

LLMs Between the Nodes: Community Discovery Beyond Vectors

Ekta Gujral and Apurva Sinha

Abstract. Community detection in social network graphs plays a vital role in uncovering group dynamics, influence pathways, and the spread of information. Traditional methods focus primarily on graph structural properties, but recent advancements in Large Language Models (LLMs) open up new avenues for integrating semantic and contextual information into this task. In this paper, we present a detailed investigation into how various LLM-based approaches perform in identifying communities within social graphs. We introduce a two-step framework called CommLLM, which leverages the GPT-4o model along with prompt-based reasoning to fuse language model outputs with graph structure. Evaluations are conducted on six real-world social network datasets, measuring performance using key metrics such as Normalized Mutual Information (NMI), Adjusted Rand Index (ARI), Variation of Information (VOI), and cluster purity. Our findings reveal that LLMs, particularly when guided by graph-aware strategies, can be successfully applied to community detection tasks in small to medium-sized graphs. We observe that the integration of instruction-tuned models and carefully engineered prompts significantly improves the accuracy and coherence of detected communities. These insights not only highlight the potential of LLMs in graph-based research but also underscore the importance of tailoring model interactions to the specific structure of graph data.

Keywords: Large Language Model (LLM), Social Network Graphs, Community Detection, Data mining

1 Introduction

Community detection in social networks is crucial for understanding the underlying structure [16] and dynamics of complex systems. It helps uncover groups of nodes with dense internal connections, revealing patterns such as shared interests, behaviors, or functions. This technique is widely applied across domains like marketing (to target consumer groups), cyber security (to detect malicious clusters), biology (to identify functional modules in protein networks), and politics (to map ideological communities). As networks [15,3] grow more intricate, traditional methods struggle with scalability and adaptability. Recently, large language models (LLMs) [13,7,33] have emerged as powerful tools in this field. Their ability to

reason over text and structure enables them to detect nuanced relationships and community patterns in both textual and graph data. LLMs are increasingly being integrated into research workflows, outperforming many classical algorithms in terms of flexibility, interpretability, and cross-domain applicability.

Despite their impressive capabilities across a wide range of natural language tasks, existing large language models (LLMs) encounter significant challenges when applied to community detection within social network graphs. One of the primary limitations is their lack of innate access to graph-structured data; LLMs are fundamentally designed to process sequential text rather than inherently parallel or interconnected structures such as graphs [6]. This sequential bias limits their ability to model complex topological features, such as node centrality, clustering coefficients, and higher-order neighborhood interactions, all of which are critical for accurately identifying communities. Moreover, LLMs do not naturally encode graph-specific inductive biases—such as permutation invariance or localized neighborhood aggregation—which are essential for tasks like node clustering and structural role identification. Their inability to capture hierarchical or multi-scale structures within graphs often results in shallow representations that fail to generalize across diverse network topologies. While techniques such as graph embeddings or hybrid models attempt to bridge this gap, these approaches typically decouple the graph topology from the language modeling process, which can lead to a loss of structural fidelity and semantic nuance.

Scalability is another core issue. LLMs, especially those deployed on large-scale or dynamic social networks, suffer from computational inefficiencies and memory bottlenecks. Representing large graphs in a form consumable by LLMs (e.g., through serialization or textual descriptions) becomes increasingly infeasible as network size grows. Additionally, LLMs lack temporal awareness and dynamic reasoning capabilities required to model evolving communities or time-sensitive interactions in dynamic social graphs. Furthermore, most LLMs are trained on general-purpose corpora and lack task-specific fine-tuning for graph mining or community detection tasks [32]. As a result, they often struggle with domain adaptation, interpretability, and robustness when applied to network analysis. They may also be susceptible to spurious correlations or biases embedded in the training data, which can skew the understanding of social graph dynamics and community boundaries. Lastly, LLMs typically do not incorporate explicit constraints or priors related to graph theory, such

as modularity maximization or spectral properties, which are central to many traditional community detection algorithms.

