Università degli Studi di Padova

MSc in Data Science

Exploring the Influence of Graph Neural Network-Based Link Prediction on Social Contagion Dynamics

IIIA-CSIC
INSTITUT
D'INVESTIGACIÓ
EN INTEL·LIGÈNCIA
ARTIFICIAL

Master Candidate: Antoni Valls Cifre

Supervisor: Dr. Alberto Testolin

UNIVERSITAS · STUDII PADUANI · MCCXXII

Overview

Introduction

Rise of Social Media

- Social media has become a **cornerstone of modern life**, profoundly shaping people's personalities, interests, and even political views.
- Traditional journalism has shifted to prioritize algorithmic engagement through strategies like clickbait and sensationalist content [1, 2].
- Politicians, too, exploit this algorithmic driven ecosystem that incentivizes extreme, divisive, and simplified narratives over balanced, complex reporting (*e.g.*, 2024 US presidential election) [3, 4].

Introduction

What is “Social Contagion”?

- Social media networked structure influences not only what users see but also how they interpret and form opinions. The position individuals hold within these networks — and the connections they forge — directly **shapes their exposure to information** [*5*].

- In this context, **social contagion** is defined as the process through which ideas, behaviours, or information propagates across networks. It is often modeled using epidemic spreading frameworks.

Introduction

Research Motivation

- Social networks continuously evolve, shaped by both organic growth and algorithmic interventions. At the heart of many online platforms lies **link prediction** (LP), a crucial technology that reshapes these networks by recommending connections. Examples include “People You May Know” on LinkedIn or “Suggested Friends” on Facebook.

- This study aims to take the first steps in addressing an interesting yet existing gap by exploring **how different LP algorithms influence social contagion** processes at both graph and node level. We focus exclusively on Graph Neural Network-based LP algorithms.

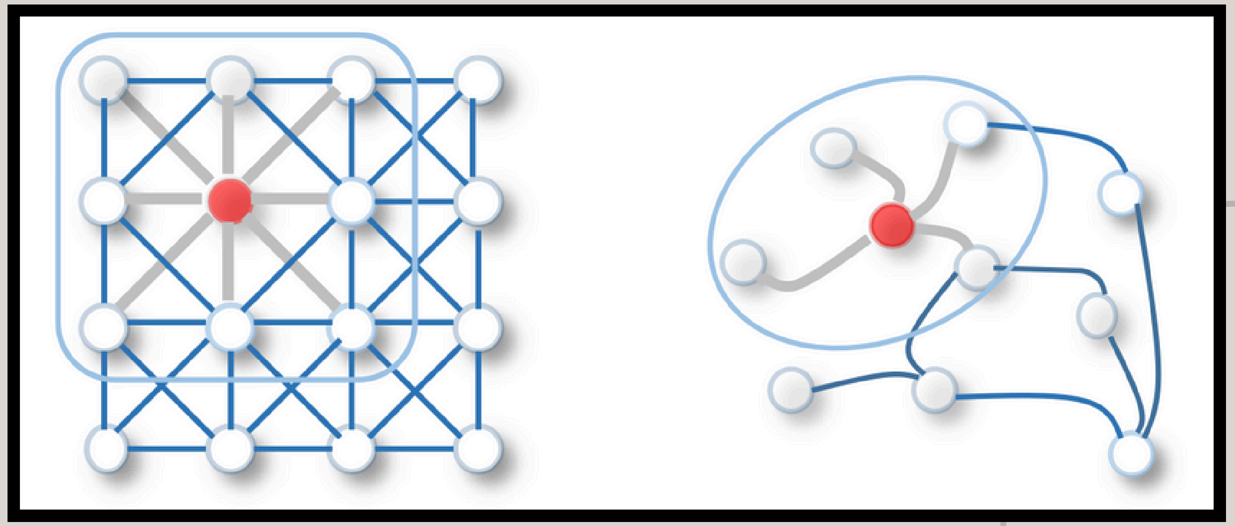
Graph Neural Networks

What are them?

- Neural networks designed to operate on **graph-structured data**, defined as *G(V,E)*.
  - GNNs use node features (such as user profiles in social networks or atomic properties in molecular graphs) to generate node embeddings.

  - During training GNNs integrate information from **both graph topology and node features**, obtaining node embeddings that capture complex, non-linear relationships in graphs. Typically, they outperform traditional approaches.

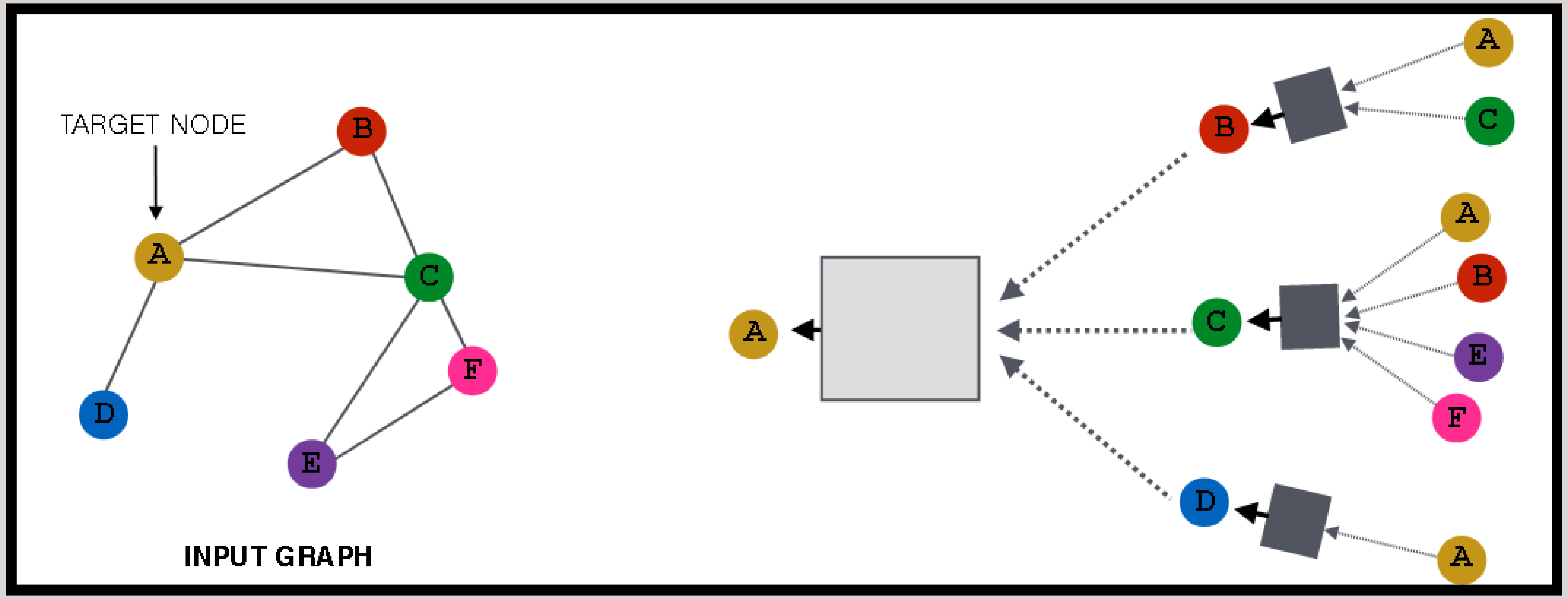
  - They **generalize convolution** from CNNs to graph data, using a message-passing mechanism to aggregate information from unordered, variable-sized neighborhoods in the graph structure.



Graph Neural Networks

Message-Passing

- The cornerstone of GNN architecture is **neural message passing.**
- During each iteration, nodes aggregate information from its local neighborhood, and as these iterations progress each node embedding gradually incorporates information from increasingly distant parts of the graph.
    - After *k* iterations every node embedding contains information about its *k*-hop neighborhood.

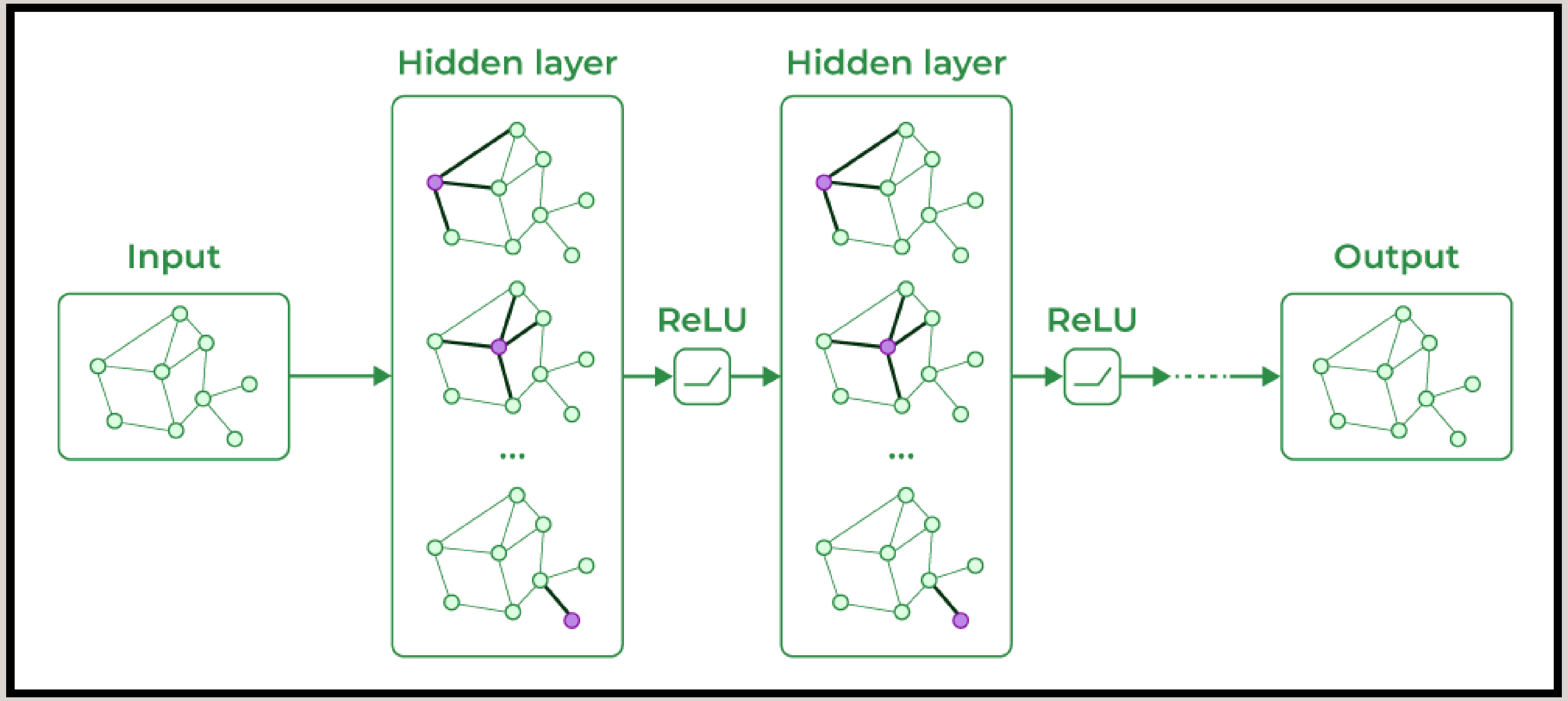


Graph Neural Networks

Message-Passing

This process can be formally expressed as:

$$\mathbf{h}_u^{(k)} = \text{UPDATE}^{(k-1)}\left(\mathbf{h}_u^{(k-1)}, \text{AGGREGATE}^{(k-1)}\left(\left\{\mathbf{h}_v^{(k-1)}, \forall v \in \mathcal{N}(u)\right\}\right)\right) = \text{UPDATE}^{(k-1)}\left(\mathbf{h}_u^{(k-1)}, \mathbf{m}_{\mathcal{N}(u)}^{(k-1)}\right)$$



Since AGGREGATE function operates on sets, these GNNs are inherently **permutation equivariant**, which is essencial as there is no natural ordering of node's neighbours [7].

