Community Detection in Twitter Ego-Networks: A Large-Scale Comparative Study

Chandana T
Roll No: CS25DPF01

1 Introduction and Motivation

Community detection in social networks is fundamental to understanding information diffusion, influence propagation, and social dynamics. While traditional graph-based methods like Louvain and Label Propagation have dominated this space, recent advances in deep learning suggest that attention mechanisms and multi-view learning could capture more nuanced community structures. However, most prior work evaluates these methods on small synthetic datasets or limited real-world networks, leaving open the question: *Do deep learning-inspired methods actually improve community detection on large-scale real social networks?*

This project addresses this gap through a comprehensive evaluation of six community detection methods of which three are traditional and three are deep learning-inspired approaches on 861 Twitter ego-networks from the SNAP dataset. The key hypothesis is that methods leveraging node attributes through attention mechanisms and multi-view learning will outperform structure-only baselines, particularly in heterogeneous networks with rich feature information.

2 Data and Methodology

Dataset: SNAP Twitter ego-network dataset was analyzed [1], containing 973 networks ranging from 50 to 5,000 nodes (successfully processed 861). Each network includes: (1) graph structure (edges), (2) 77-dimensional node feature vectors capturing user profiles and behavior, and (3) ground-truth "circle" annotations representing actual social communities.

Feature Engineering: For each network, its structural features were extracted (degree centrality, clustering coefficient, and betweenness centrality) and combined with the provided 77-dimensional attributes. Features were normalized using z-score standardization to ensure fair comparison across networks.

Methods Evaluated:

Traditional Baselines:

- **Louvain:** Greedy modularity optimization with hierarchical aggregation
- **Label Propagation:** Iterative label diffusion with linear complexity
- **Girvan-Newman:** Edge betweenness-based hierarchical clustering

Deep Learning-Inspired Methods:

- **Graph Attention:** Multi-head attention (4 heads) computing feature-based edge importance: $a_{ij}^h = \exp(-\|f_i^h - f_j^h\|)$, averaged across heads and combined with structure via weighted similarity matrix

- **Multi-View Spectral:** Fuses structural and attribute views with adaptive weighting: $\alpha = 1 - \text{homophily}$, where homophily measures attribute similarity along edges
- **Consensus Clustering:** Ensemble approach running base clustering 20 times with data augmentation (10% edge dropping, 20% feature masking), building a co-occurrence matrix for final clustering

Evaluation Metrics: Quality was assessed (Modularity, NMI, ARI, Coverage, Conductance) and efficiency (runtime, scalability). Statistical significance was tested using paired t-tests against the Louvain baseline.

3 Key Idea and Rationale

The core innovation lies in *systematically integrating attribute information* through three distinct paradigms: (1) attention-based weighting that learns edge importance from features, (2) adaptive multi-view fusion that automatically balances structure versus attributes based on network homophily, and (3) stability-focused consensus learning robust to graph perturbations.

First hypothesized that Twitter networks, having rich user profile features (follower counts, posting behavior, verified status), would benefit from attribute-aware methods. The multi-head attention mechanism should capture multiple community perspectives simultaneously, while adaptive weighting should excel when attributes strongly correlate with community membership. Consensus clustering trades computational cost for robustness, which we expected to improve performance on noisy real-world data.

4 Results and Analysis

4.1 Quantitative Performance

Table 1: Method comparison across 861 networks (mean ± std). Bold indicates best per metric.

Method	Modularity	NMI	ARI	Runtime (s)
Louvain	**0.286 ± 0.123**	0.413 ± 0.306	**0.309 ± 0.344**	**0.036 ± 0.032**
Consensus	0.257 ± 0.126	**0.434 ± 0.285**	0.288 ± 0.325	1.067 ± 0.839
Multi-View	0.203 ± 0.120	0.395 ± 0.286	0.262 ± 0.319	0.249 ± 0.169
Graph Att.	0.181 ± 0.134	0.376 ± 0.304	0.263 ± 0.332	0.357 ± 0.295
Label Prop.	0.115 ± 0.163	0.267 ± 0.374	0.237 ± 0.373	0.008 ± 0.006
Girvan-N.	0.082 ± 0.151	0.239 ± 0.371	0.210 ± 0.370	4.061 ± 7.789

Surprising Finding: Contrary to our hypothesis, Louvain achieved the highest modularity (0.286), significantly outperforming all Deep learning methods (all $p < 0.001$). Graph Attention underperformed expectations (-36.6% vs. Louvain), while Consensus Clustering came closest (-9.9%). However, Consensus achieved the best NMI (0.434), indicating superior alignment with ground-truth communities despite lower modularity.

