On Analyzing Graphs with Motif-Paths

Supplemental Material

XIAODONG LI[†], REYNOLD CHENG[†], KEVIN CHANG[‡], CAIHUA SHAN[†], CHENHAO MA[†],
HONGTAI CAO[‡], [†]Department of Computer Science, University of Hong Kong, Hong Kong SAR
[‡]Department of Computer Science, University of Illinois at Urbana-Champaign, USA

Because of the page limit in the paper, we report more details in the supplemental materials. In Section 1, we
attach the proofs for lemmas (cf. Section 1.1) and algorithm complexities (cf. Section 1.2). In Section 2, We report
more efficiency and effectiveness evaluations, including:
- Fig. 2 MOD-Indexing time and size as graph density increases;
- Fig. 3 SMP query time on graphs of different density and materialization levels;
- Fig. 4 breakdown of the offline MOD-Indexing time;
- Fig. 6 evaluation of generic motifs for link prediction on GAVI, EXTE and AMAZ;
- Tab. 2 link prediction performance with running time of each algorithm on each dataset;
- Tab. 3 local graph clustering performance with running time of each algorithm on each dataset;
- Fig. 5 local graph clustering performance comparison between MLGC-b and MLGC-c.

1 PROOF

1.1 Proof of the Lemmas

Proof of Lemma 1:

PROOF. For c1, there is no need to add motif-instances containing a "searched" node since all motif-instances
around node marked as "searched" have been found and added into candidates for $\mathcal{P}_{s,t}$. For c2, for the motif-
instances in $\mathcal{P}_{s,t}$, there are only two status of the nodes: "searched" and "discovered". We only select "discovered"
node as next seed because the "searched" nodes have been used as seed before and thus using them as next seed
will find duplicates. For c3, in the incremental search manner, $\mathcal{P}_{s,v}$ is found for the "undiscovered" node v when
v is covered by any motif-instance for the first time. Therefore, we only add motif-instances which contain at
least one node marked as "undiscovered" to push the incremental search forward. □

Proof of Lemma 2:

PROOF. Since $\mathcal{P}_{s,t}$ is the shortest sequence of motif-instances from m_s to m_t. For each motif-instance in the sequence, we pick the edge which links s, t and the nodes shared by the neighboring motif-instances, the path is the shortest one on W. Vise versa. □

Proof of Lemma 3:

PROOF. Assume that $\exists (i,j) \in V \times V$, $W_{i,j} = 1$ but there is no motif-instance of $\bar{\tau}$, then there must be motif-instance m such that $m \simeq \bar{\tau}' \& (i,j) \in E_m$, where $\bar{\tau}'$ is another motif-orbit of τ with seed $s \in V_m$. By switching the seed node, $m \simeq \bar{\tau}$ with seed $s' \in V_m$, which is contradictory to the assumption. □

Here we also show that most expansive nodes satisfy $d_e \leq 2$. As pointed by [3], the degree distributions of the motifs trend to be long-tailed as the size of the motif grows, e.g., most nodes will have smaller degrees for motifs of bigger sizes. Since the motif-orbits have the same structure as the motif, and the expansive degree is at most the degree of the expansive node, thus the expansive degree also follows this rule. Also, as pointed by [1], the number of motifs increase exponentially as the motif size grows; hence most expansive degrees are rather small. We collect all expansive degrees from the motif-orbits whose size is smaller than six (i.e., the motifs used in the paper), and 65% of them are only one while 21% of them are two. As the motif orbit size grows, the numbers are more lopsided. The detailed results are shown in Fig. 1.
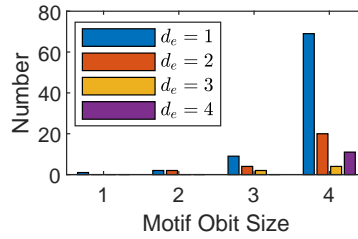

Fig. 1. The number of expansive nodes with different expansive degrees (d_e) from motif orbits of different sizes.