Graph Neural Networks

The Basic GNN

The standard GNN message-passing mechanism is defined as [9]:

$$\mathbf{h}_u^{(k)} = \sigma\left(\mathbf{W}_{\text{self}}^{(k)}\mathbf{h}_u^{(k-1)} + \mathbf{W}_{\text{neigh}}^{(k)}\sum_{v\in\mathcal{N}(u)}\mathbf{h}_v^{(k-1)} + \mathbf{b}^{(k)}\right)$$

So, the aggregation operator of the basic GNN simply takes the sum of the neighbor embeddings:

$$\mathbf{m}_{\mathcal{N}}(u) = \text{AGGREGATE}^{(k)}\left(\{\mathbf{h}_v^{(k-1)}, \forall v \in \mathcal{N}(u)\}\right) = \sum_{v\in\mathcal{N}(u)} h_v^{(k-1)}$$

Graph Neural Networks

Graph Convolutional Network (GCN)

In 2016, Kipf et al. [10] extended this approach by proposing a **symmetric normalization** technique, which scales each neighboring node's contribution based on the degrees of both the target node and each neighboring node:

$$\mathbf{m}_{\mathcal{N}}(u) = \sum_{v \in \mathcal{N}(u)} \frac{h_v}{\sqrt{|\mathcal{N}(u)||\mathcal{N}(v)|}}$$

GCNs became one of the most influential GNN models, combining this symmetric-normalized aggregation operation with self-loop updates:

$$\mathbf{h}_u^{(k)} = \sigma\left(\mathbf{W}^{(k)} \sum_{v \in \mathcal{N}(u) \cup \{u\}} \frac{h_v}{\sqrt{|\mathcal{N}(u)||\mathcal{N}(v)|}}\right)$$

Graph Neural Networks

Graph Attention Network (GAT)

Instead of simply summing or averaging neighboring embeddings, a more sophisticated approach is to apply an **attention mechanism**. Veličković et al. [11] introduced the Graph Attention Network (GAT) model, where the aggregation step becomes a weighted sum of neighbor embeddings:

$$\mathbf{m}_{\mathcal{N}}(u) = \sum_{v \in \mathcal{N}(u)} \alpha_{u,v} h_v$$

In the original GAT model, these attention weights are calculated as:

$$\alpha_{u,v}^{(k)} = \frac{\exp\left(\text{LeakyReLU}\left(\mathbf{a}^{(k)\top}[\mathbf{W}^{(k)}\mathbf{h}_u^{(k-1)} \oplus \mathbf{W}^{(k)}\mathbf{h}_v^{(k-1)}]\right)\right)}{\sum_{v' \in \mathcal{N}(u) \cup \{u\}} \exp\left(\text{LeakyReLU}\left(\mathbf{a}^{(k)\top}[\mathbf{W}^{(k)}\mathbf{h}_u^{(k-1)} \oplus \mathbf{W}^{(k)}\mathbf{h}_{v'}^{(k-1)}]\right)\right)}$$

Graph Neural Networks

Supervised Graph Attention Network (SuperGAT)

To further enhance the effectiveness of GATs, especially in noisy graphs, SuperGAT [12] introduce a mechanism that **supervises the attention** process to help the model prioritize informative edges and down-weight irrelevant connections.

In the original paper, the authors introduce four types of SuperGAT models, each defined by a specific attention mechanism. This work focuses on the variant called MX, which combines two attention mechanisms: the original GAT attention (GO) and the dot-product attention (DO):

$$e_{u,v}^{\mathrm{GO}} = \mathbf{a}^{(k)\top}[\mathbf{W}^{(k)}\mathbf{h}_u^{(k-1)} \oplus \mathbf{W}^{(k)}\mathbf{h}_v^{(k-1)}], \quad e_{u,v}^{\mathrm{DP}} = \left(\mathbf{W}^{(k)}\mathbf{h}_u^{(k-1)}\right)^\top \cdot \mathbf{W}^{(k)}\mathbf{h}_v^{(k-1)}$$

Graph Neural Networks

Supervised Graph Attention Network (SuperGAT)

The combined attention is then computed as:

$$\alpha_{u,v}^{(k)} = \frac{\exp\left(\mathrm{LeakyReLU}\left(e_{u,v}^{\mathrm{MX}}\right)\right)}{\sum_{v' \in \mathcal{N}(u) \cup \{u\}} \exp\left(\mathrm{LeakyReLU}\left(e_{u,v'}^{\mathrm{MX}}\right)\right)},$$

where:

$$e_{u,v}^{\mathrm{MX}} = e_{u,v}^{\mathrm{GO}} \cdot \sigma(e_{u,v}^{\mathrm{DP}})$$

Graph Neural Networks

GraphTransformer

The literature on graph transformers is extensive, addressing various approaches to applying transformers to graph-structured data. We build on Shi et al.'s, [13] approach which extends the traditional self-attention mechanism for graph data, aligning with the principles of the GAT, by adopting the **vanilla multi-head attention** framework from the original transformer architecture [14] and customizing it for graph learning tasks.

The message-passing step is:

$$h_u^{(k)} = \mathbf{W}_1^{(k)} h_u^{(k-1)} + \mathbf{W}_2^{(k)} \sum_{v \in \mathcal{N}(u)} \alpha_{u,v}^{(k)} h_v^{(k-1)}$$

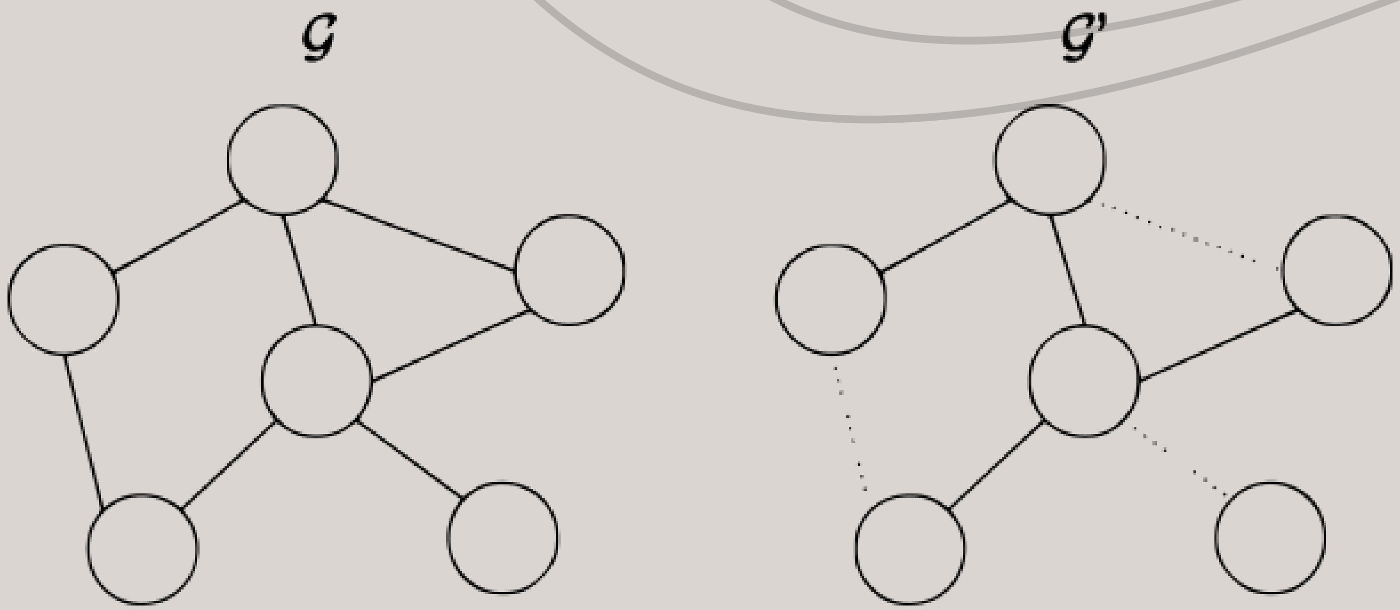
where the attention coefficients are computed using a multi-head dot-product attention mechanism:

$$\alpha_{u,v} = \mathrm{softmax}\left(\frac{(\mathbf{W}_3^{(k)} \mathbf{h}_u^{(k-1)})^\top (\mathbf{W}_4^{(k)} \mathbf{h}_v^{(k-1)})}{\sqrt{d}} \right)$$

Link Prediction

The primary goal of link prediction (LP) is to determine **whether two nodes in a network are likely to form a connection**. The LP problem seeks to identify missing edges in a partial or incomplete version of the graph, denoted as G', which is a subset of the complete graph G.

LP has diverse applications, such as uncovering criminal networks, suggesting connections in social networks, recommending items in systems, and predicting protein interactions in biology.



Previous research has shown that LP models can introduce biases, reinforcing existing inequalities or creating "filter bubbles" [15, 16].

Link Prediction

GNNs have revolutionized LP by integrating graph topology with node and edge features, capturing both local and global context and outperforming traditional structure-based methods.

The Area Under the Receiver (**AUC-ROC**) score is commonly used as a primary evaluation metric for training LP models. However, relying solely on AUC-ROC may not provide a complete picture of a model's performance. To address this, an additional local measure called Vertex-Centric Max Precision Recall at *k* (**VCMPR@*k***) is computed after model training [16, 17]:

$$\text{VCMPR@}k \text{ for vertex } i = \frac{t_i(k)}{max(k, d_i^{\text{test}})},$$

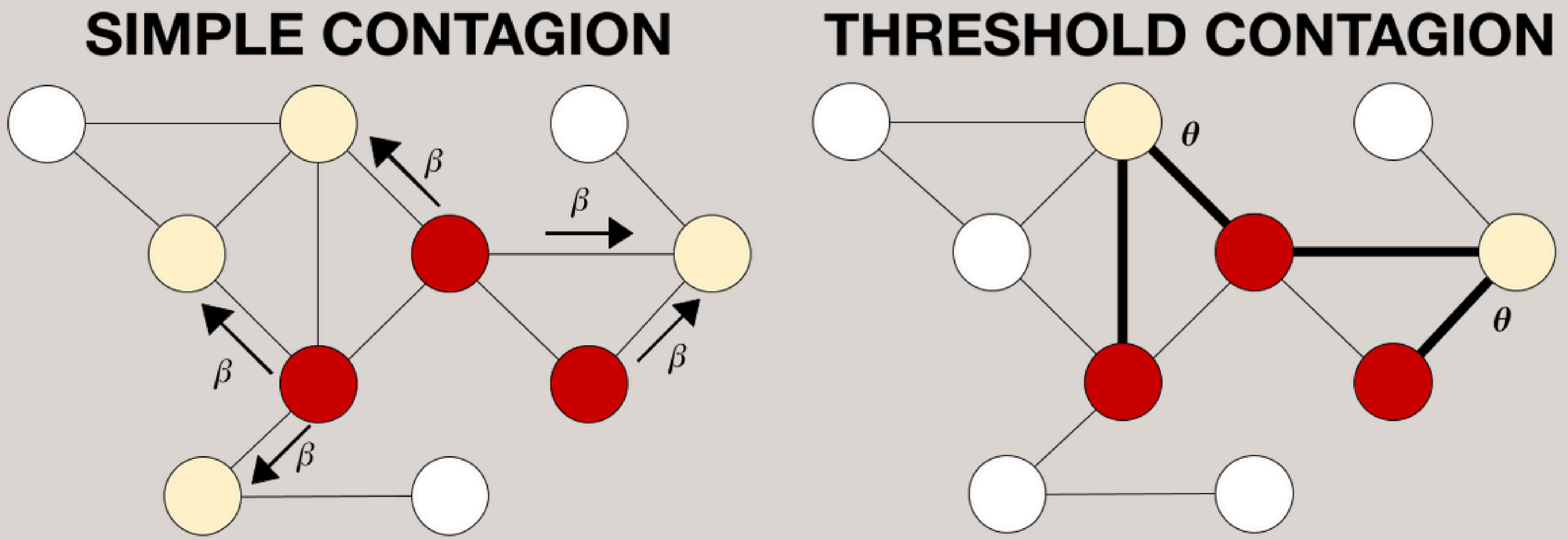
where $t_i(k)$ counts the number of true edges between vertex *i* and other vertices within the top *k* predictions, and d_i^{test} is the degree of vertex *i* in the test set.

Social Contagion Dynamics

Contagion processes on networks, whether modeling disease transmission, information diffusion, or the propagation of social behaviors, can be broadly categorized into two types:

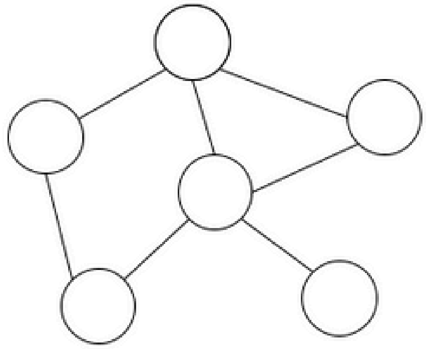
- **Simple contagion**: Spread occurs with a **single interaction** between susceptible and infected nodes, each susceptible having an independent infection probability β. This type of contagion is commonly used to model phenomena such as infectious disease spread, or information access.

- **Complex contagion**: Spread requires **multiple reinforcing interactions**, with a node becoming infected only if its infected neighbors exceed a threshold. This is more representative of social behaviors or innovations, where peer influence and thresholds play a crucial role.