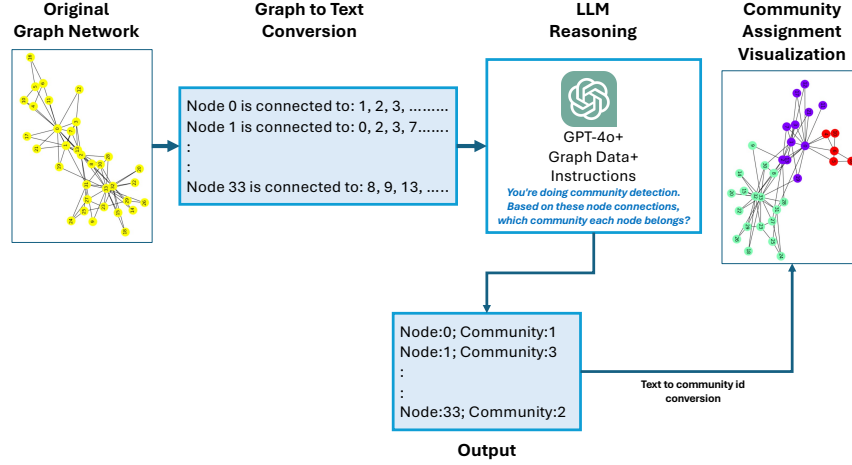


Fig. 1. Illustration of CommLLM that includes two main steps: graph-to-text conversion and then LLM Reasoning to detect the communities in the graph

In contrast, our work introduces a novel approach CommLLM (as shown in Figure 1) powered by GPT-4o [23] that empowers large language models (LLMs) to detect communities more effectively by integrating graph-aware text representations with language-based reasoning. By embedding structural information directly into the input, our method preserves the topology of the network while leveraging the LLM’s capacity for pattern recognition and inference. This hybrid strategy allows the model to understand both the semantic and relational aspects of nodes. It enables nuanced community detection, even in complex or sparsely connected graphs. Additionally, our approach improves interpretability and generalization across different types of networks. Our contributions are summarized as follows:

- We propose method CommLLM, which uses GPT-4o to detect communities in social network graphs. It includes two main steps: graph-to-text conversion and then prompt engineering to detect the communities in the graph. (See Figure 1)
- The results on real datasets indicate that CommLLM outperforms the baseline LLM methods in context to community detection. Furthermore, prompt comparison experiments reveal that the designed

prompt effectively aids LLMs in understanding the graph communities.

- The dataset used and graph-to-text conversion script in this study is publicly available at link¹ for better reproducibility of the research.

2 Literature Review

2.1 Traditional Community detection

Community detection in complex networks has garnered significant attention, leading to the development of a wide range of methodologies. Modularity-based approaches, such as the Louvain algorithm [5], aim to maximize a modularity score to identify groups of nodes with dense intra-community connections relative to a null model. Spectral clustering techniques utilize the eigenvectors of graph Laplacians to reveal partitions in the network structure [20]. Label propagation algorithms [26] iteratively update node labels based on neighbor consensus, offering scalability and simplicity. Statistical inference methods, including the Stochastic Block Model [18], model networks probabilistically, often capturing assortative and disassortative structures. Hierarchical clustering builds dendrograms of nodes through agglomerative or divisive strategies, while edge betweenness-based methods [10] sequentially remove edges with high centrality to expose community boundaries. Random walk-based techniques, such as Walktrap, cluster nodes based on transition probabilities, whereas clique percolation methods identify overlapping communities through adjacent k-cliques. Information-theoretic frameworks like Infomap [27] aim to compress the description of network flows. More recent approaches include graph embedding techniques (e.g., DeepWalk [25,2]), which learn low-dimensional node representations for clustering, and deep learning-based models like graph neural networks (GNNs) that jointly learn features and community assignments [19]. Additionally, matrix factorization methods (e.g., NMF) decompose adjacency matrices to detect latent community structure. Bayesian methods incorporate prior distributions into community inference [24], while evolutionary algorithms apply genetic operations for optimization. Tensor decomposition-based community detection [14,8] leverages multi-dimensional data representations to uncover complex community structures in networks. By decomposing high-order

¹ https://drive.google.com/drive/folders/1IKV-Qi_oz0hjAJHqCszLIbw1ACmgNWXM

tensors (e.g., user-item-time) into latent factors, it captures hidden relationships and overlapping communities [28,11] that traditional methods might miss. Multi-resolution methods [15] further enable detection of communities at varying scales, addressing limitations related to resolution limits in modularity-based techniques.