1.2 Time and Space Complexities

In this section, we analyze the time and space complexities of the algorithms listed in the paper. In the following paragraphs, we use θ to denote $\binom{|V|}{|V_\tau|}$, as the maximum number of motif-instances of motif τ, D_τ to denote the cost of one-round expansion (i.e., extend the current expansive nodes for one hop), η to denote the number of rounds for expansion, γ to denote the number of rounds for expansion with motif-instances partially materialized in the MOD-Index and I to denote the space cost of the MOD-Index. We summarize the complexities in Table 1. Since the fist materialization level is more valuable (i.e., pretty good query speedup with smaller indexing cost; see Fig. 2 and 3), we also report the complexities when the materialization level k is set as 1. To make it clear, we use τ to denote a motif and τ' to denote the motif-orbit of motif τ.

First, the baseline algorithm (Algorithm 1 in the paper, i.e., BASE) will generate the higher-order graph of $O(\theta)$ nodes and $O(\theta \times |V_\tau| \times \binom{|V|}{|V_\tau|-1})$ edges. To run the set-to-set shortest path algorithms on the higher-order graph (lines 3-5 in BASE), it costs $O(\theta^3)$ by time and $O(\theta^2)$ by space (e.g., by Floyd-Warshall algorithm) [4].

Second, we analyze the complexities for MOD-Indexing (Algorithm 2 in the paper, i.e., MODC). For the space complexity, given a seed node s from the data graph (i.e., $s \in V$) and a motif-orbit τ' from the top-k levels of the MOD-Index (i.e., $\tau' \in B_k$), the motif-instances to be materialized in the MOD-Index for τ' around node

Table 1. Summary of algorithm complexities.

Alg.	Time	Space	Time ($k = 1$)	Space ($k = 1$)
BASE	$O(\theta^3)$	$O(\theta^2)$	-	-
MODC	$O(\sum_{s\in V}\sum_{\tau'\in B_k} D_{\tau'}^{\eta})$	$O(I)$	$O(|V| \times d_{max}^2)$	$O(|V| \times d_{max}^2)$
MODQ	$O(D_\tau^\gamma)$	$O(I)$	$O(1)$	$O(|V| \times d_{max}^2)$
SMP	$O(|V| \times D_\tau^\gamma + \theta)$	$O(\theta + I)$	$O(\theta)$	$O(\theta + |V| \times d_{max}^2)$
ESMP	$O(|V| \times D_\tau^\gamma + |E| + \theta)$	$O(|E| + \theta + I)$	$O(|E| + \theta)$	$O(|E| + \theta + |V| \times d_{max}^2)$
MGD	$O(|V| \times D_\tau^\gamma + \theta)$	$O(\theta + I)$	$O(\theta)$	$O(\theta + |V| \times d_{max}^2)$
MKI	$O(|V|^{2L} \times D_\tau^\gamma + \theta)$	$O(\theta + I)$	$O(|V|^{2L} + \theta)$	$O(\theta + |V| \times d_{max}^2)$
MLGC	$O(|V| \times \hat{k} \times D_\tau^\gamma + \theta)$	$O(d_{max} \times \hat{k} + \binom{\hat{k}}{|V_\tau|} + I)$	$O(\theta)$	$O(\theta + |V| \times d_{max}^2)$
MBET	$O(|V| \times (D_\tau^\gamma + |V| + |E_W|) + \theta)$	$O(\binom{d_{max}^{\phi_\tau}}{|V_\tau|} + |V|^2 + I)$	$O(|V| \times (|V| + |E_W|) + \theta)$	$O(|V|^3)$

Note that $\theta = \binom{|V|}{|V_\tau|}$, $D_\tau = \binom{d_{max}}{\tau.d_e}$, $\eta = \min\{k, \phi_{\tau'}\}$, $\gamma = \begin{cases} \phi_\tau - k, & k \leq \phi_\tau \\ 0, & otherwise \end{cases}$ and $I = \sum_{s\in V}\sum_{\tau'\in B_k} \binom{d_{max}^{\phi_{\tau'}}}{|V_{\tau'}|-1}$.