Methodology

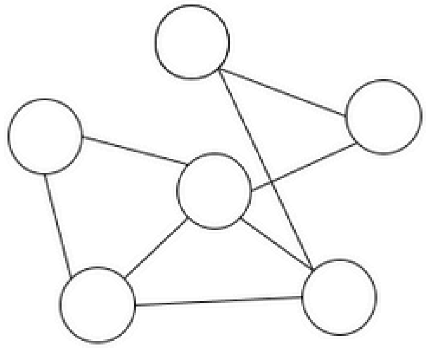
GRAPH ANALYSIS



Graph-Level:
- Average Degree
- Global Clustering
- Gini Indices

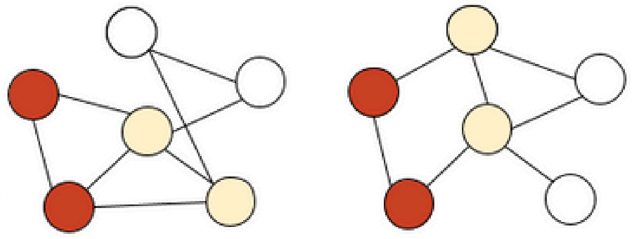
Node-Level:
- Degree Centrality
- Betweenness Centrality
- …

MODEL TRAINING AND LINK-PREDICTION



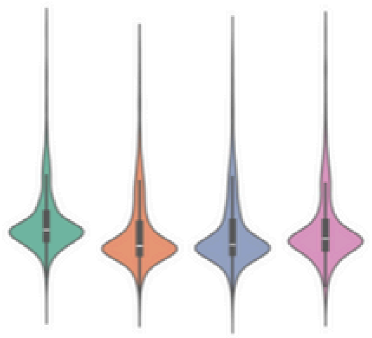
GNN-based LP Models:
- GCN
- GAT
- SuperGAT
- Graph Transformer

SI SIMULATIONS



Contagion dynamics:
- Simple Contagion with β={0.3, 0.5, 0.6, 0.9}.
- Threshold Contagion with $\mu\Theta$ = {0.2, 0.3, 0.4, 0.6} and $\sigma\Theta$ = 0.2

DIFFUSION ANALYSIS



Graph-Level:
- Total Iterations
- Infection Size
- Infection Rate

Node-Level:
- Vulnerability
- Recency

Methodology

Data

Dataset	Category	Nodes	Edges	Features
Cora	Citation Network	2708	10556	1433
CiteSeer	Citation Network	3327	9104	3703
Facebook	Social Network	4039	88234	1283
Wikipedia	Page-Page Network	2405	17981	4973
Twitch ES	Social Network	4648	123412	128
LastFMAsia	Social Network	7624	55612	128

Methodology

Graph Analysis

Node Centrality Measures

- Degree Centrality
- Betweeness Centrality
- Eigenvector Centrality
- Complex-Path Centrality
- Diffusion Centrality

Graph Topological Measures

- Average Degree
- Global Clustering Coefficient
- Gini Coefficient of the centrality measures

Methodology
Model Training and LP

For each LP model:

- We train **10 independent versions** using different random train-validation-test split, across two GNN architectures (two-layer and three-layer models).
- Adam optimizer with a fixed learning rate of 0.01, optimizing model performance through the Binary Cross-Entropy with Logits loss function (BCEWithLogitsLoss).
- Primary metric AUC-ROC supplemented by VCMPR@*k* local measure to capture nuanced model performance.

Methodology

Model Training and LP

For each predicted and real networks pairs we conduct 100 social contagion simulations. We employ the **Susceptible-Infected (SI) epidemic modeling**:

- **Initialitzation**:
    - Simple contagion: Randomly select one node to be initially infected (I).
    - Complex contagion: Randomly select an initial node for infection. Subsequently, the node's neighbors are then examined, and a number of neighbors are infected based on the node's specific threshold θ^u.

Methodology

Model Training and LP

- **Infection Dynamics**: At each discrete time step, implement infection propagation:
  - Simple Contagion: Each infected node attempts to infect its susceptible neighbors with a fixed probability $\beta \in \{0.3, 0.5, 0.6, 0.9\}$.
  - Complex Contagion: A susceptible node becomes infected only when a sufficient number of its neighbors are already infected, with node thresholds drawn from a truncated normal distribution with mean $\mu_\theta \in \{0.2, 0.3, 0.4, 0.6\}$ and standard deviation $\sigma_\theta = 0.2$.

Methodology

Model Training and LP

- **State Transition**: Update the infection status of all nodes based on the specific contagion rules.

- **Repeat**: Continue until the infection **stabilizes**, *i.e.*, no new nodes become infected in the next time step.

Methodology

Social Contagion Metrics

- *Iterations*: The number of time steps taken for the contagion to stabilize.
- *Infection Size*: The proportion of nodes infected by the end of the process.
- *Infection Rate*: The speed of diffusion, calculated as:

$$\text{Infection Rate} = \frac{\text{Infection Size} \times \text{Number of Nodes}}{\text{Iterations}}$$

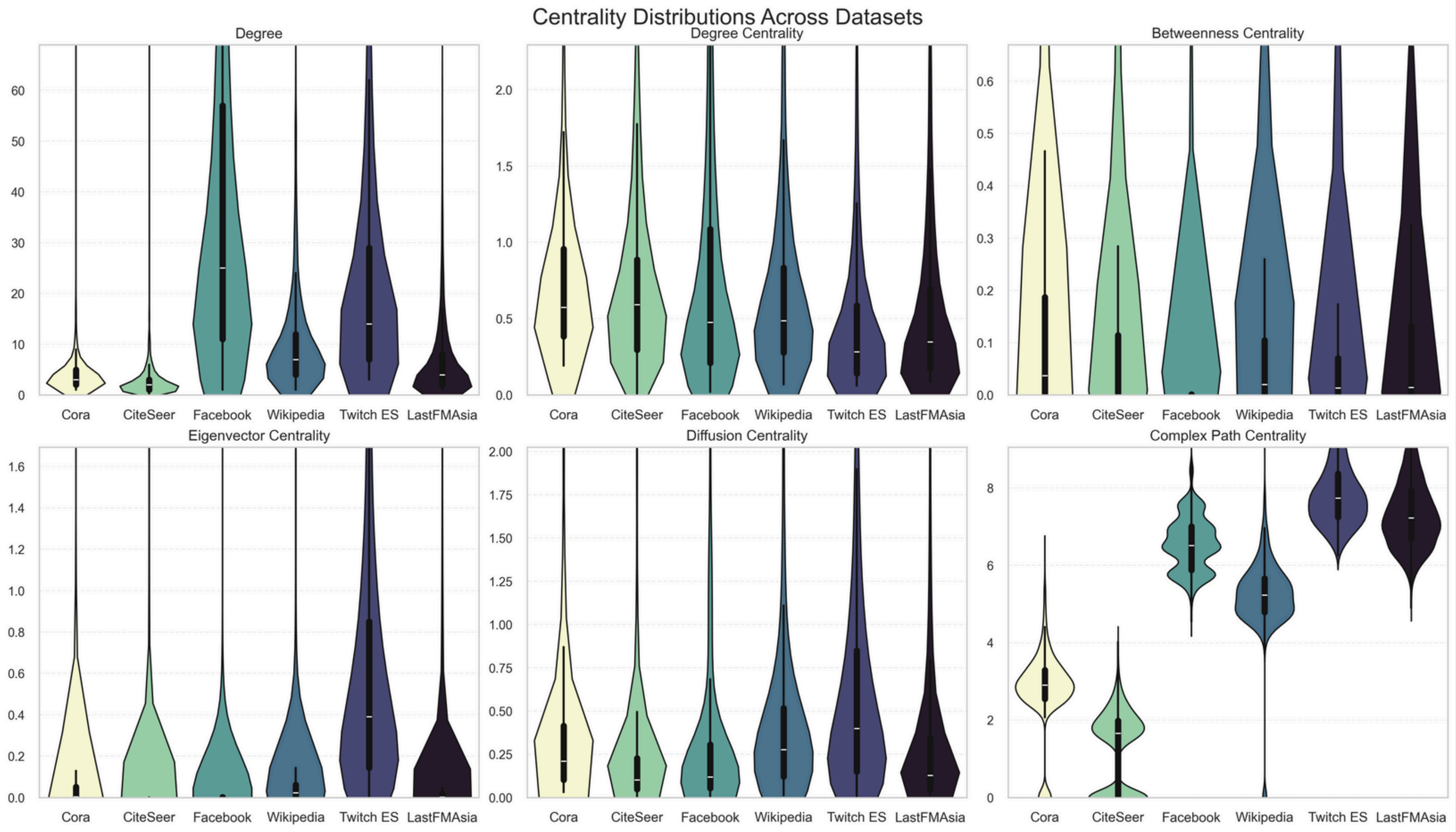
- V*ulnerability* of a node *u*: The proportion of simulations in which node *u* becomes infected.

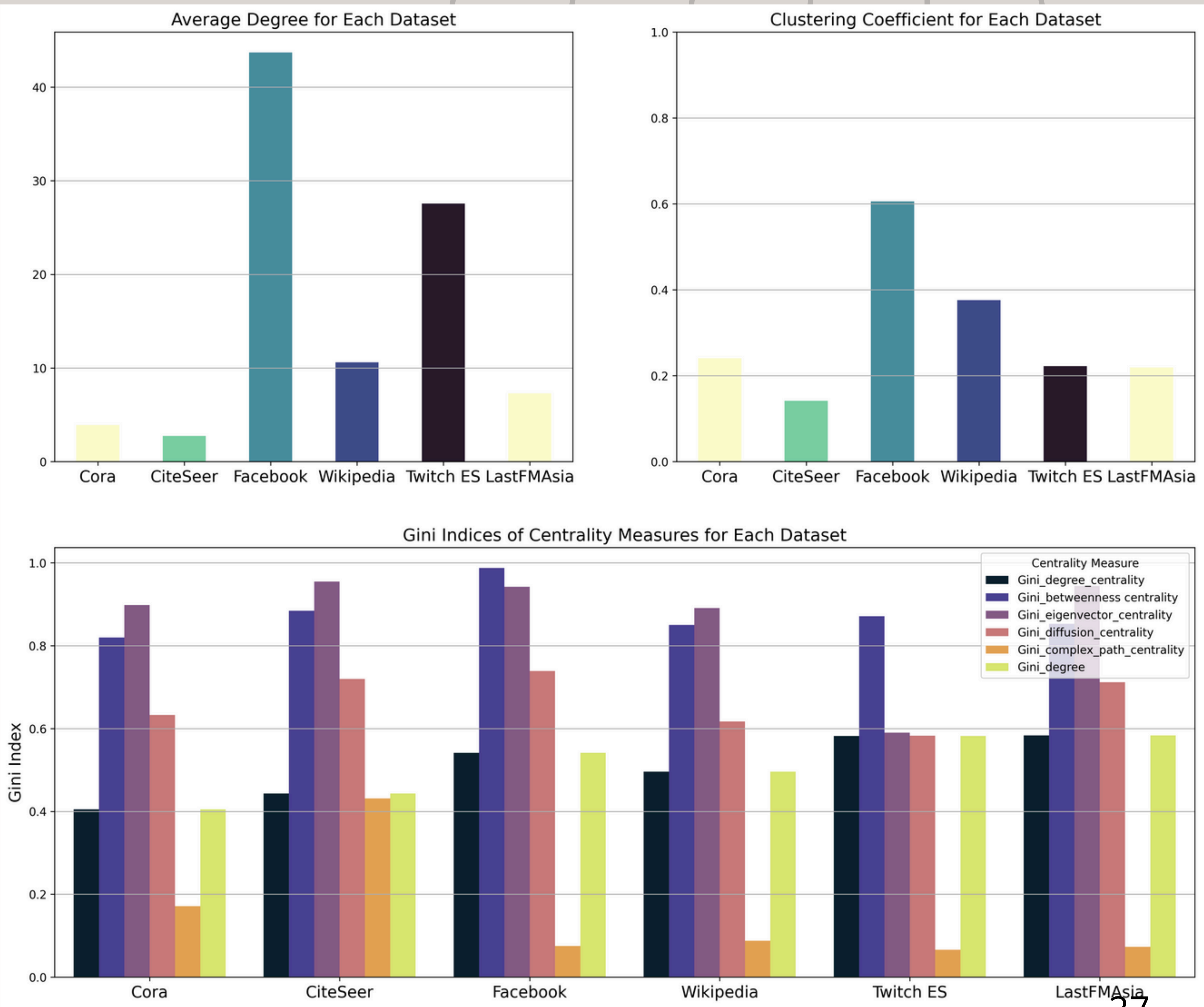
- *Recency* of a node *u:* Metric that quantifies how quickly a node typically becomes infected during the contagion process.

