate our Python implementation of RAND-ESU and the corrected ESA baseline on the four SNAP datasets from Section 5. We study sampling behavior and motif-class diversity for different k values, report detailed $k = 3$ motif distributions per dataset, and assess directed-triad significance . We first describe the experimental configuration and setup in Section 6.1, then summarize sampling statistics and motif diversity in Section 6.2. Finally, we report significance and baseline comparisons in Section 6.3 and runtime/scalability results in Section 6.4.

Table 4: Average RAND-ESU samples per run (mean across 3 seeds).

Dataset	k=3	k=4	k=5
Wiki-Vote	1,332,978	25,134,149	51,975,215
Amazon0302	768,368	1,947,472	1,075,664
CA-AstroPh	1,004,974	9,238,533	10,887,742
roadNet-CA	575,514	139,324	3,704

6.1 Experimental Setup

All experiments used the fine probability schedule with Bernoulli child selection:

- $k = 3$: $q = 0.1$ (10% sampling fraction)
- $k = 4$: $q = 0.01$ (1% sampling fraction)
- $k = 5$: $q = 0.0001$ (0.01% sampling fraction)

Significance analysis was performed only for directed graphs at $k = 3$, consistent with Wernicke (2005) [18] and its focus on directed triads. Summary tables report means across three RAND-ESU seeds; detailed $k = 3$ tables report seed 1 counts/concentrations. Significance uses mean original concentrations (3 seeds) and mean random-graph concentrations (100 graphs), and direct Bender-Canfield uses full enumeration for observed concentrations with a single $T = 100{,}000$ run (seed 1) for expectations. All experiments were executed via Jupyter notebook, which orchestrates the scripts in the src/ directory. We ran the full pipeline in Google Colab on a TPU v6e-1 environment with 44 CPU cores and 175 GB of system RAM.

6.2 Results Overview

Table 4 summarizes the average samples per run (mean across 3 seeds), with counts varying strongly by network structure.

Across all datasets and seeds, total samples are 11,045,503 ($k = 3$), 109,378,437 ($k = 4$), and 191,826,976 ($k = 5$). Wiki-Vote yields the most samples due to its dense connectivity, while roadNet-CA yields the fewest due to sparse, near-planar structure.

Our results highlight the combinatorial explosion in directed motifs. As shown in Table 5, undirected graphs discover only 2 classes at $k = 3$ (open triads and triangles), while directed graphs discover all 13 possible triad classes. At $k = 5$, directed networks reach up to 6,468 unique classes (mean across seeds).

6.3 Motif Distribution by Dataset

In this section we report the $k = 3$ motif distribution for each dataset and discuss the dominant patterns.

In Wiki-Vote network (Table 6) the distribution is dominated by asymmetric out-star patterns reflecting the voting structure. Out-stars (021D) dominate at 43.471), reflecting highly active voters who vote for many candidates. In-stars (021U) are 24.25% (seed 1),

[1] This is why the raw headers show 396,160 edges for CA-AstroPh and 5,533,214 for roadNet-CA, while the processed graphs report 198,050 and 2,766,607, respectively.

Table 5: Unique motif classes discovered by k (mean across three RAND-ESU seeds; rounded to nearest whole unless a decimal is shown)

Dataset	Type	k=3	k=4	k=5
Wiki-Vote	Directed	13	199	6,468
Amazon0302	Directed	13	196	1,411
CA-AstroPh	Undirected	2	6	21
roadNet-CA	Undirected	2	5.67	12

capturing candidates receiving votes from multiple users. Complete triangles here mean the directed triad 300 (all six directed edges); this is extremely rare (0.016seed 1), consistent with hierarchical, non-reciprocal voting. This is distinct from undirected triangle counts reported in SNAP metadata.

Table 6: Top k=3 motif concentrations in Wiki-Vote (RAND-ESU seed 1).

Rank	Triad	Count	Conc.	Description
1	021D	579,557	43.47%	Out-star
2	021U	323,255	24.25%	In-star
3	021C	274,601	20.60%	Chain
4	111U	56,001	4.20%	Mutual + in
5	030T	46,532	3.49%	Transitive

The dominant out-star motif (43.47% in seed 1) represents voters who vote for multiple candidates. The 24.25% in-star concentration (seed 1) represents candidates who receive votes from multiple voters. The rare complete triangles (300, only 0.016% in seed 1) indicate that fully reciprocal voting is uncommon.Across three seeds, RAND-ESU estimates are highly stable. The 021D concentration is 43.47%, 43.44%, and 43.50% for seeds 1, 2, and 3 (std dev = 0.02%).