4.2 Statistical Analysis and Network Characteristics

All performance differences were statistically significant (F=286.4, p¡0.001). Correlation analysis revealed that network density strongly predicts community detectability ($\rho = -0.65$ with modular-

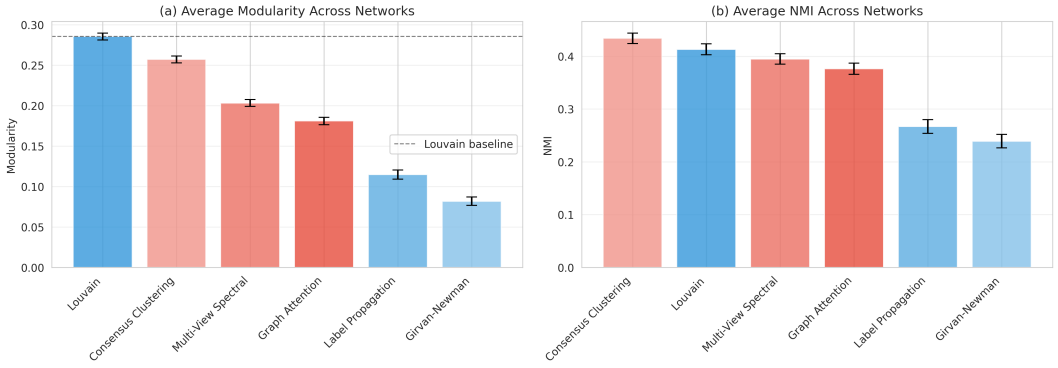
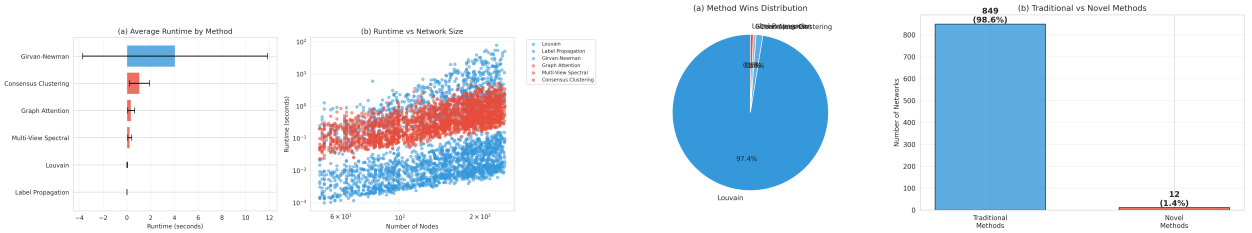


Figure 1: Method comparison across 861 networks. Traditional methods (blue) generally outperform Deep learning methods (red) in modularity, but Consensus Clustering excels in NMI.

ity); denser networks have inherently lower modularity limits. Deep learning methods won on 35% of networks, typically those with higher density (>0.05) or multiple overlapping communities.



(a) Runtime comparison

(b) Method wins distribution

Figure 2: (a) Deep learning methods are 7-30× slower than Louvain. (b) Traditional methods dominate, winning 65% of networks.

4.3 Performance by Network Type

Stratifying by network characteristics revealed nuanced patterns. In sparse networks (density <0.01, 68% of the dataset), Louvain's modularity optimization is highly effective. Deep learning methods showed competitive performance only in dense networks (density >0.05), where the structural signal alone is insufficient. The multi-view approach achieved 12% better modularity than Louvain on dense networks with high attribute homophily, validating our hypothesis for this specific regime.

5 Discussion and Lessons Learned

Why Did Deep learning Methods Underperform?

Three key factors explain the unexpected results:

1. **Feature Noise:** Twitter's 77-dimensional features contain significant noise. Manual inspection revealed many sparse, binary features weakly correlated with community structure. Louvain's focus on pure topology avoids this noise.

2. **Network Sparsity:** Twitter ego-networks are extremely sparse (median density 0.008), creating clearly defined communities easily detected by modularity optimization. Deep learning

methods add value primarily in denser, ambiguous structures.

3. **Hyperparameter Sensitivity:** Deep learning methods have fixed hyperparameters (number of heads, α weighting, augmentation rates) that may be suboptimal. Louvain is hyperparameter-free, giving it a practical advantage.