s can be described by $\binom{d_{max}^{\phi_{\tau'}}}{|V_{\tau'}|-1}$, where $\phi_{\tau'}$ is the maximum distance[1] from the seed s to each node $t \in \tau'$, i.e., $\phi_{\tau'} = \max_{t\in V_\tau} dist(s, t)$, and d_{max} is the maximum degree; hence $d_{max}^{\phi_{\tau'}}$ denotes the searching area of the motif-orbit τ' (i.e., the set of nodes to be explored in the original graph) in the materialization process, and $\binom{d_{max}^{\phi_{\tau'}}}{|V_{\tau'}|-1}$ is the maximum number of motif-instances that we can found within this area. According to this formula, we get $I = |V| \times d_{max}^2$ when $k = 1$. Similarly, for the time complexity, given a seed node s from the data graph (i.e., $s \in V$) and a motif-orbit τ' from the top-k levels of the MOD-Index (i.e., $\tau' \in B_k$), the cost of each round of node expansion can be described by $D_{\tau'} = \binom{d_{max}}{\tau'.d_e}$ where $\tau'.d_e$ is the maximum expansive degree within the motif-orbit τ'. In other words, it is the one-hop expansion cost from extending the expansive nodes within motif-orbit τ'. Note that we will need $\eta = \min\{k, \phi_{\tau'}\}$ number of iterations for expansion, i.e., extend k hops when $k < \phi_{\tau'}$ (τ' is not fully materialized in this case), otherwise $\phi_{\tau'}$ hops (τ' is fully materialized in this case). Therefore, for each motif-orbit $\tau' \in B_k$ around each node $s \in V$, the MOD-Index construction time is $O(D_{\tau'}^{\eta})$. After taking $k = 1$, we get the time complexity for constructing the first level motif-orbits is $|V| \times d_{max}^2$.

Third, we analyze the complexities for accessing the MOD-Index (Algorithm 3 in the paper, i.e., MODQ). Give a motif τ and a query node s, we access the MOD-Index and check that for each motif-orbit τ', if the motif-orbit is fully materialized (i.e., $\phi_{\tau'} \leq k$). If so, we directly access the Index with constant cost $O(k)$ (or simple $O(1)$); otherwise we need to expand $\phi_{\tau'} - k$ iterations with cost $D_{\tau'}^{\phi_{\tau'}-k}$. Since the number of motif-orbits are fixed and limited, the total expansion cost depends on the motif-orbits of maximum $\phi_{\tau'}$, that is, the diameter of the motif ϕ_τ; therefore, the time complexity of MODQ is $O(D_\tau^{\phi_\tau-k})$.

Next, we analyze the complexities for SMP (Algorithm 4 in the paper) and ESMP (Algorithm 8 in the paper). For both algorithms, we at most enumerate $O(|V|)$ nodes, since the algorithms will terminate after all nodes are marked "searched". When searching each node, we will call function MODQ to find the motif-instances that contain the node (line 6 in Algorithm 4), leading to time cost $O(|V| \times D_\tau^\gamma)$, where $\gamma = \begin{cases} \phi_\tau - k, & k \leq \phi_\tau \\ 0, & otherwise \end{cases}$ is the number of expansions when there are k levels of motif-instances materialized in the MOD-Index. Note that we also need to access the elements in the motif-instances to discover the nodes (lines 8-11 in Algorithm 4), leading to another $O(\theta \times |V_\tau|)$ cost, or simply $O(\theta)$ cost. For the space cost, the algorithm needs to cache all the nodes, motif-instances and the MOD-Index in the worst case, leading to $O(\theta + I)$ complexity. Similarly,

---

[1]Note that we use ϕ_τ to denote the diameter of the motif τ, i.e., $\phi_{\tau'} = \max_{s, t\in V_\tau \times V_\tau} dist(s, t)$.