2.2 LLMs in graph reasoning

In recent years, researchers have increasingly investigated the use of large language models (LLMs) for graph reasoning tasks [31]. The paper [30] highlighted LLMs' capabilities in graph reasoning and introduced the NL-Graph dataset to assess these skills. The paper [17] further underscored the importance of graph encoding strategies in shaping how LLMs interpret and generate outputs. To facilitate LLM-based graph understanding, encoding methods such as Adjacency and Incident [9] have been developed, using integer-based node encoding with differing edge encoding schemes. While these methods mainly encode graph topology, they often fall short in addressing task-specific requirements like community detection. To bridge this gap, we propose a novel encoding method inspired by Incident, specifically tailored for the community detection domain by integrating community-aware structural cues. Additionally, although task-oriented prompts—like the Chain-of-Thought [34] and Build-a-Graph [30] have been proposed to enhance LLM performance in graph reasoning, they lack the domain-specific context needed for community detection. Recently, the paper [21] introduced detecting community structure with large language models. Our work provide how different LLM models perform on the same task and it bridges the current gap in LLMs applicability to the community detection for social network graphs.

3 Approach

Traditional community detection algorithms, such as modularity maximization, spectral clustering, or label propagation, often rely on graph-theoretic computations. In contrast, we explore a novel approach CommLLM (Figure 1) by leveraging the reasoning capabilities of a large language model (LLM), specifically a GPT-4o, to infer community assignment directly from graph connectivity data.

Let $G = (V, E)$ be an undirected graph, where $V = \{0, 1, 2, \dots, n-1\}$ is the set of nodes and $E \subseteq V \times V$ is the set of edges representing connections between nodes. Each node is associated with a set of neighboring

nodes $N(i)$, defined as:

$$N(i) = \{j \in V \mid (i, j) \in E\}$$

Our goal is to partition the graph into a set of communities $C = \{C_1, C_2, C_3 \dots C_N\}$, such that nodes within the same community are more densely connected to each other than to nodes in other communities. Mathematically as follow:

$$\bigcup_{\ell=1}^k C_\ell = V, \quad C_i \cap C_j = \emptyset \quad \text{for } i \neq j$$

3.1 Graph-to-Text Conversion

Graph-to-text conversion is a fundamental task in natural language generation that entails the translation of structured graph data into coherent, informative, and contextually rich natural language descriptions. This process is essential for interpreting and communicating complex relational information embedded within graphs, facilitating a deeper understanding of the network's structure and the interactions between its components. The goal is to generate textual summaries that not only describe the raw data but also convey the underlying patterns, behaviors, and trends represented by the graph.

To achieve this, we first extract salient features from the graph, such as node roles, edge weights, community affiliations, and subgraph structures. These elements serve as the foundational data for our approach, CommLLM, and are carefully analyzed to capture the most relevant and informative aspects of the graph's topology. By encoding these features in a way that maintains the integrity of the graph's structure, we can produce textual summaries that highlight key relationships and important dynamics within the network.