In Amazon0302 network (Table 7) the distribution is dominated by in-star patterns reflecting the recommendation system's hub-and-spoke structure.It suggests popular products that receive many "customers also bought" links from other products. Complete triads (300) have edge-swap ratios of 71,726×, with 120D at 26,759× and 210 at 20,419× (edge-swap means across seeds), indicating strong reciprocal and feed-forward structure relative to degree-preserving random graphs.

Table 7: Top k=3 motif concentrations in Amazon0302 (RAND-ESU seed 1).

Rank	Triad	Count	Conc.	Description
1	021U	372,125	48.45%	In-star
2	111D	129,139	16.81%	Mutual + out
3	021C	79,508	10.35%	Chain
4	111U	56,504	7.36%	Mutual + in
5	021D	32,694	4.26%	Out-star

In the undirected networks, only two motif classes exist open triads (wedges) and closed triads (triangles) Table 8.13.44% oftriangles in CA-AstroPh in seed 1, reflects strong triadic closure in co-authorship. In contrast, roadNet-CA is 2.10% triangles in seed 1 and rest is open triads , reflects how road intersections are represented in the data .Many potential three-way junctions are subdivided into sequences of intermediate nodes, breaking topological triangles into open triads.

Table 8: k=3 motif distributions in undirected networks (RAND-ESU seed 1).

Dataset	Motif	Count	Conc.
CA-AstroPh	Open triad	870,094	86.56%
	Triangle	135,099	**13.44%**
roadNet-CA	Open triad	564,616	97.90%
	Triangle	12,087	**2.10%**

6.4 Significance and Method comparison

In this section, we report motif significance under degree-preserving baselines using concentration ratios as the primary metric, comparing an edge-swaping method with Bender-Canfield expectations for directed $k = 3$. We then compare RAND-ESU against ESA to validate sampling behavior .

We use edge-swap random graph method for Wiki-Vote and Amazon0302 network with 100 randomized graphs per dataset. Original concentrations are means across three RAND-ESU seeds whilerandom means and p-values are computed across the 100 randomized graphs. Tables 9 and 10 report a compact selection: the strongest ratios, plus the motifs used in the edge-swap-vs-direct comparison (021U, 030T, 300), ensuring similarity with Bender–Canfield tables.In Wiki-Vote Network The complete triad (300) has a concentration ratio of 3.49× over the random mean , while feed-forward patterns (120D, 120U) show roughly 2× ratios.In Amazon0302 Network ,concentration ratios are higher because random baseline concentrations are very small. Complete triads (300) have edge-swap ratios of 71,726× (mean across seeds and 100 graphs), indicating substantially more reciprocal, densely connected product clusters than under degree-preserving rewiring.

It should be noted that for Amazon0302 z-scores reach magnitudes of 10^6 to 10^9 because random-graph standard deviations are on the order of 10^{-7} for rare triads. But these values should not be interpreted as quantitatively meaningful beyond "highly over-represented." This is exactly why Wernicke (2005) [18] emphasize using concentration ratios. Undirected networks on other hand show limited motif diversity. CA-AstroPh and roadNet-CA have only two triad classes, with triangles at 13.44% (seed 1) and 2.10% (seed 1), respectively.

Tables 11 and 12 report selected motifs under the direct Bender–Canfield estimator ($T = 100{,}000$ sampled triples with λ correction, seed 1), with observed concentrations obtained by full enumeration of connected triads. Table 13 compares the edge-swap concentration ratios (mean across three RAND-ESU seeds and 100 graphs) against the corresponding Bender–Canfield ratios; both methods agree on which motifs have ratios above one, while ratio magnitudes diverge when expected concentrations are extremely small.

Next we compare RAND-ESU and ESA on directed datasets focusing on triad coverage and $k = 3$ concentration estimates. After correcting ESA to use weak connectivity for directed graphs, both

Table 9: Wiki-Vote: selected concentration ratios from edge-swap random graphs (100 graphs; original concentrations average over three RAND-ESU seeds).

Triad	Concentration Ratio $C_k^i(G)/\hat{C}_k^i(G)$	Original	Random Mean	P-value
300	3.49	0.016%	0.0047%	0.00
120D	2.21	0.342%	0.155%	0.00
120U	2.14	0.437%	0.204%	0.00
210	2.09	0.116%	0.055%	0.00
030T	1.72	3.47%	2.02%	0.00
021U	0.97	24.26%	24.93%	1.00

Table 10: Amazon0302: selected concentration ratios from edge-swap random graphs (100 graphs; original concentrations are mean across three RAND-ESU seeds).