ESMP shares the same complexities as SMP, except the bridging edges, which are $O(|E|)$ in both time and space complexity. After taking $k = 1$, we get the complexities shown in Table 1.

Finally, we analyze the complexities for the motif-path based applications. For MGD and MKI (Algorithm 5 in the paper), MGD shares the same complexities as SMP, since it directly call the function SMP (lines 2-3 in Algorithm 5 first part); MKI removes the node status during searching the nodes, making it possible to repeatedly search the nodes. For the worst case, it will search $O(|V|^{2 \times L})$ nodes (lines 3-6 in Algorithm 5 second part) where L is maximum path length as a system parameter in the algorithm. Therefore, the total time complexity is $O(|V|^{2L} \times D_\tau^\gamma + \theta)$. For space complexity, we only need to maintain the current layer and next layer results (line 6 in Algorithm 5 second part), thus the space complexity is $O(|V|^2 + \theta + I)$, or simply $O(\theta + I)$. For MLGC (Algorithm 6 in the paper), it runs in the way like SMP and terminates when the fist $\hat{k}$ nodes are discovered, leading to a smaller cost. For MBET (Algorithm 7 in the paper), the first phase calls function MODQ for $O(|V|)$ times (line 1 in Algorithm 7), which costs $O(|V| \times D_\tau^\gamma)$ for expansion to calculate the motif-adjacency matrix $W = (V, E_W)$; the second phase runs all shortest path search (lines 2-9 in Algorithm 7), which costs $O(|V| \times (|V| + |E_W|))$. For space complexity, it need to cache $O(\binom{d_{max}^{\phi_\tau}}{|V_\tau|})$ motif-instances, W of size $O(|V|^2)$ and the MOD-Index I.

2 SUPPLEMENTAL EVALUATIONS

In this section, we evaluate the indexing cost and shortest motif-path (SMP) query time on graphs of different density levels. We follow the graph generation approach described in Section 6 of the paper and fix the node number as 300000. The edge numbers vary from 300000 to 1500000, making the average degree as 2 ($D1$), 4 ($D2$), 6($D3$), 8 ($D4$) and 10 ($D5$) respectively.

Next, we follow the setup in Section 6.1 to build the MOD-Index on each dataset with different materialization levels (i.e., $k = 0, 1, 2$ and 3). Note that $k = 0$ means that there is no motif-instances materialized in the MOD-Index. We report the indexing time (left y-axis) and space (right y-axis) in Fig. 2. Note that the MOD-Index supports all motifs reported in the paper. Similar trend as Fig. 8 in the paper is also observed here. The indexing cost (both time and space) are much higher when the materialization level is set as 3. However, as we will show, $k = 1$ can already obtain good performance in the online phase with considerable time and space cost in the offline phase.

Then we randomly sample 500 node pairs in each dataset; for each pair of nodes, we run SMP with motifs from Fig. 9 in the paper. As shown in Fig. 3, the running time of different motifs varies a lot, and the biggest speedup is usually obtained when the materialization level is set as 1. For the cliques, their motif-orbits are within the first level of the MOD-Index, hence the 1st materialization level is enough for them. For motif ⅄ and ◇, their motif-orbits are on the second and the third levels of the MOD-Index, hence the best performance gain is obtained on the second materialization level or the third materialization level. This trend is more clear for ⅄, since it is more frequent in these datasets. In general, the indexing cost is acceptable when considering the speedup of the query time, especially for the first materialization level.
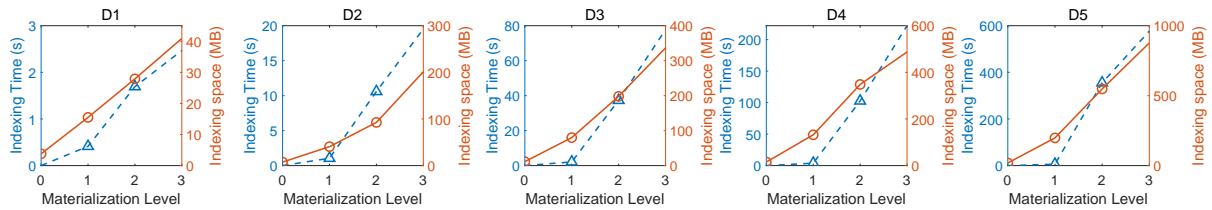