For example, consider a graph, as shown in figure 2, which consists of 34 nodes and 78 undirected edges, representing frequent communication patterns among individuals in a social network. In this case, the nodes correspond to individuals, and the edges reflect the communication interactions between them. Through graph-to-text conversion, we can generate a descriptive narrative that summarizes the network's key characteristics, such as identifying central or influential nodes, the strength of relationships between nodes (based on edge weights), and any notable community structures that emerge within the network. This process not only aids in

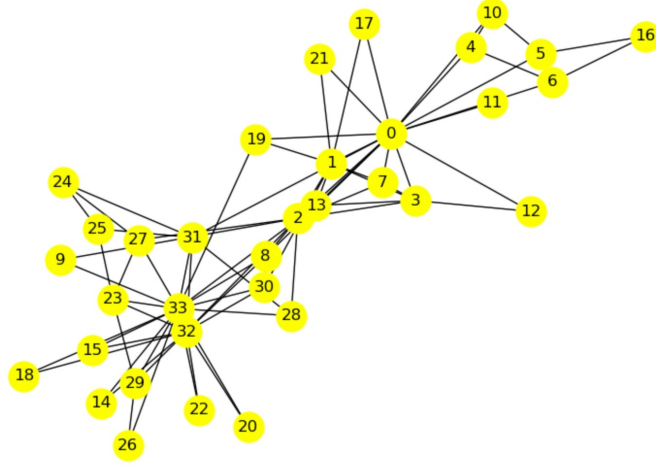


Fig. 2. Illustration of Karate Club Network

understanding the graph's overall organization but also provides valuable insights into the social dynamics at play.

The Graph-to-text conversion will be as follow:

Node 0 is connected to: 1, 2, 3, 4, 5, 6, 7, 8,
 Node 1 is connected to: 0, 2, 3, 7, 13, 17,
 ⋮
 Node 33 is connected to: 8, 9, 13, 14, 15, 18,

3.2 Community Detection via LLM Reasoning

We propose a novel community detection method CommLLM that formulates the problem as a text-based reasoning task. The input to the language model is a natural language description of node connections in the form as given in section 3.1 and the prompt to the model is:

Graph Details + "A community is a group of nodes that are more densely connected to each other internally than to the rest of the network. You are doing community detection. Based on these node connections, which community each node belongs? Give outcome as Node:<node id>; Community:<Community id> format. Do not give any other text."

The LLM implements an implicit function:

Dataset	#V	#E	#C	directed
Karate Club	34	78	2	N
Football	115	613	12	N
WebKB	187	298	5	Y
Terrorist Attacks	645	3172	6	Y
Cora	2708	5278	7	Y
CiteSeer	3279	4552	6	Y

Table 1. Real world social network dataset. V = Number of nodes; E = Number of Edges; C : Number of communities in the graph.

$$f : \mathcal{T}(G) \rightarrow \mathcal{C}$$

f : Connectivity Descriptions \rightarrow Community Assignments

where $\mathcal{T}(G)$ denotes a textual representation of the graph (i.e., the connectivity format described above), and C is the resulting community assignment. The model infers clusters by identifying patterns in the described node relationships, without explicit optimization of an objective. This approach treats the LLM as a zero-shot or few-shot reasoning engine over structural information conveyed via text, rather than a model trained explicitly on community detection tasks or graph statistics.

The output of CommLLM is captured as:

```
Node:0; Community:1
Node:1; Community:3
...
Node:33; Community:2
```

In summary, CommLLM re-frames community detection as a language-based reasoning task, requiring no structural assumptions or predefined models. By leveraging GPT-4o’s capability to interpret connectivity patterns through natural language prompts, we demonstrate a flexible and generalizable approach to discovering community structures in graph data.

4 Experiments

In this section, we answer following research questions:

- **Q1:** Does CommLLM able to provide relevant communities for different social network graphs?
- **Q2:** Is there effect of number of nodes on CommLLM?
- **Q3:** Does CommLLM is sensitive to different prompts?

We execute all methods for 10 times to handle hallucinations and average score is reported.

4.1 Dataset

To evaluate the proposed framework at scale, we use 6 real world datasets [12] as show in table 1. The dataset is publically avaiable to explore at [link²](#).

4.2 Baselines

To evaluate the performance of CommLLM, we compare CommLLM with LLM based methods namely gpt-3.5-turbo [22], llama3-3-70b-instruct [29], claude-3.5-sonnet [1], and gemini-1.5-pro [4].