$$\text{Recency}(u) = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{\text{timestep}_i(u) + 1}$$

Results

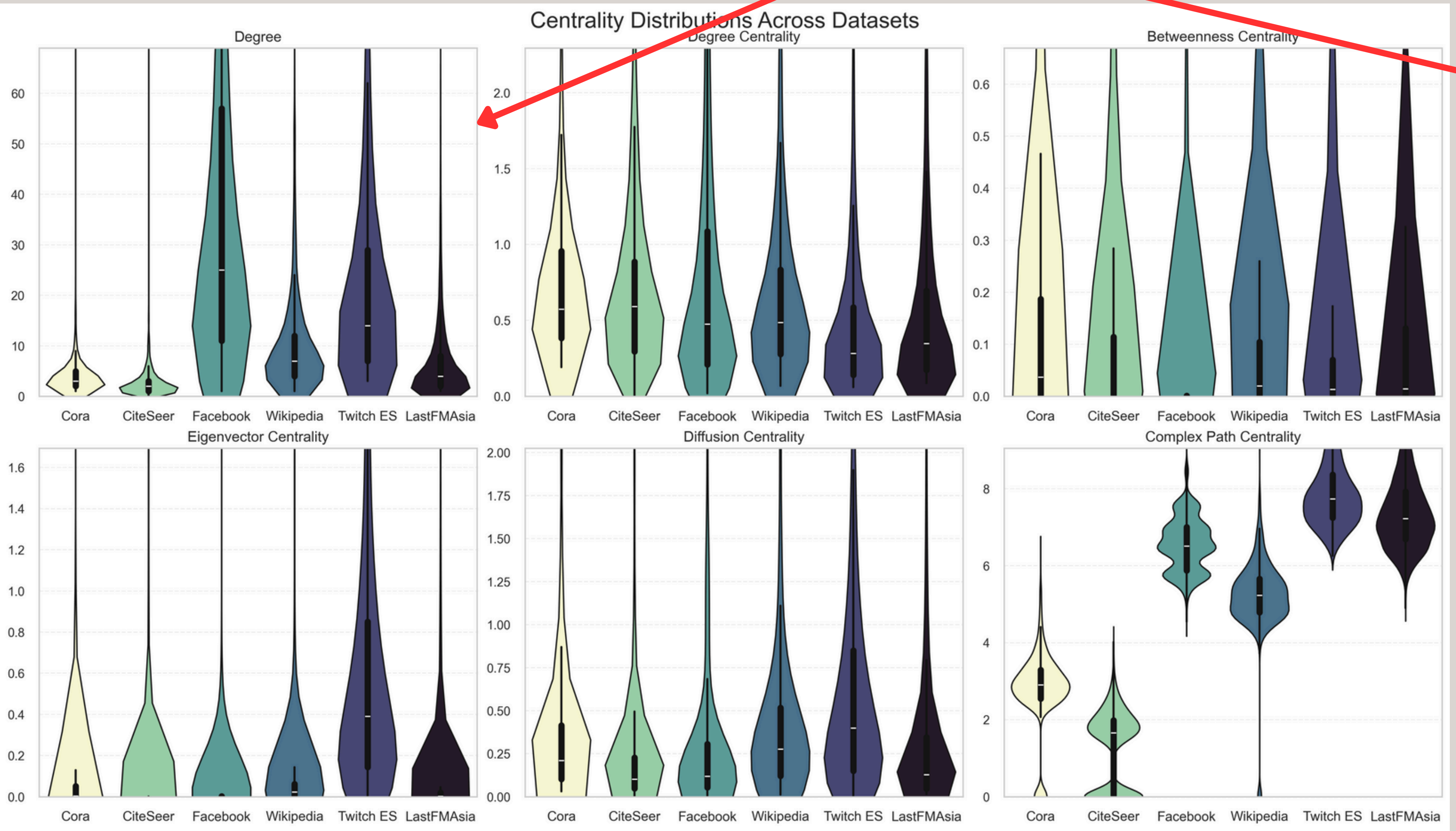
Initial Exploratory Analysis

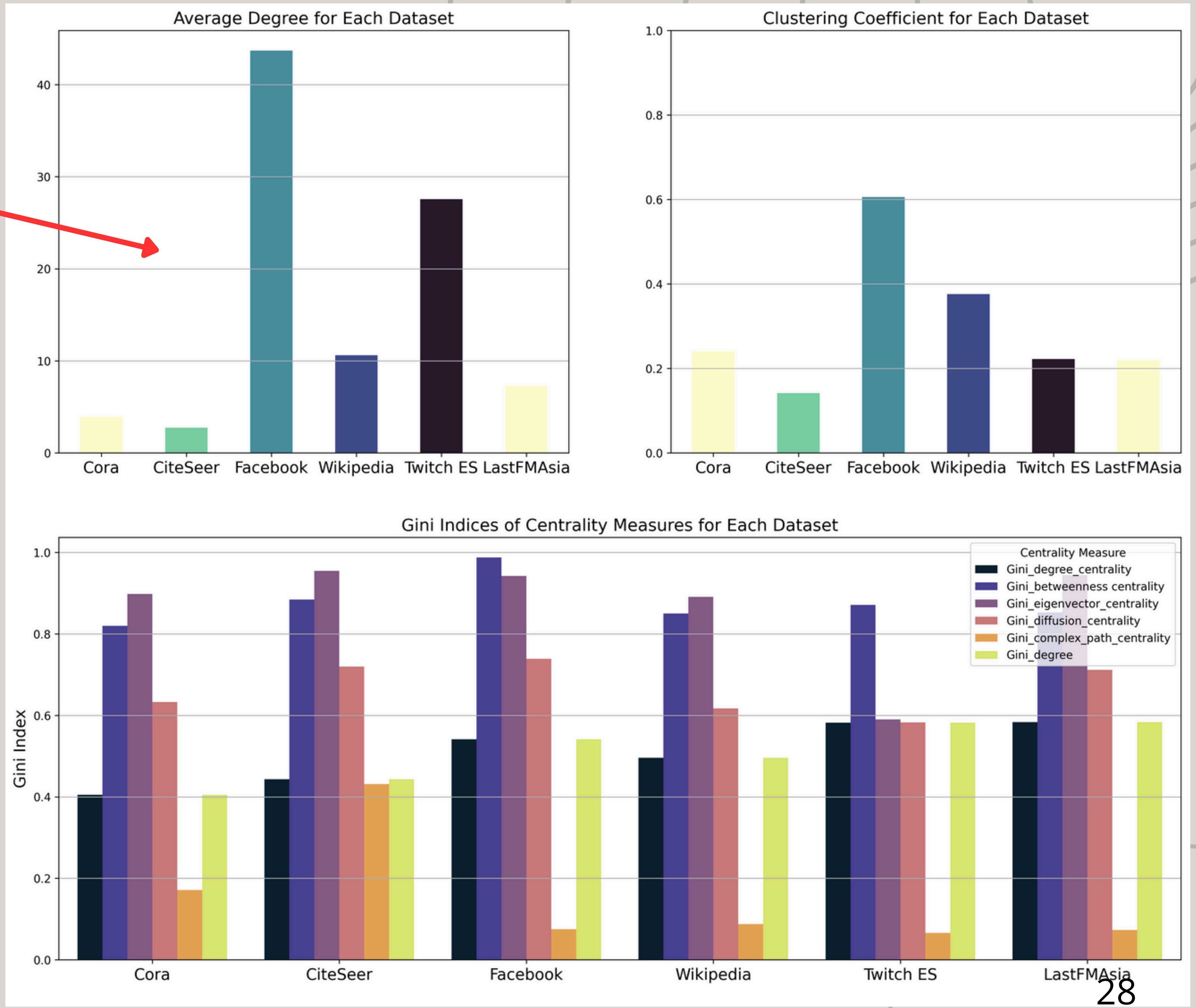




Initial Exploratory Analysis

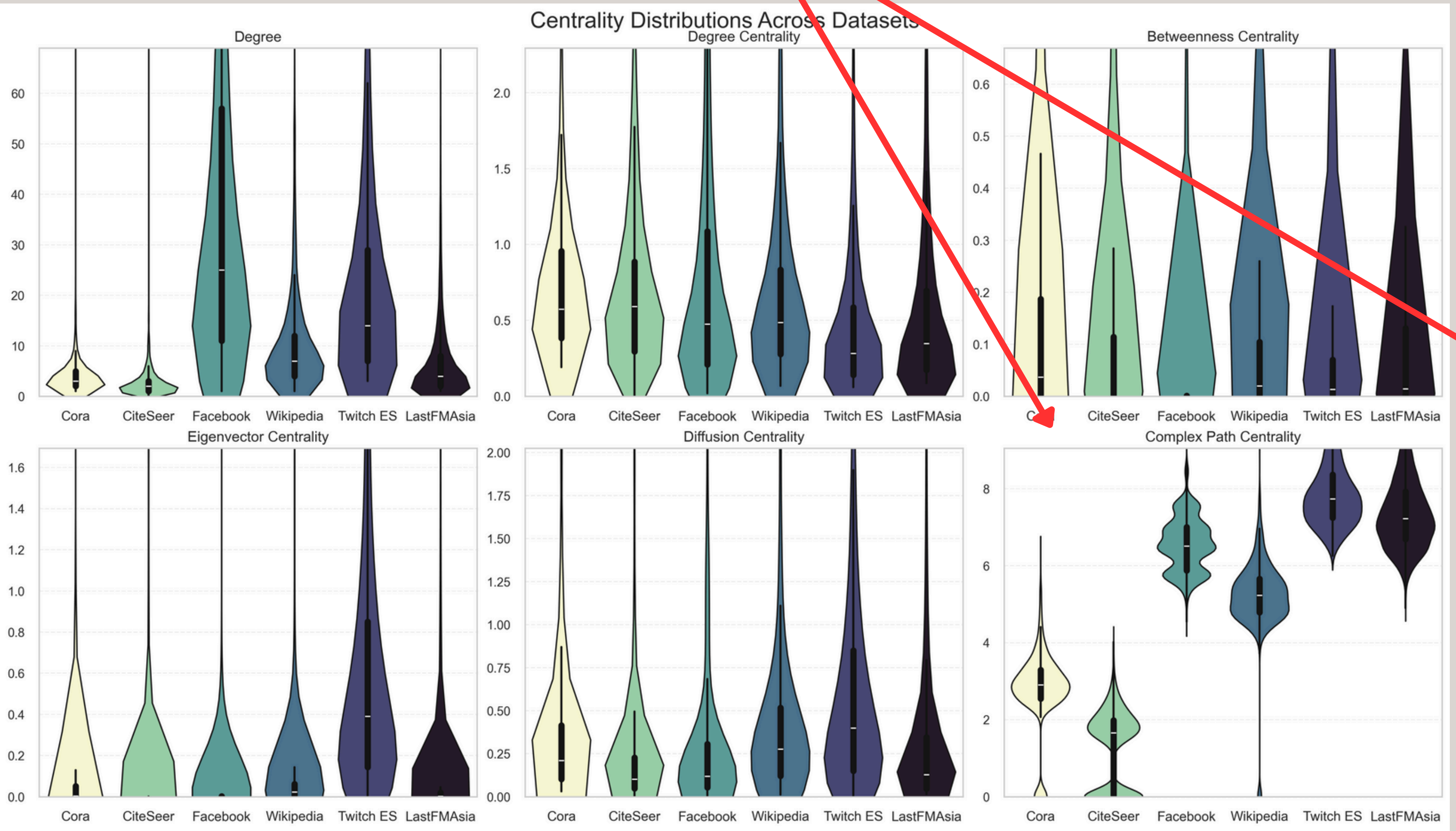
Facebook and *Twitch ES* stand out as networks with higher degree distributions, while *CiteSeer* having the lowest.