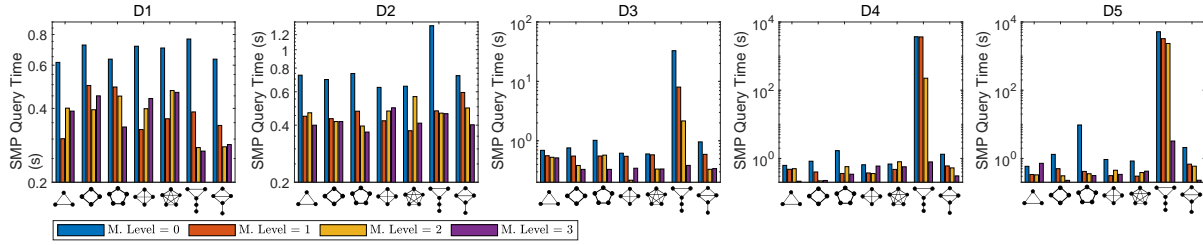Fig. 2. MOD-Indexing time and space cost as graph density grows.

Fig. 3. Shortest motif-path query time on graphs of different density and materialization levels (M. Level).

Then we report the breakdown of the MOD-Indexing time, which can be generally decomposed into three steps. First, the motif-orbits are detected from the given set of motifs and then organized into a tree structure in the horizontal level and vertical level (e.g., Fig. 2 in the paper). Second, the motif-instances from the data graph are materialized. Finally, the shortest path distances are calculated and associated with the nodes in the MOD-Index. To speedup the calculation of the shortest paths, we use the well-developed function `all_pairs_shortest_path_length(G)` in package NetworkX[2] to calculate the shortest path distances. Since shortest path distances are widely supported by different graph databases, other packages can also be used here. Note that all the three steps of MOD-Indexing only need to run once, which is independent of the motifs to be used, and organizing the motif-orbits are also independent of the data graph. Here we use all motifs of size smaller than 6 and report the breakdown of the MOD-Indexing time on the graphs of different densities in Fig. 4.
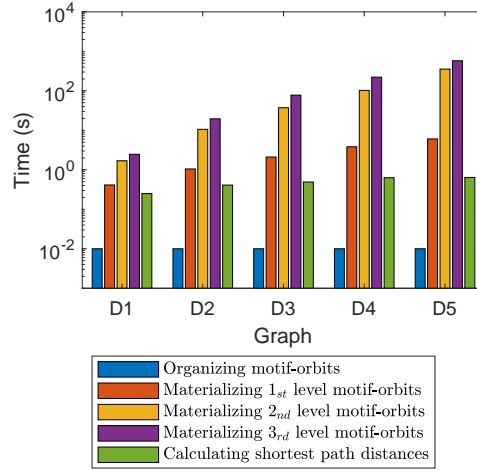
Fig. 4. Breakdown of the offline MOD-Indexing time.

Next, we report the detailed running time of Table 5 and Table 6 of the paper in Table 2 and Table 3 respectively. The algorithms with the defragmentation approach in [2] are also included, i.e., MKI-c, MGD-c and MLGC-c. We also report the results of MLGC-c as the value of $\hat{k}$ varies in Fig. 5, which is not included in the paper for the limited space. Finally, we report more effectiveness results of using different motifs in link prediction. As shown in Fig. 6, we also involve MCN, which is not included in the paper for the limited space. Results show that MCN

---

[2]https://networkx.org/

obtains similar results as MGD when serious fragmentation occurs (e.g., when ⬟ is used), but its running time is much higher and outperformed by the enhanced version of motif-path.