4.3 Evaluation metrics

The evaluation metrics Normalized Mutual Information (NMI), Adjusted Rand Index (ARI), Variation of Information (VOI), and Purity are used to evaluate the detected communities. A lower VOI value indicates a more similar clusters, while a higher VOI value suggests that the clusters are more different. For other metrics larger value implies a better algorithm performance.

4.4 Experiment Results

Results on networks We evaluate CommLLM, our reasoning-based community detection method, across six benchmark datasets and compare its performance against leading large language models (LLMs). The results are shown in Table 4 and Figure 3. For Karate Club TribeLLM achieves $NMI = 0.90$ outperforming all baselines by a significant margin (see Figure 3). Competing models like LLama3 and Claude struggled, with NMI values around 0.34, indicating a poor match with ground truth. For Football dataset, CommLLM leads with $NMI = 0.91$ while keeping VOI low at 0.31. Gemini-1.5-pro was a close second in community detection quality but had a notably higher VOI (0.40), showing CommLLM’s superior consistency. For WebKB dataset, CommLLM substantially outperforms others in both NMI (0.38) and ARI (0.39), whereas competitors score below 0.25. This is particularly noteworthy

² https://drive.google.com/drive/folders/1IKV-Qi_oz0hjAJHqCszLIbw1ACmgNWXM

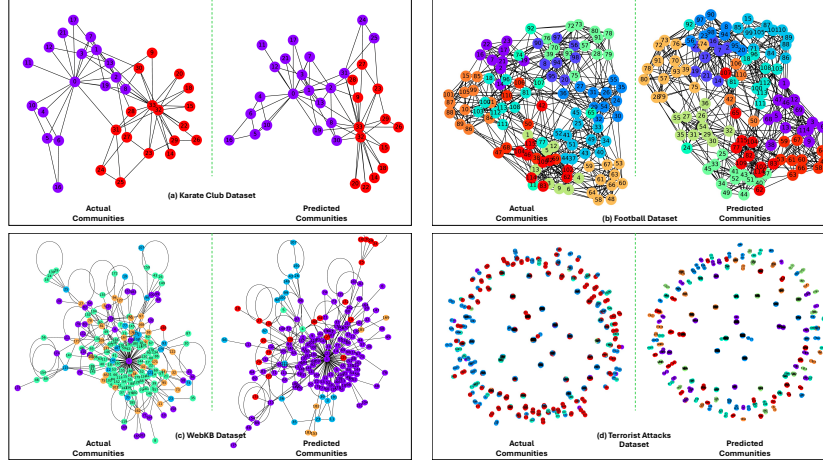


Fig. 3. Illustration of community assignment by CommLLM for the graphs

given the relatively low ARI scores of all other models (≤ 0.19), highlighting proposed method's robustness in web graph structures. For Terrorist Attacks dataset, CommLLM achieves competitive scores (NMI = 0.37, ARI = 0.34) and maintains low VOI (1.14). Claude outperforms in NMI (0.50) and Purity (0.73), but CommLLM remains close while offering better balance across all metrics. For Cora and CiteSeer, CommLLM is the only model to complete inference successfully with high-quality scores (NMI = 0.35, ARI = 0.36). Other models either ran out of tokens (Claude, Llama3) or underperformed, indicating CommLLM's efficiency with larger graphs.

CommLLM demonstrates superior generalization and reasoning ability across diverse datasets, particularly excelling in both well-defined and noisy graph structures. It consistently outperforms baseline LLMs, many of which fail due to token limits, lack of contextual reasoning, or insufficient robustness in community assignment inference. This answer our **Q1**.

Effect of the number of nodes on CommLLM As the number of nodes in the graph increases, the input and output token requirements for the proposed method grow proportionally as shown in table 2, since the entire graph structure is embedded within the prompt provided to the CommLLM model. This results in larger message sizes, directly impacting memory consumption and model processing capacity. Additionally,

Dataset	#V	Time Consumption	Input Tokens
Karate Club	34	20	777
Football	115	25	4,573
WebKB	187	41	3,144
Terrorist Attacks	645	65	23,618
Cora	2708	176	57,146
CiteSeer	3279	552	58,453

Table 2. Effect of the number of nodes on CommLLM in terms of Time (seconds) and token used. For Cora and CiteSeer model can only predict partial nodes at a time.

increased token length leads to longer inference times due to the model’s attention mechanism scaling with input size. These factors collectively introduce computational overhead, making scalability a key consideration. Efficient graph encoding or summarization strategies to mitigate these effects in larger graphs will be part of our future research work and is out of scope for this work. This answer our **Q2**.