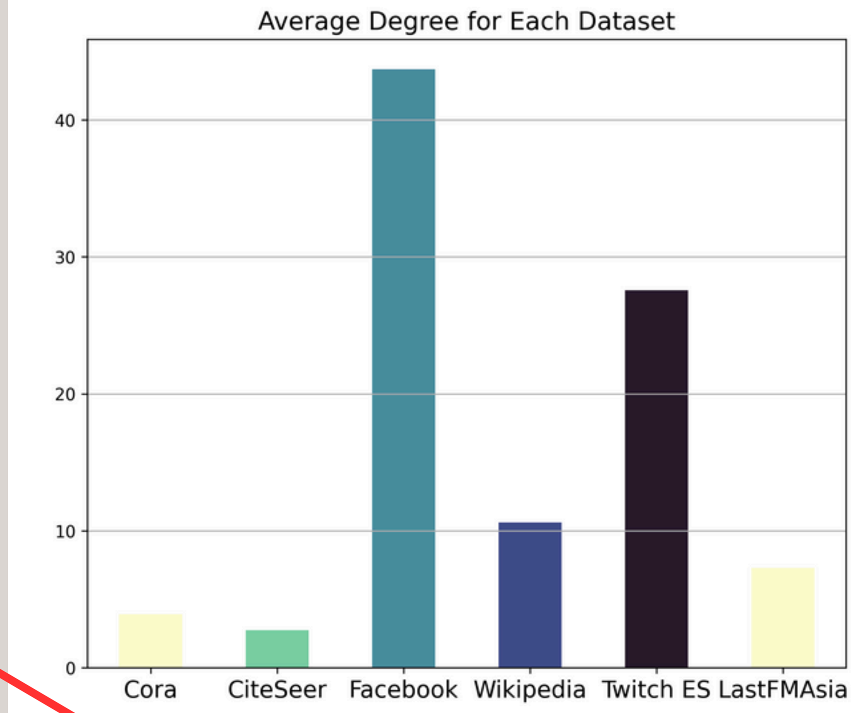


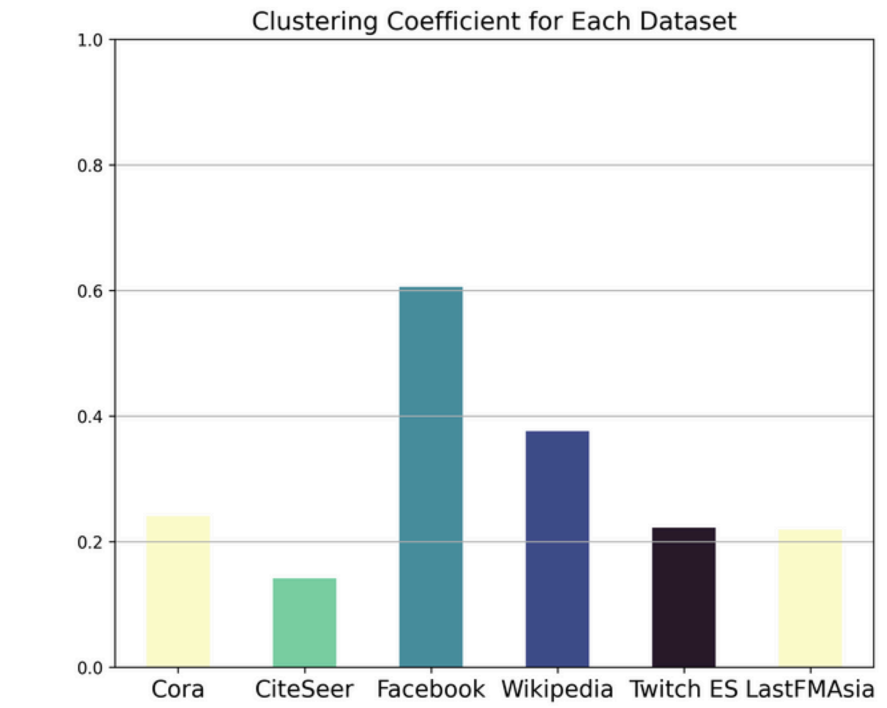


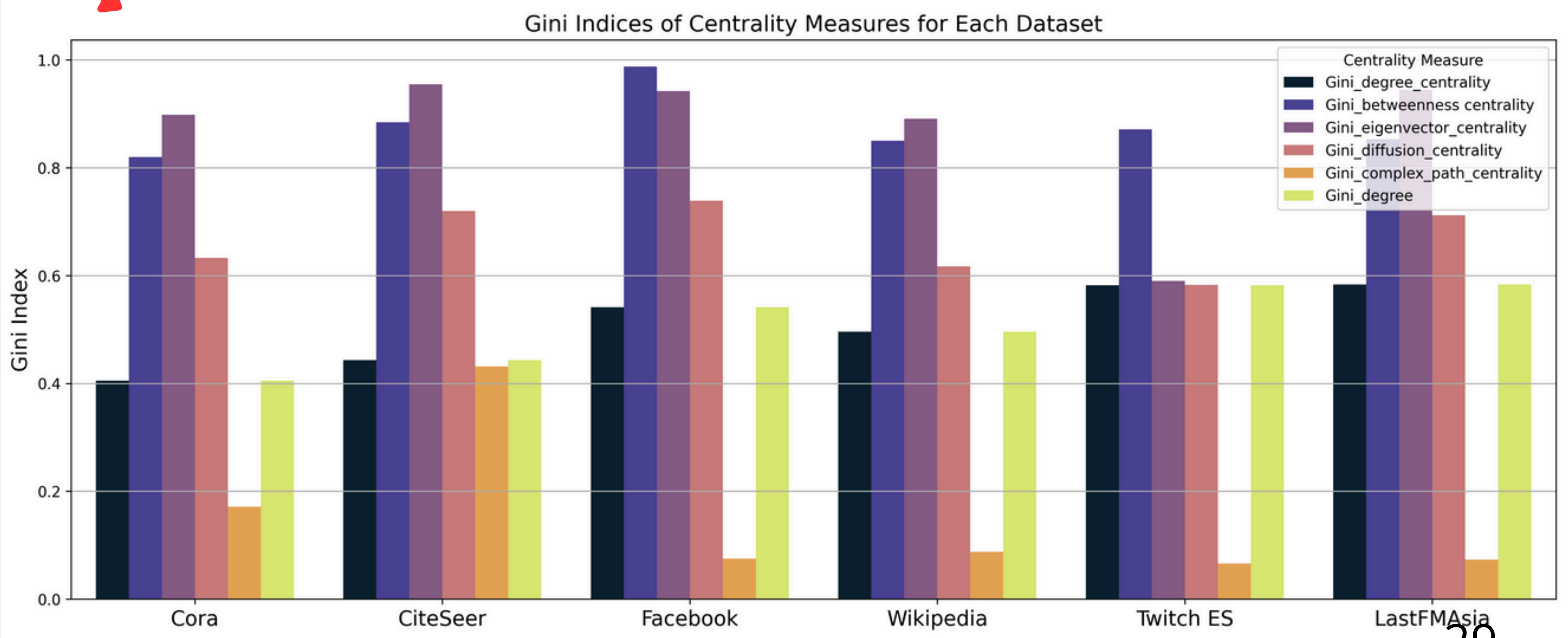
Initial Exploratory Analysis

The distributions of *Complex Path Centrality* are particularly diverse and egalitarian.