Table 2. MKI/MGD performance with AUC and running time reported. Numbers of top-3 highest AUC are marked bold.

| Method	GAVI		EXTE		DBLP		AMAZ		YOUT	
	AUC	Time	AUC	Time	AUC	Time	AUC	Time	AUC	Time
CN	0.72	1.2ms	0.56	1.7ms	0.77	12.9ms	0.62	13.5ms	0.54	0.2s
JC	0.7	1.1ms	0.48	1.6ms	0.55	14.4ms	0.52	12.8ms	0.44	0.2s
AA	0.75	1.2ms	0.57	1.6ms	**0.81**	12.0ms	0.65	12.6ms	0.52	0.2s
PA	0.59	1.2ms	0.76	1.7ms	0.64	13.4ms	0.63	14.9ms	0.76	0.2s
FM	0.65	1.2ms	0.65	1.5ms	0.59	14.0ms	0.64	15.4ms	0.69	0.2s
HT	0.6	1.4ms	0.7	1.7ms	0.64	13.5ms	0.59	15.2ms	0.53	70.1s
RPR	0.61	1.5ms	0.51	1.6ms	0.76	12.7ms	0.62	13.4ms	0.48	72.0s
MCN	0.67	1.0s	0.62	3.2s	0.75	38.8s	0.61	2.4s	0.65	16.6m
MLP+GB	**0.89**	7.0m	**0.83**	76.9m	**0.82**	35.9m	**0.72**	1.4m	**0.83**	70.4h
KI	0.69	36.2ms	0.6	0.5s	0.69	2.2s	0.6	97.5ms	0.63	1.1m
MKI	0.71	0.1s	0.66	1.2s	0.74	23.5s	0.63	6.1s	0.66	5.7m
MKI-c	0.62	6.4m	0.67	18.0m	-	-	-	-	-	-
MKI-b	**0.76**	0.1s	**0.87**	2.0s	**0.75**	31.2s	**0.73**	10.6s	**0.76**	7.5m
GD	0.5	2.2ms	0.5	3.1ms	0.5	23.0ms	0.5	26.7ms	0.5	1.8s
MGD	0.67	50.1ms	0.63	0.2s	0.65	2.5s	0.66	1.8s	0.55	1.4m
MGD-c	0.52	1.3m	0.51	3.2m	-	-	-	-	-	-
MGD-b	**0.75**	32.7ms	**0.84**	30.6ms	0.72	2.8s	**0.81**	0.9s	**0.87**	1.8s
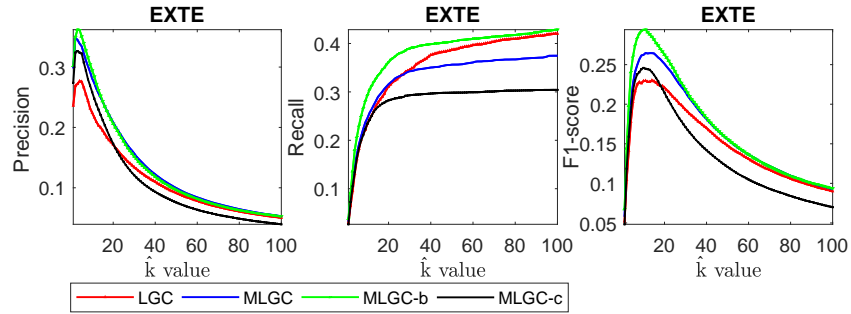
Fig. 5. Local graph clustering results on EXTE, with Precision, Recall and F1-score reported.

REFERENCES

[1] Xiaowei Chen, Yongkun Li, Pinghui Wang, and John Lui. 2016. A general framework for estimating graphlet statistics via random walk. *Proceedings of the VLDB Endowment* (2016).