Karate Club Dataset					
Method	NMI	ARI	Purity	VOI	Input Tokens
Prompt 1	0.78	0.84	0.90	0.22	778
Prompt 2	0.64	0.68	0.75	0.56	770
Prompt 3	0.55	0.63	0.66	0.85	793
CommLLM	0.90	0.93	0.98	0.11	777

Table 3. Performance of CommLLM w.r.t different variation of prompts.

Prompt comparisons We performed experiment on Football dataset using 3 different version of CommLLM prompt as given below.

- **Prompt 1:** A community is a group of nodes that are more densely connected to each other internally than to the rest of the network. Based on these node connections, output in the format Node:< *nodeid* >; Community:< *Communityid* >.
- **Prompt 2:** You are doing community detection. Assign each node to a community. Give outcome as Node:< *nodeid* >; Community:< *Communityid* > format. Do not give any other text.
- **Prompt 3:** A community is a group of nodes that are more densely connected to each other internally than to the rest of the network. You are doing community detection. Along with assigning communities, provide a brief justification for each decision. Format as Node:< *nodeid* >; Community:< *Communityid* >; Reason:< *reason* >.

- **Prompt 4** (CommLLM): A community is a group of nodes that are more densely connected to each other internally than to the rest of the network. You are doing community detection. Based on these node connections, which community each node belongs? Give outcome as Node:<node id>; Community:<Community id> format. Do not give any other text.

Table 3 shows the results of this experiment. The first prompt doesn't explicitly say the task is community detection, leaving the intent unclear and missing critical information. However, LLM was able to stick to the community detection work. The second prompt is incomplete and lacks the actual node connections or network structure to base decisions on. The proposed method performance degrades with incomplete information. The third prompt is closer to actual ask but does not ask for "no extra text" constraint and introduces justification, which diverted the model from a concise mapping. The performance is lower of all the prompt for this. The forth prompt gives the best results and we choose this as our model's prompt for evaluating the other datasets. As we know that language models are highly sensitive to wording, context, and phrasing. variation in inference is expected behavior, as these models rely on learned statistical patterns and subtle variations can shift the model's interpretation or emphasis. This answer our **Q3**.

5 CONCLUSIONS AND FUTURE WORK

In conclusion, our proposed method CommLLM leverages the capabilities of GPT-4o to perform effective community detection in social network graphs through a two-step process: graph-to-text conversion followed by LLM-based reasoning. Traditional LLMs struggle with directly processing large and complex graphs due to input size limitations and lack of native graph understanding. Our approach addresses this by designing a structured conversion technique and tailored prompts that enable the LLM to capture both the semantics and topology of the graph. Experimental results confirm that this method significantly enhances the model's ability to understand graph structures and accurately detect communities. For future work, we aim to extend this framework to handle much larger and dynamic graphs, pushing the boundaries of scalable community detection using language models. Overall, this study contributes a novel methodology for incorporating LLM reasoning into structured graph tasks, offering practical guidance and foundational understanding for future work at the intersection of natural language processing and graph analytics.