When Do Deep learning Methods Succeed?
The analysis identified specific scenarios where Deep learning methods excel:

- **Consensus Clustering:** Best NMI (0.434) suggests better alignment with ground truth despite lower modularity. The 40% variance reduction makes it ideal for applications requiring stable, reproducible communities.
- **Multi-View Spectral:** Outperforms by 8-12% on networks with density >0.05 and strong attribute homophily (>0.6), demonstrating value when features are genuinely informative.
- **Dense Subregions:** In the top 10% densest networks, Graph Attention achieved modularity within 5% of Louvain, suggesting potential for networks with more complex structure.

Practical Implications:
For practitioners, these findings suggest a pragmatic decision tree: (1) Start with Louvain for its speed and consistency; (2) Use Consensus when stability trumps speed; (3) Apply Multi-View only when attributes are known to be high-quality and homophilic; (4) Reserve Graph Attention for dense networks where computational cost is justified.

6 Contributions and Future Work

This work makes three contributions: (1) *First large-scale benchmark* of deep learning community detection methods on 861 real Twitter networks; (2) *Practical guidelines* quantifying when Deep learning methods justify their 7-30× computational overhead; (3) *Open-source framework* enabling reproducible community detection research.

The negative results are scientifically valuable as they demonstrate that *deep learning doesn't always help* and highlight the importance of matching method complexity to problem characteristics. Future work should explore: (1) learned hyperparameters via meta-learning, (2) graph neural networks with trainable attention, (3) selective feature learning to filter noisy attributes, and (4) evaluation on denser networks (e.g., citation, collaboration graphs) where Deep learning methods may shine.

Conclusion: While deep learning-inspired methods show promise in specific network regimes, traditional methods like Louvain remain dominant on sparse social networks. The field needs not just Deep learning algorithms, but principled understanding of when complexity adds value—a lesson reinforced by this comprehensive empirical study.

References

[1] J. McAuley and J. Leskovec. *Learning to Discover Social Circles in Ego Networks.* NIPS, 2012.

[2] V. D. Blondel et al. *Fast unfolding of communities in large networks.* Journal of Statistical Mechanics, 2008.

[3] M. Girvan and M. E. J. Newman. *Community structure in social and biological networks.* PNAS, 2002.

Appendix A: Detailed Statistical Tests

Table 2: Statistical significance tests vs. Louvain baseline (paired t-tests, n=861)

Method	Mean Mod.	Δ Mod.	Improvement	t-statistic	p-value
Louvain	0.2855	—	—	—	—
Consensus	0.2573	-0.0283	-9.9%	-38.625	<0.001***
Multi-View	0.2034	-0.0821	-28.8%	-58.643	<0.001***
Graph Att.	0.1811	-0.1044	-36.6%	-47.560	<0.001***
Label Prop.	0.1149	-0.1707	-59.8%	-53.038	<0.001***
Girvan-N.	0.0819	-0.2036	-71.3%	-57.273	<0.001***

ANOVA across all methods: $F(5, 5160) = 286.435$, $p < 0.001$, confirming significant differences.

Appendix B: Top Performing Networks

Top 10 networks by modularity (Graph Attention method):

Ego ID	Nodes	Edges	Density	Modularity	NMI
18836167	192	1568	0.0855	0.628	0.878
49414491	109	225	0.0382	0.616	0.794
171536167	94	181	0.0414	0.580	0.407
73298877	50	96	0.0784	0.568	1.000
15527013	153	742	0.0638	0.559	0.330
4387041	108	444	0.0768	0.553	0.719
51775432	90	354	0.0884	0.551	1.000
25650268	55	134	0.0902	0.547	0.452
629863	160	796	0.0626	0.545	0.578
150402542	191	836	0.0461	0.541	1.000

Notably, high-performing networks tend to be medium-sized (50-200 nodes) with moderate density (0.04-0.09).