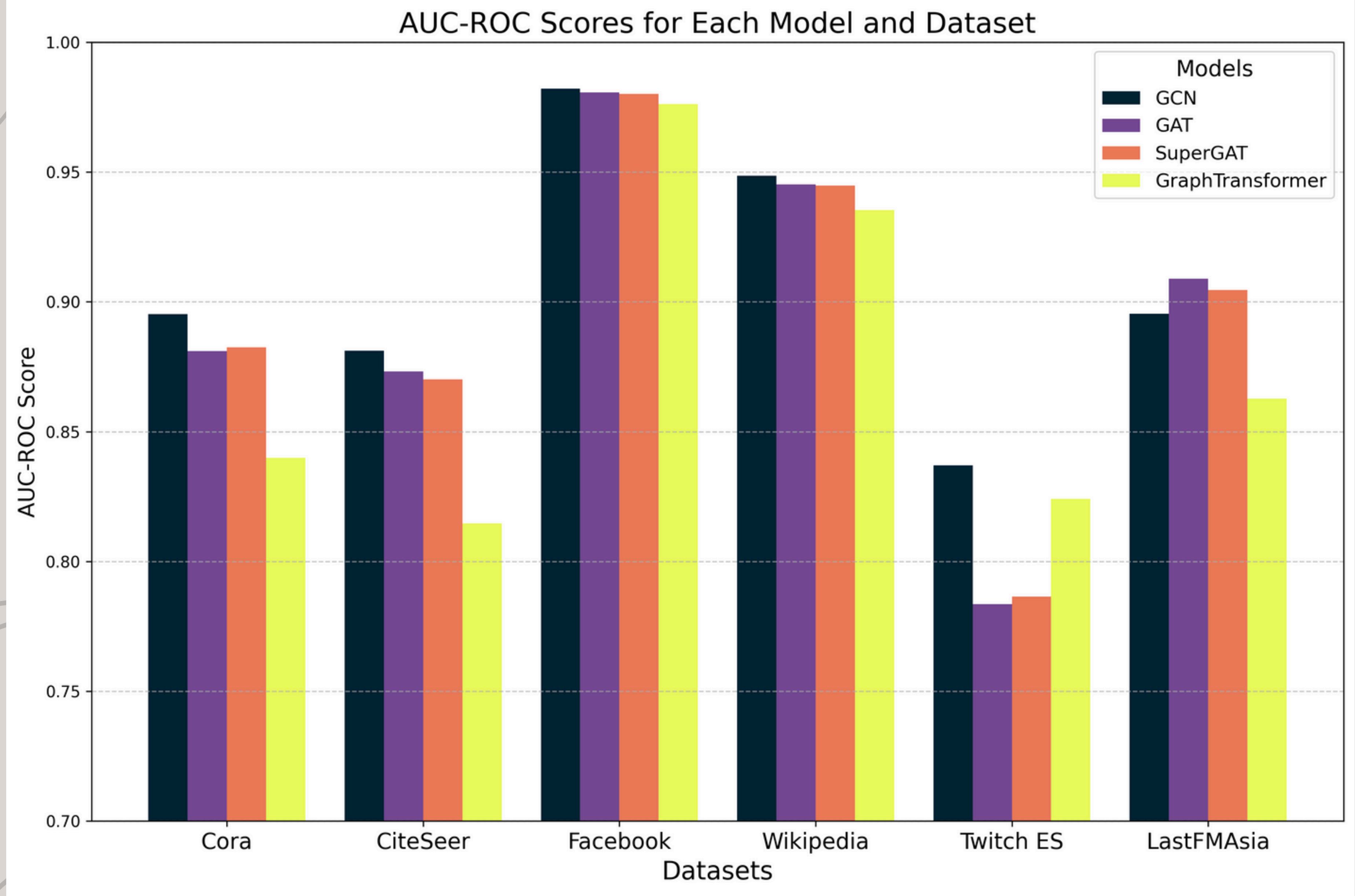


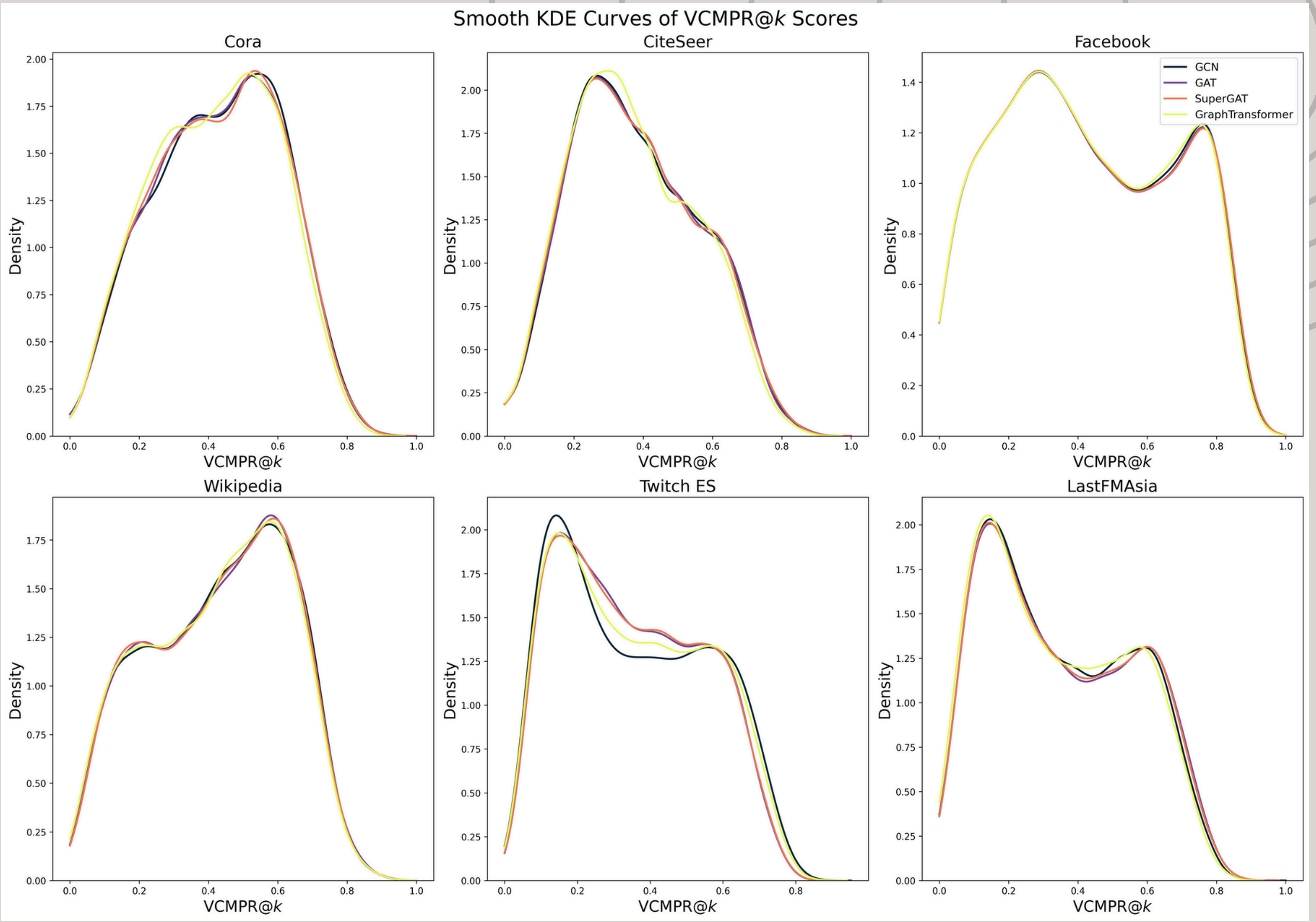






LP Performance Evaluation



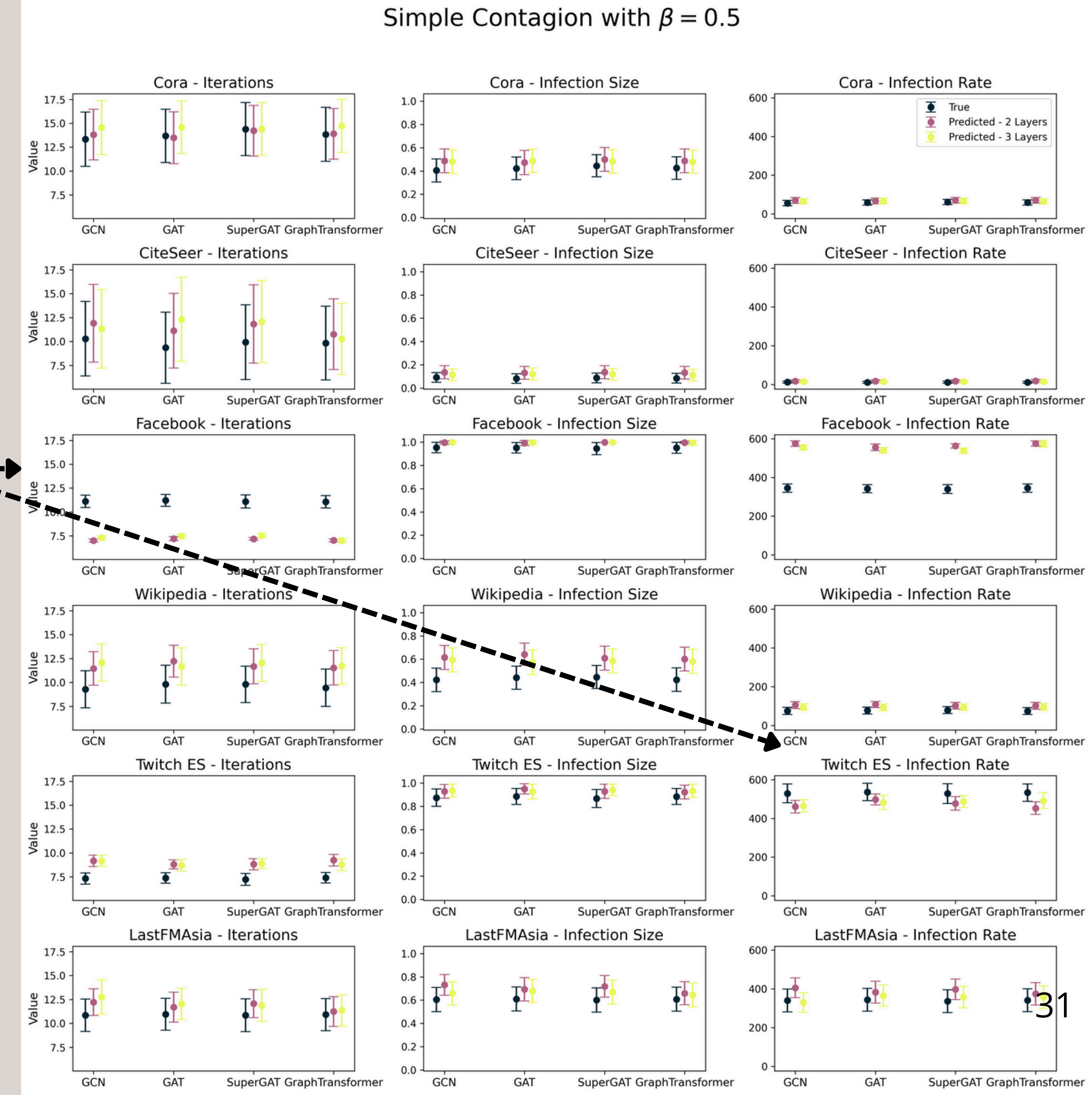


- In AUC-ROC, GCN clearly outperformes the other models.
- VCMPR@k curves are skewed, with little variation between models.

Simple Contagion

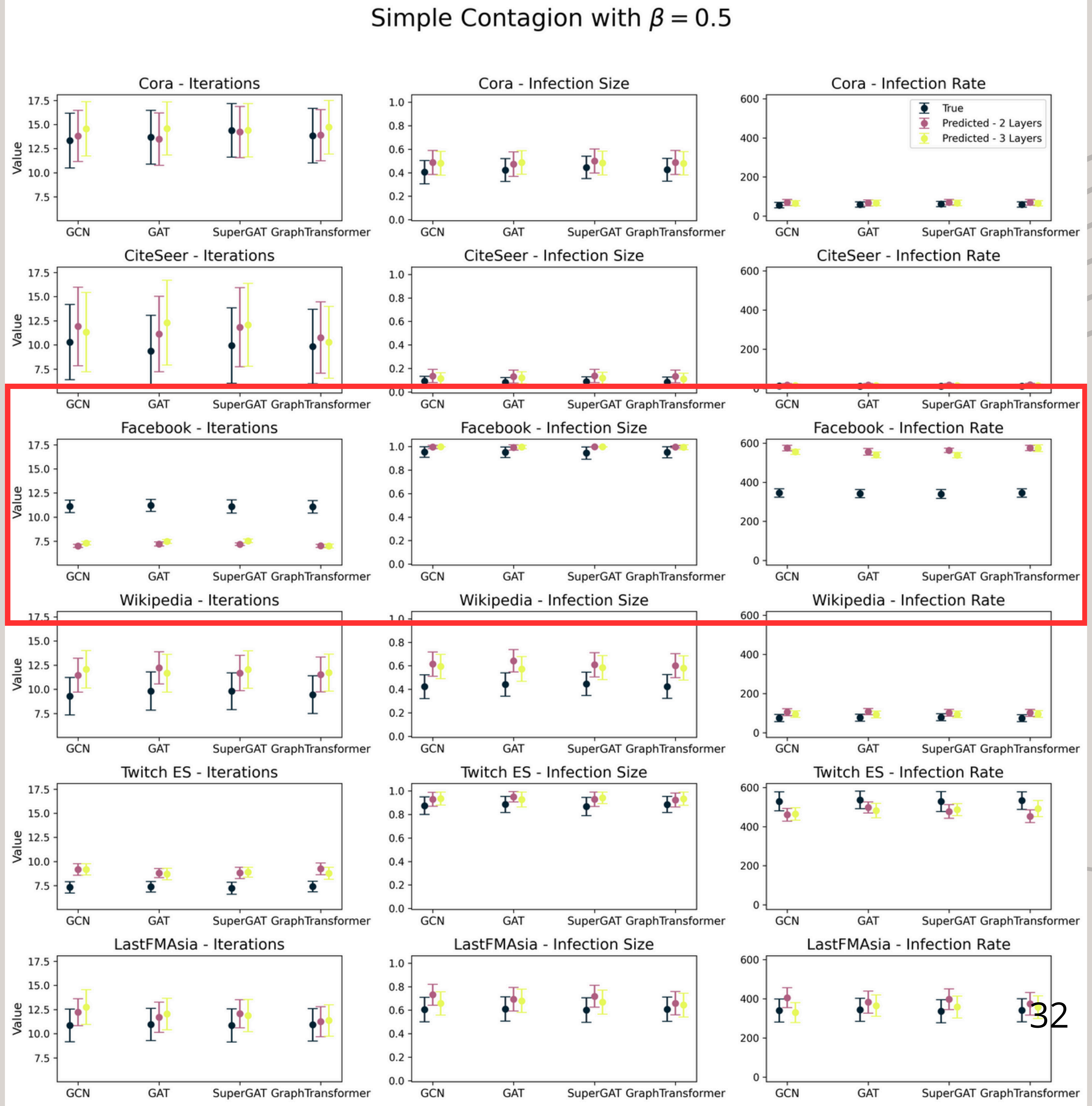
Graph-Level

- **Predicted networks generally exhibit higher values** for all contagion metrics compared to their real counterparts across all datasets, with the exception of the *Facebook* dataset for *Iterations,* and the *Twitch ES* dataset for *Infection Rate.*



Simple Contagion
Graph-Level

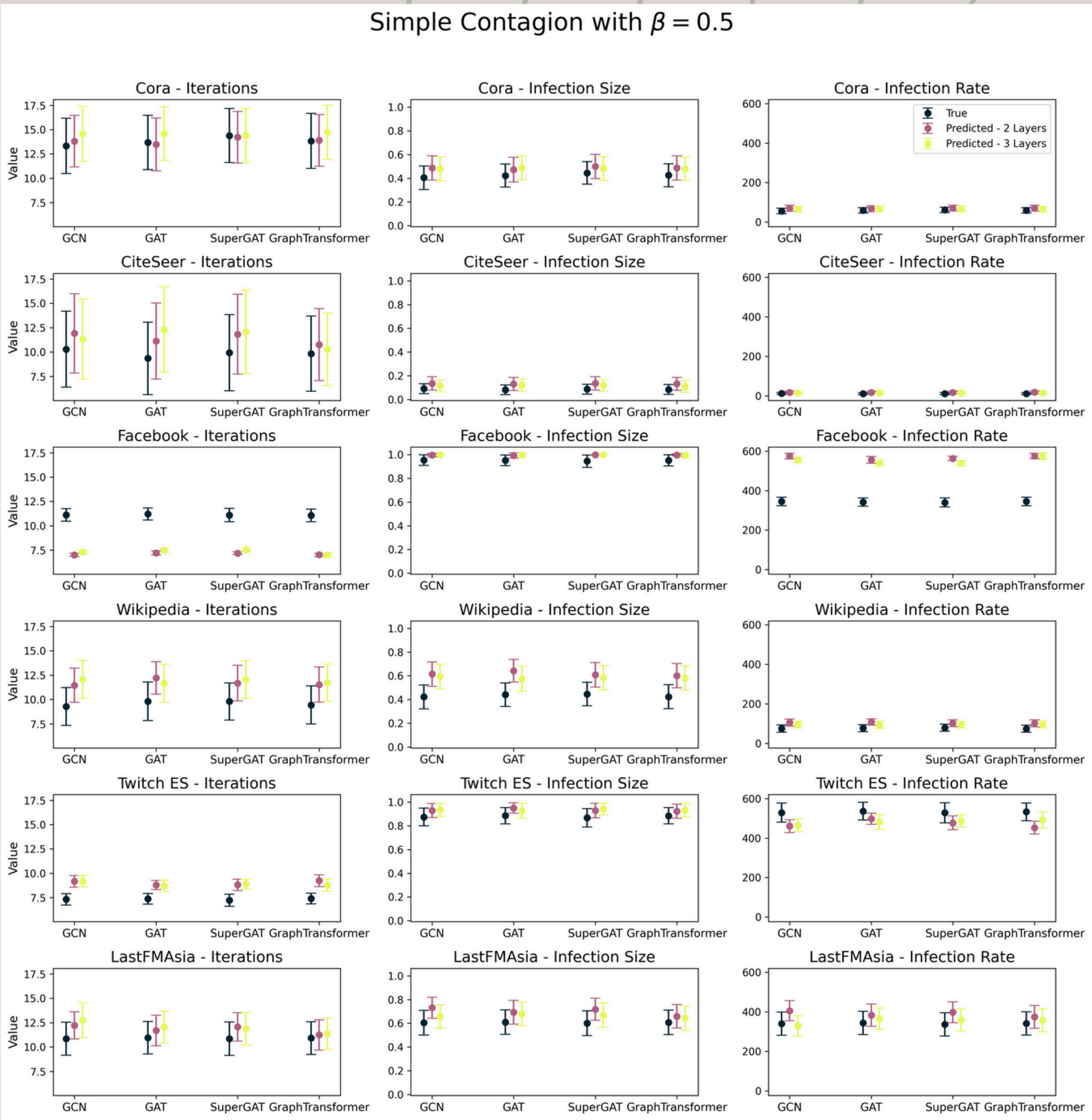
- *Facebook* (highest *Average Degree* and *Clustering Coefficient*), within less than ten iterations the spread of information in the predicted network rapidly stabilizes, effectively reaching the entire population.