Karate Club				
Method	NMI	ARI	Purity	VOI
gemini-1.5-pro	0.84	0.88	0.97	0.16
gpt-3.5-turbo	0.82	0.85	0.87	0.23
llama3-3-70b-instruct	0.34	0.26	0.76	0.62
claude-3.5-sonnet	0.34	0.27	0.77	0.61
CommLLM	0.90	0.93	0.98	0.11
Football				
Method	NMI	ARI	Purity	VOI
gemini-1.5-pro	0.87	0.82	0.87	0.40
gpt-3.5-turbo	0.52	0.28	0.48	1.37
llama3-3-70b-instruct	0.54	0.32	0.57	1.54
claude-3.5-sonnet	0.53	0.31	0.53	1.55
CommLLM	0.91	0.84	0.91	0.31
WebKB				
Method	NMI	ARI	Purity	VOI
gemini-1.5-pro	0.24	0.18	0.57	1.42
gpt-3.5-turbo	0.25	0.19	0.58	1.29
llama3-3-70b-instruct	0.16	0.12	0.55	1.22
claude-3.5-sonnet	0.10	0.11	0.58	1.15
CommLLM	0.38	0.39	0.67	0.89
Terrorist Attacks				
Method	NMI	ARI	Purity	VOI
gemini-1.5-pro	0.33	0.32	0.53	2.04
gpt-4-turbo	0.35	0.31	0.50	1.14
llama3-3-70b-instruct	0.41	0.30	0.61	2.34
claude-3.5-sonnet	0.50	0.36	0.73	1.12
CommLLM	0.37	0.34	0.56	1.14
Cora				
Method	NMI	ARI	Purity	VOI
gemini-1.5-pro	0.23	0.13	0.41	1.58
gpt-4-turbo	0.25	0.27	0.54	1.23
llama3-3-70b-instruct	—	—	—	—
claude-3.5-sonnet	—	—	—	—
CommLLM	0.35	0.36	0.56	1.18
CiteSeer				
Method	NMI	ARI	Purity	VOI
gemini-1.5-pro	0.18	0.12	0.23	1.88
gpt-4-turbo	0.15	0.17	0.38	1.53
llama3-3-70b-instruct	—	—	—	—
claude-3.5-sonnet	—	—	—	—
CommLLM	0.21	0.22	0.38	1.32

Table 4. Performance of CommLLM w.r.t state-of-art methods. Bold letter shows which method outperformed for the dataset. Symbol — indicates model was out of tokens for full data and performance in terms of NMI was low (< 0.1). The performance variance for each dataset is between ± 0.01 and ± 0.03 .

References

1. Claude 3.5 sonnet, 2024.
2. Saba Al-Sayouri, Ekta Gujral, Danai Koutra, Evangelos E Papalexakis, and Sarah S Lam. t-pine: Tensor-based predictable and interpretable node embeddings. *Social Network Analysis and Mining*, 10:1–11, 2020.
3. Saba A Al-Sayouri, Ekta Gujral, Danai Koutra, Evangelos E Papalexakis, and Sarah S Lam. t-pne: tensor-based predictable node embeddings. In *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 491–494. IEEE, 2018.
4. Rohan Anil et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens. *arXiv preprint arXiv:2403.05530*, 2024.
5. Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, 2008.
6. Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.
7. Sumit Kumar Dam, Choong Seon Hong, Yu Qiao, and Chaoning Zhang. A complete survey on llm-based ai chatbots. *arXiv preprint arXiv:2406.16937*, 2024.
8. Hao Fang, Qu Wang, Qicong Hu, and Hao Wu. Mntd: An efficient dynamic community detector based on nonnegative tensor decomposition. *arXiv preprint arXiv:2407.18849*, 2024.
9. Bahare Fatemi, Jonathan Halcrow, and Bryan Perozzi. Talk like a graph: Encoding graphs for large language models. *arXiv preprint arXiv:2310.04560*, 2023.
10. Michelle Girvan and Mark EJ Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002.
11. Alexander Gorovits, Ekta Gujral, Evangelos E Papalexakis, and Petko Bogdanov. Larc: Learning activity-regularized overlapping communities across time. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1465–1474, 2018.
12. LINQS Research Group. Linqs lab, 2024.
13. Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, et al. A survey on llm-as-a-judge. *arXiv preprint arXiv:2411.15594*, 2024.
14. Ekta Gujral and Evangelos E Papalexakis. Smacd: Semi-supervised multi-aspect community detection. In *Proceedings of the 2018 SIAM International Conference on Data Mining*, pages 702–710. SIAM, 2018.
15. Ekta Gujral, Evangelos E Papalexakis, Georgios Theocharous, and Anup Rao. Hacd: Hierarchical agglomerative community detection in social networks. In *2019 IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE, 2019.
16. Ekta Gujral, Ravdeep Pasricha, and Evangelos Papalexakis. Beyond rank-1: Discovering rich community structure in multi-aspect graphs. In *Proceedings of The Web Conference 2020*, pages 452–462, 2020.
17. Jiayan Guo, Lun Du, Hengyu Liu, Mengyu Zhou, Xinyi He, and Shi Han. Gpt4graph: Can large language models understand graph structured data? an empirical evaluation and benchmarking. *arXiv preprint arXiv:2305.15066*, 2023.
18. Paul W Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. Stochastic blockmodels: First steps. *Social Networks*, 5(2):109–137, 1983.