Triad	Concentration Ratio $C_k^i(G)/\hat{C}_k^i(G)$	Original	Random Mean
300	71,726	1.57%	2.18e-05%
120D	26,759	1.97%	7.35e-05%
210	20,419	2.32%	1.14e-04%
120U	6,235	1.55%	2.48e-04%
120C	3,095	0.35%	1.14e-04%
030T	2,713	1.60%	5.88e-04%
030C	2,263	0.025%	1.10e-05%
021U	1.11	48.46%	43.50%

Table 11: Wiki-Vote: direct Bender -Canfield expectations (T=100,000, seed 1; observed concentrations from full enumeration).

Triad	Observed (%)	Expected (%)	Ratio $C_k^i(G)/\hat{C}_k^i(G)$
021D	43.50%	44.10%	0.99
021U	24.30%	25.10%	0.97
021C	20.60%	24.90%	0.83
111U	4.20%	1.77%	2.36
030T	3.47%	2.46%	1.41
111D	2.68%	1.39%	1.93
300	0.0159%	4.38e-05%	363.1

Table 12: Amazon0302: direct Bender-Canfield expectations (T=100,000, seed 1; observed concentrations from full enumeration).

Triad	Observed (%)	Expected (%)	Ratio $C_k^i(G)/\hat{C}_k^i(G)$
021U	48.50%	44.10%	1.10
111D	16.80%	0.00169%	9.92e+03
021C	10.30%	39.90%	0.26
111U	7.39%	0.00062%	1.19e+04
021D	4.28%	16.00%	0.27
201	3.41%	1.30e-08%	2.62e+08
030T	1.58%	0.00134%	1.18e+03
300	1.56%	5.18e-18%	3.01e+17

Table 13: Selected motifs: edge-swap concentration ratios (mean across three RAND-ESU seeds and 100 graphs) vs. direct Bender-Canfield ratios (T=100,000, seed 1).

Dataset	Triad	Edge-swap Ratio $C_k^i(G)/\hat{C}_k^i(G)$	BC Ratio $C_k^i(G)/\hat{C}_k^i(G)$
Wiki-Vote	021U	0.97	0.97
Wiki-Vote	030T	1.72	1.41
Wiki-Vote	300	3.49	363.1
Amazon0302	021U	1.11	1.10
Amazon0302	030T	2.71e+03	1.18e+03
Amazon0302	300	7.17e+04	3.01e+17

Table 14: RAND-ESU vs. ESA (means across three seeds; ESA uses 5K samples per seed) on directed networks.

Dataset	Triad	RAND-ESU	ESA (mean)	Diff.
Wiki-Vote	021U	24.26%	24.30%	+0.04%
Amazon0302	021U	48.46%	47.63%	-0.83%

algorithms detect all 13 triads. Table 14 reports mean 021U concentrations across three seeds for both RAND-ESU and ESA (ESA uses 5K samples per seed). We cap ESA at 5K samples per seed to keep runtime manageable on large graphs. We piced 021U because it was the class most affected by the directed expansion bug, so this table is primarily a sanity check that ESA now recovers it and that concentrations are in the right range.

While ESA now discovers all triads, concentration estimates still differ slightly from RAND-ESU. With the Equation (??) probability correction applied, ESA is unbiased in expectation, but its non-uniform sampling distribution and much smaller sample size lead to higher variance. The small differences in Table 14 should therefore be interpreted as sampling noise, not evidence that ESA is more accurate; the purpose is to show that ESA no longer misses 021U and yields comparable estimates after the directed expansion fix.

6.5 Runtime and Scalability

Table 15 shows the efficiency of our implementation across different configurations (mean across three seeds). Even the largest network (roadNet-CA) is processed in about one minute for $k = 3$.This is because the network is a sparse structure and thus limits its branching.Wiki-Vote ,on otherhand has the highest runtime at larger k due to its dense structure and large branching factor in the ESU-tree.

7 Conclusion

The primary objective of this study was to develop a scalable and unbiased method for network motif detection to analyze modern large-scale networks without the combinatorial explosion associated with exhaustive enumeration. We addressed this through a parallelized Python implementation of RAND-ESU featuring automated probability scheduling, which successfully scaled to process

Table 15: Average runtime per run (seconds, mean across three seeds).