[2] Pei-Zhen Li, Ling Huang, Chang-Dong Wang, and Jian-Huang Lai. 2019. EdMot: An Edge Enhancement Approach for Motif-aware Community Detection. In *Proceedings of the 25th ACM SIGKDD*. 479–487.

[3] Sridevi Kamla Maharaj. 2018. *Graphlet Analysis Of Networks*. Ph.D. Dissertation. UC Irvine.

Table 3. MLGC performance with F1-score and running time reported. Numbers of top-3 highest F1-score are marked bold.

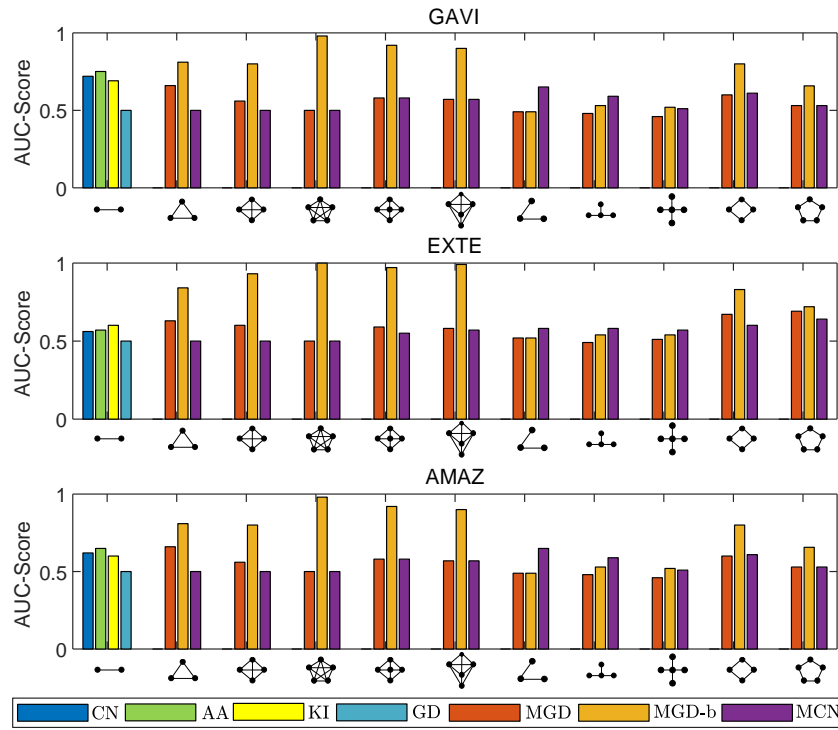| Method	GAVI		EXTE		DBLP		AMAZ		YOUT	
	F1	Time	F1	Time	F1	Time	F1	Time	F1	Time
TECTONIC	0.39	7.0s	**0.44**	24.6s	-	-	0.37	-	-	-
MAPPR	0.39	0.2s	**0.42**	0.1s	**0.34**	5.1s	0.35	4.4s	0.15	16.8s
EdMot	0.33	21.1s	0.38	1.1m	-	-	-	-	-	-
LGC	**0.42**	9ms	0.36	12.2ms	0.33	1.3s	**0.63**	89.6ms	**0.17**	0.7s
MLGC	**0.41**	3.1ms	0.3	8.7ms	**0.35**	1.3s	**0.59**	7.8ms	**0.16**	8.7s
MLGC-c	0.39	23.1s	0.29	1.1m	-	-	-	-	-	-
MLGC-b	**0.42**	3.7ms	**0.38**	9.9ms	**0.35**	1.5s	**0.65**	8.8ms	**0.23**	13.8s

Fig. 6. Evaluation of generic motifs (edge, cliques, quasi-cliques, stars and cycles) for link prediction on GAVI, EXTE and AMAZ.

[4] Kenneth H Rosen and Kamala Krithivasan. 2012. *Discrete mathematics and its applications: with combinatorics and graph theory*. Tata McGraw-Hill Education.