Simple Contagion

Graph-Level

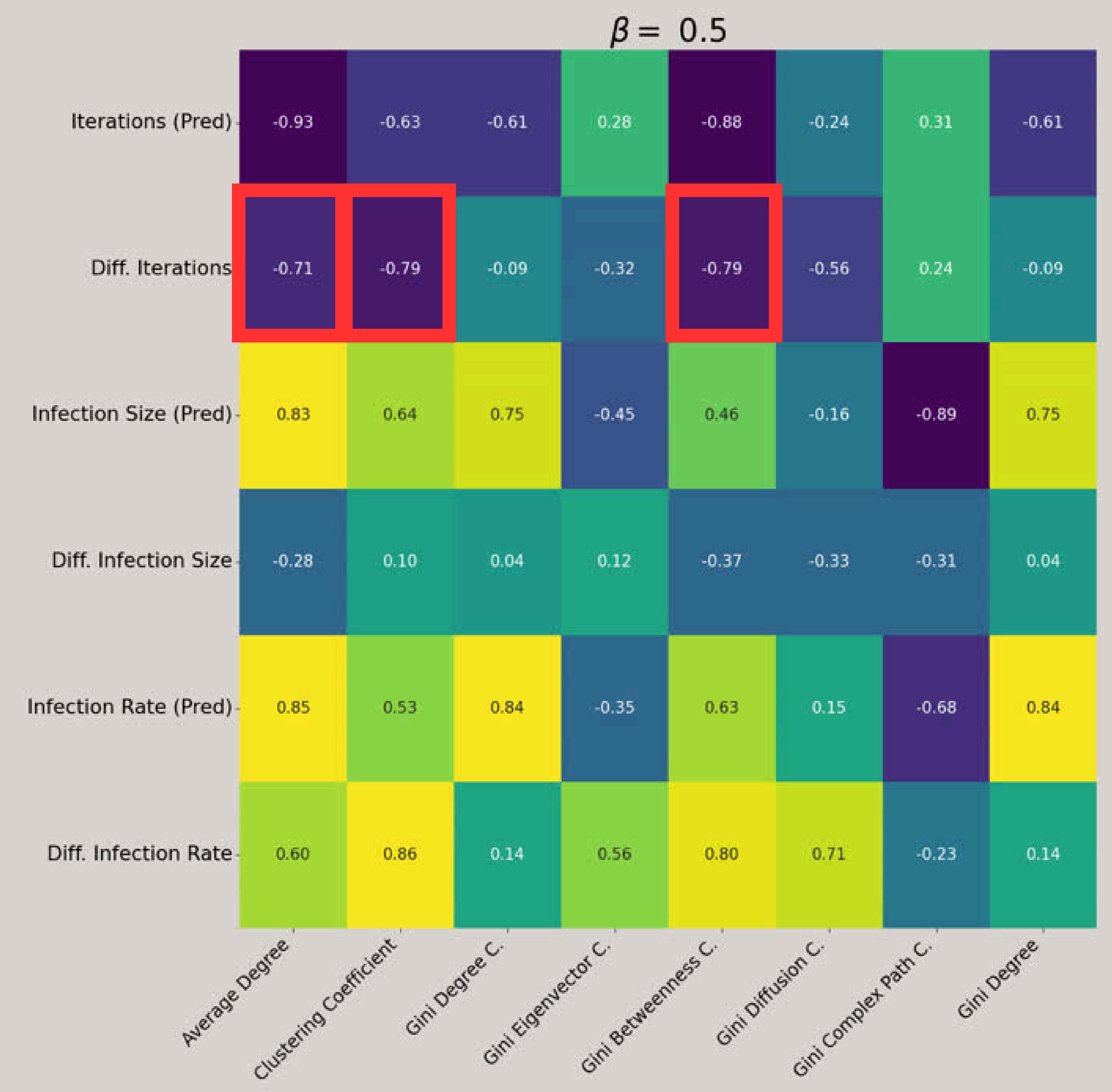
- **Minimal variation across different models and model depths**, suggesting that the structural changes introduced by the various LP algorithms lead to largely consistent social diffusion patterns.



Simple Contagion

Graph-Level Correlations

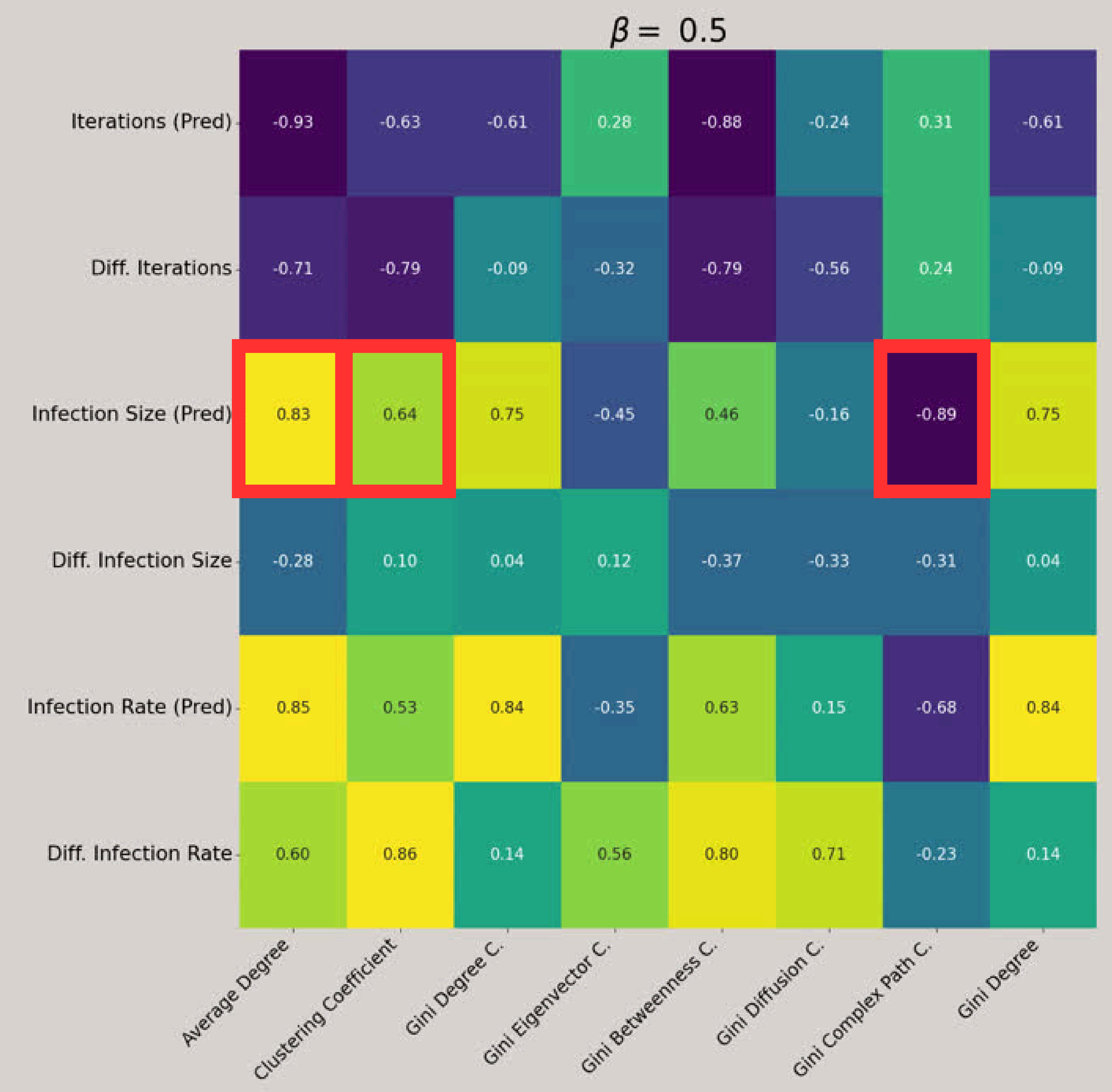
Diff. Iterations shows strong negative correlations with *Average Degree*, *Clustering Coefficient*, and *Gini Betweenness Centrality*. This explains *Facebook* inverse behaviour.



Simple Contagion

Graph-Level Correlations

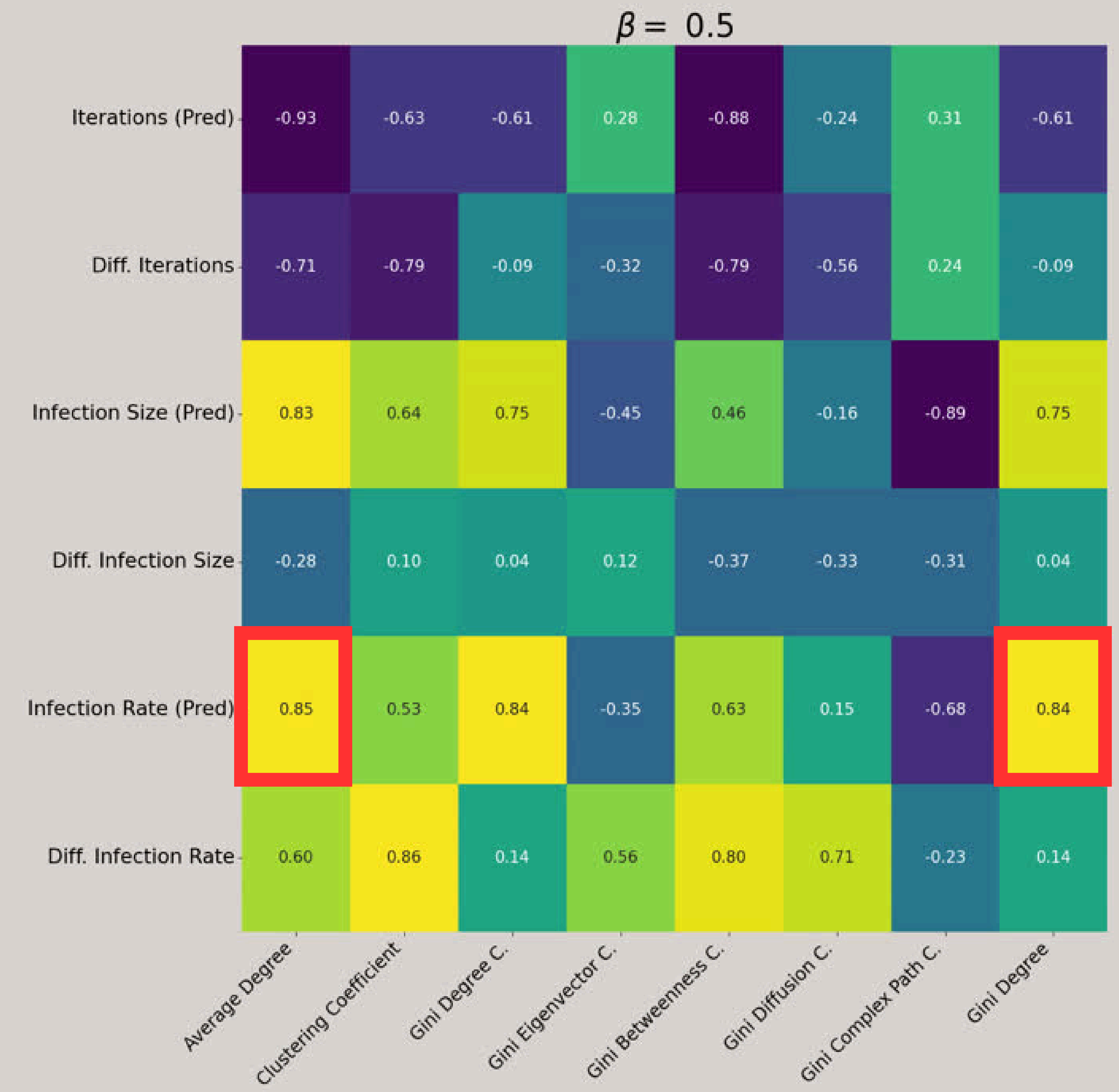
Infection Size shows strong positive correlations with *Average Degree* and *Clustering Coefficient*, while displaying a strong negative correlation with *Gini Complex Path Centrality*. This indicates that **denser networks with more uniform** *Complex Path Centrality* **distributions experience larger contagion spread**.