Dataset	k=3	k=4	k=5
Wiki-Vote	11.89	758.39	38,135.16
Amazon0302	23.95	89.47	1,412.51
CA-AstroPh	9.93	444.60	17,953.01
roadNet-CA	62.24	61.23	64.47

the nearly two-million-node roadNet-CA network in approximately one minute. Our experiments revealed that RAND-ESU provides highly stable concentration estimates varying by less than 0.05% across seeds and confirmed that domain-specific signatures, such as out-star dominance in voting networks and high triadic closure in collaboration graphs, are clearly discernible. We found that while concentration ratios provide a robust measure for significance, z-scores often produce uninterpretable values when random-graph variance is low; however, memory constraints remain a challenge for directed graphs at k=5 due to the discovery of thousands of unique motif classes. Although our parallel strategy and automated scheduling effectively manage large datasets, future work should implement streaming or approximate counting to handle the memory overhead of high-diversity motif classes and increase the random-graph ensemble size to improve the precision of statistical significance measures.

Acknowledgments

We thank the SNACS course staff at LIACS, Leiden University, for instruction and guidance throughout the course. All experiments were conducted using publicly available SNAP datasets. Reference C implementations from mfinder [4] and FANMODPlus [5] were used for methodology alignment (technical implementation).

References

[1] Y. Artzy-Randrup, S. J. Fleishman, N. Ben-Tal, and L. Stone. 2004. Comment on "Network Motifs: Simple Building Blocks of Complex Networks". *Science* 305 (2004), 1007c.

[2] P. W. Holland and S. Leinhardt. 1971. Transitivity in Structural Models of Small Groups. *Comparative Group Studies* 2, 2 (1971), 107–124.

[3] N. Kashtan, S. Itzkovitz, R. Milo, and U. Alon. 2004. Efficient Sampling Algorithm for Estimating Subgraph Concentrations and Detecting Network Motifs. *Bioinformatics* 20, 11 (2004), 1746–1758.

[4] Uri Alon Lab. 2004. Network Motif Software: mfinder. https://www.weizmann.ac.il/mcb/alon/download/network-motif-software. Accessed: 2025. Reference C implementation of ESA and edge-swap randomization.

[5] Zaritzky Lab. 2022. FANMODPlus: Fast Network Motif Detection Plus. https://github.com/zaritskylab/FANMODPlus. Accessed: 2025. Reference C++ implementation of RAND-ESU.

[6] J. Leskovec, L. A. Adamic, and B. A. Huberman. 2007. The Dynamics of Viral Marketing. *ACM Trans. on the Web* 1, 1 (2007), Article 5.

[7] J. Leskovec, D. Huttenlocher, and J. Kleinberg. 2010. Predicting Positive and Negative Links in Online Social Networks. In *WWW 2010*. ACM, 641–650.

[8] J. Leskovec, D. Huttenlocher, and J. Kleinberg. 2010. Signed Networks in Social Media. In *CHI 2010*. ACM, 1361–1370.

[9] J. Leskovec, J. Kleinberg, and C. Faloutsos. 2007. Graph Evolution: Densification and Shrinking Diameters. *ACM Trans. on Knowledge Discovery from Data* 1, 1 (2007), Article 2.

[10] J. Leskovec and A. Krevl. 2014. SNAP Datasets: Stanford Large Network Dataset Collection. http://snap.stanford.edu/data.

[11] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney. 2009. Community Structure in Large Networks: Natural Cluster Sizes and the Absence of Large Well-Defined Clusters. *Internet Mathematics* 6, 1 (2009), 29–123.

[12] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. 2002. Network Motifs: Simple Building Blocks of Complex Networks. *Science* 298, 5594 (2002), 824–827.

[13] SNAP. 2014. Amazon product co-purchasing network, March 02 2003. https://snap.stanford.edu/data/amazon0302.html. Accessed: 2025.

[14] SNAP. 2014. California road network. https://snap.stanford.edu/data/roadNet-CA.html. Accessed: 2025.

[15] SNAP. 2014. Collaboration network of Arxiv Astro Physics. https://snap.stanford.edu/data/ca-AstroPh.html. Accessed: 2025.

[16] SNAP. 2014. Wikipedia who-votes-on-whom network. https://snap.stanford.edu/data/wiki-Vote.html. Accessed: 2025.

[17] A. Vázquez, R. Dobrin, D. Sergi, J.-P. Eckmann, Z. N. Oltvai, and A.-L. Barabási. 2004. The Topological Relationship Between the Large-Scale Attributes and Local Interaction Patterns of Complex Networks. *PNAS* 101, 52 (2004), 17940–17945.

[18] S. Wernicke. 2005. A Faster Algorithm for Detecting Network Motifs. In *WABI 2005 (LNBI, Vol. 3692)*, R. Casadio and G. Myers (Eds.). Springer-Verlag, 165–177.