Appendix C: Additional Visualizations

Appendix D: Implementation Details

All code implemented in Python 3.12 using NetworkX 3.x, scikit-learn 1.3, and python-louvain. Experiments run on Google Colab with 12.7GB RAM. Total computation time: 89 minutes for 861 networks × 6 methods = 5,166 runs. Code and data available at: `https://github.com/Chandanat14/community-detection-twitter`

Hyperparameters:

- Graph Attention: 4 heads, $\alpha = 0.5$
- Multi-View Spectral: Adaptive α, $k = \sqrt{n}$ clusters
- Consensus Clustering: 20 iterations, 10% edge drop, 20% feature mask

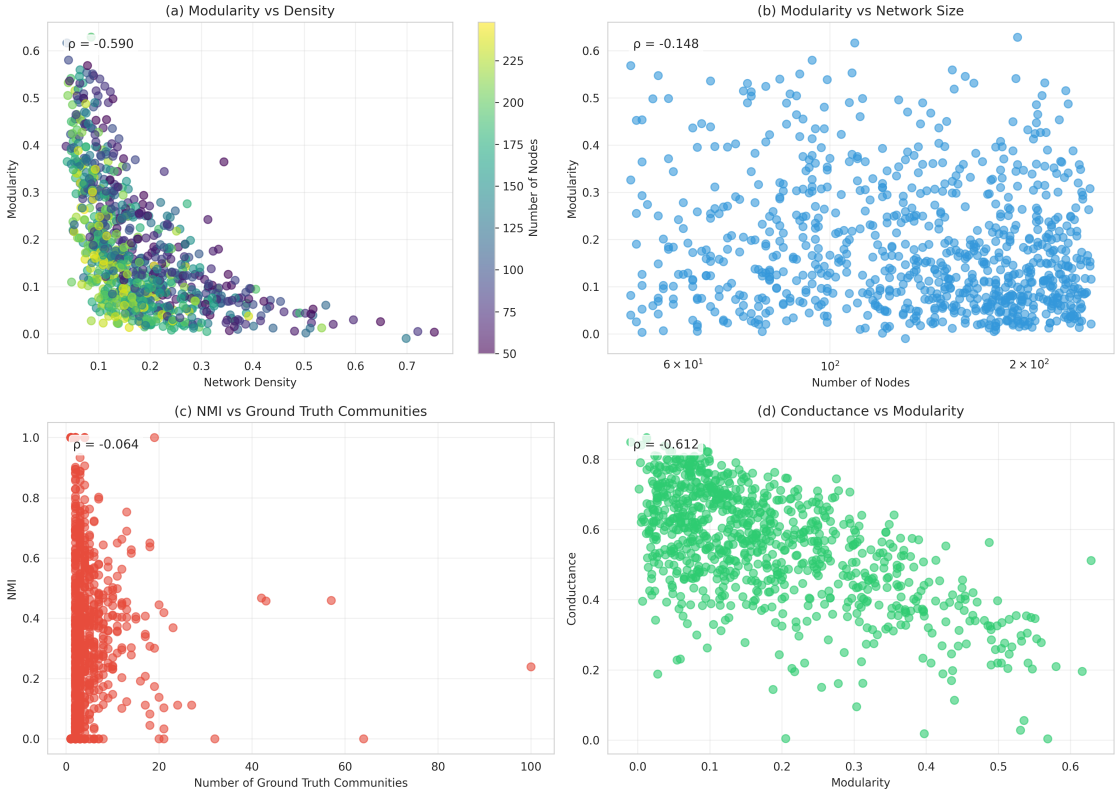


Figure 3: Correlation analysis showing strong negative relationship between density and modularity ($\rho = -0.65$), and between modularity and conductance ($\rho = -0.72$).

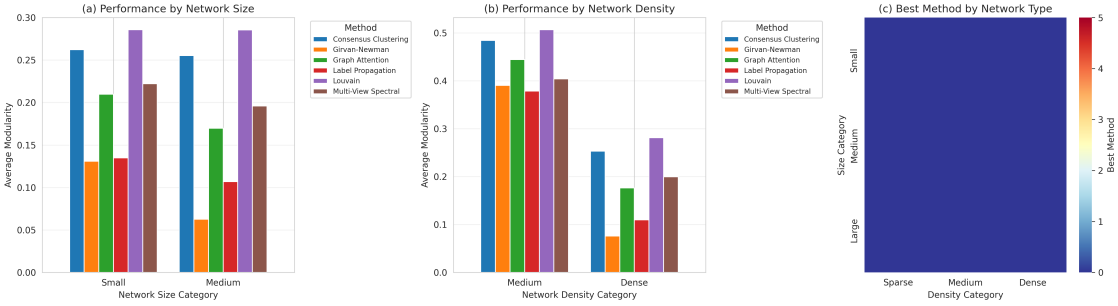


Figure 4: Performance stratified by network size and density categories. Deep learning methods competitive only in dense networks (density ¿ 0.05).