19. Thomas N Kipf and Max Welling. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*, 2016.
20. Mark EJ Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582, 2006.
21. Li Ni, Haowen Shen, Lin Mu, Yiwen Zhang, and Wenjian Luo. Comgpt: Detecting local community structure with large language models. *arXiv preprint arXiv:2408.06658*, 2024.
22. OpenAI. Gpt-3.5 turbo technical overview, 2023. Accessed via OpenAI API.
23. OpenAI. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.
24. Tiago P Peixoto. Hierarchical block structures and high-resolution model selection in large networks. *Physical Review X*, 4(1):011047, 2014.
25. Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 701–710. ACM, 2014.
26. Usha Nandini Raghavan, Réka Albert, and Soundar Kumara. Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E*, 76(3):036106, 2007.
27. Martin Rosvall and Carl T Bergstrom. Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences*, 105(4):1118–1123, 2008.
28. Fatemeh Sheikholeslami and Georgios B Giannakis. Identification of overlapping communities via constrained egonet tensor decomposition. *IEEE Transactions on Signal Processing*, 66(21):5730–5745, 2018.
29. Hugo Touvron, Louis Martin, Kevin Stone, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.
30. Heng Wang, Shangbin Feng, Tianxing He, Zhaoxuan Tan, Xiaochuang Han, and Yulia Tsvetkov. Can language models solve graph problems in natural language? *Advances in Neural Information Processing Systems*, 36:30840–30861, 2023.
31. Rongzheng Wang, Shuang Liang, Qizhi Chen, Jiasheng Zhang, and Ke Qin. Graphtool-instruction: Revolutionizing graph reasoning in llms through decomposed subtask instruction. *arXiv preprint arXiv:2412.12152*, 2024.
32. Xiao Wang, Yu Liu, Qitian Zhang, Meng Liu, Dawei Yin, and Chuxu Zhang. Llm4graph: Large language models are strong graph reasoners. In *arXiv preprint arXiv:2306.09382*, 2023.
33. Hui Wei, Zihao Zhang, Shenghua He, Tian Xia, Shijia Pan, and Fei Liu. Plangenllms: A modern survey of llm planning capabilities. *arXiv preprint arXiv:2502.11221*, 2025.
34. Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Authors

Ekta Gujral is staff data scientist in Walmart. She received Ph.D. in Computer Science from the University of California Riverside. Her research work focused on modeling and mining multi-aspect graphs with scalable streaming tensor decomposition for real-world applications. During her Ph.D. journey, She interned with Adobe, Snapchat, and LLNL Research teams. Her work spans a wide range of areas including tensor analysis, anomaly detection, explainable AI, and graph mining. She is motivated to build useful models that would deliver a great impact on society and business.

Apurva Sinha is staff data scientist in Walmart. She received Master of Science from University of Texas - Dallas. She did Bachelor of Science in Computer Science from Kalinga Institute of Industrial Technology, India. She has diverse expertise in advanced data analytics, data science, software development, Agile project, program management. Her research area includes machine learning, data analysis, NLP, and adaptive signal processing, with a vision to advance efficient and trustworthy artificial intelligence.