Simple Contagion

Graph-Level Correlations

Infection Rate positively correlates with both *Average Degree* and *Gini Degree*. This suggests that **simple contagion processes are accelerated in networks where high average degree is driven by a few highly connected nodes**, rather than by uniformly high connectivity across all nodes. The behavior of the *Twitch ES* exemplifies this phenomenon.

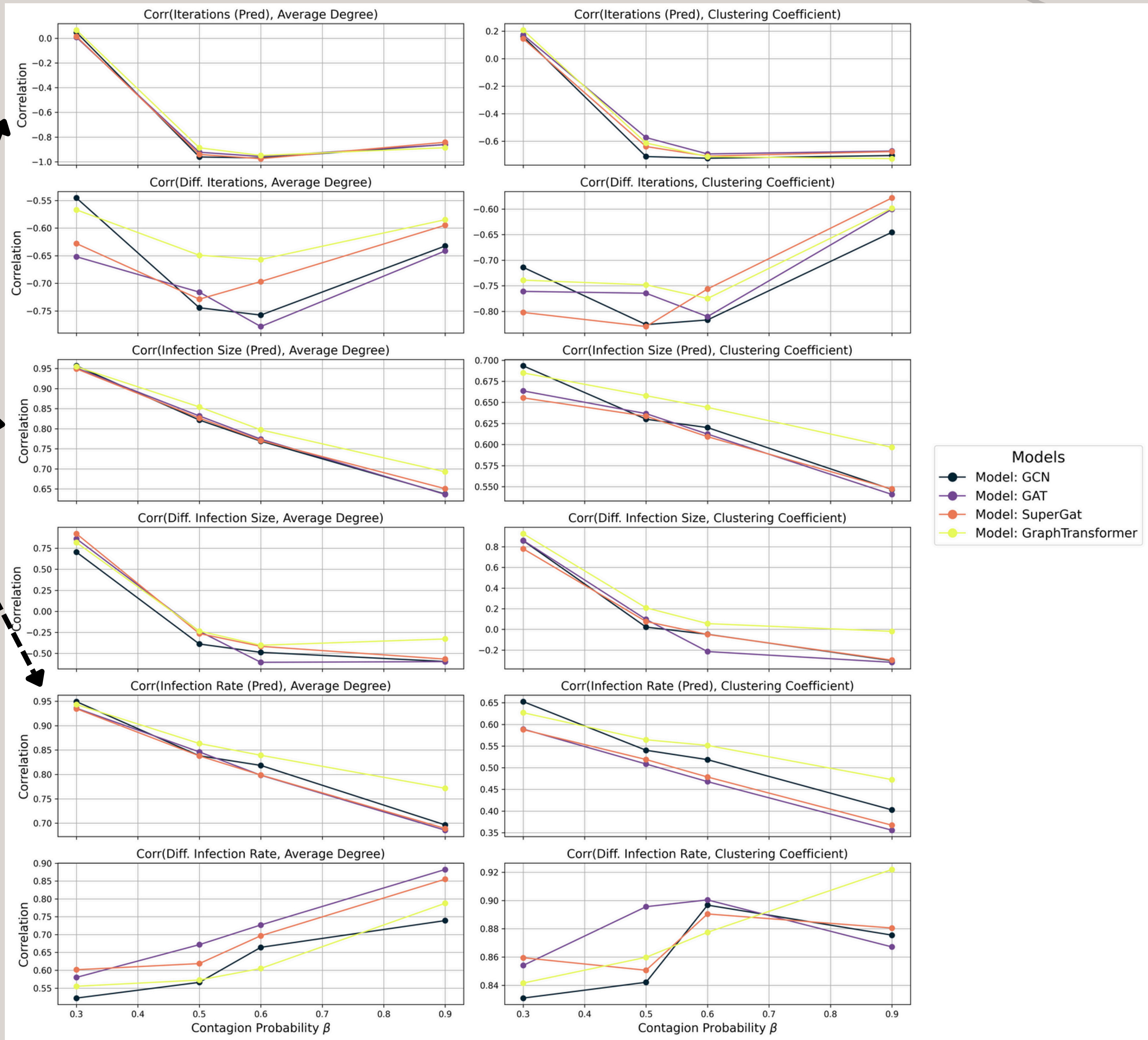


Simple Contagion

Graph-Level Correlations

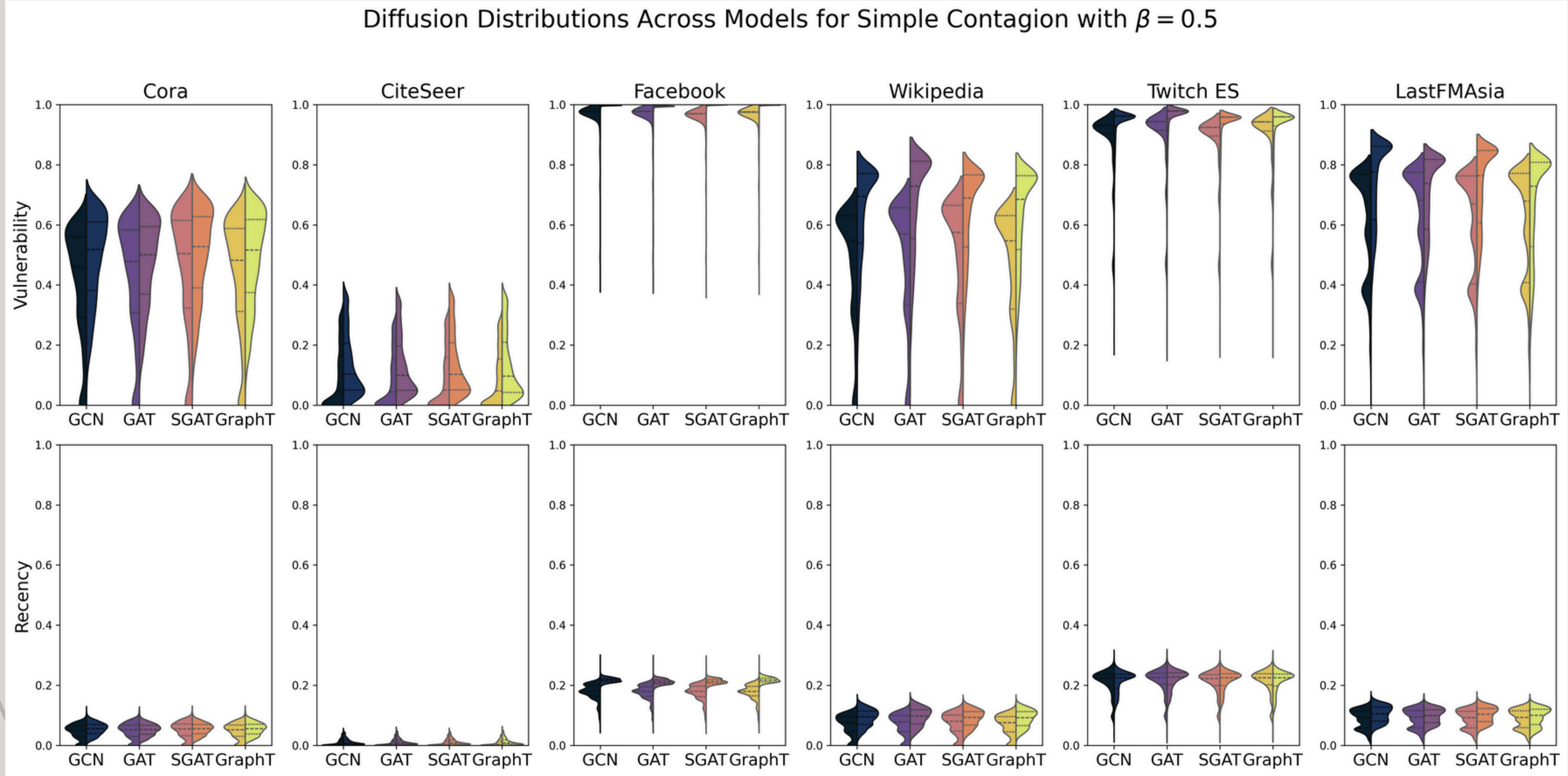
At higher values of β, the contagion process becomes more accessible to all nodes, regardless of their position in the network. This reduces the influence of high connectivity on the *Infection Size* and *Infection Rate*.

Additionally, a strong negative correlation between the topological features and *Iterations* is observed. This suggests that connectivity is critical for accelerating the stabilization of contagion.



Simple Contagion

Node-Level



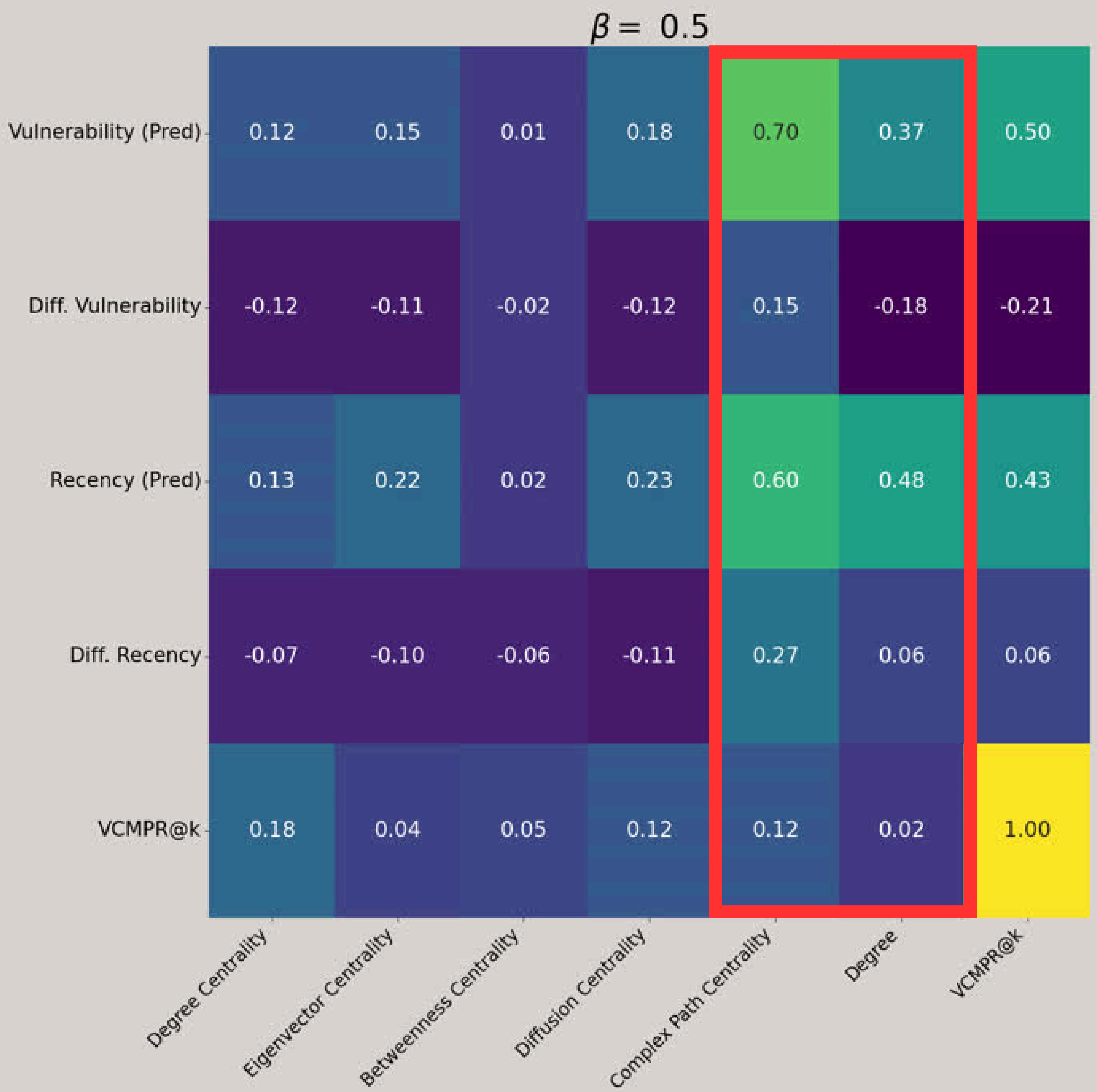
LP models tend to increase nodes' susceptibility to contagion, aligning with the observed graph-level diffusion patterns.

Simple Contagion

Node-Level Correlations

Unlike the graph-level correlations, these node-level correlations exhibit notably weaker relationships.

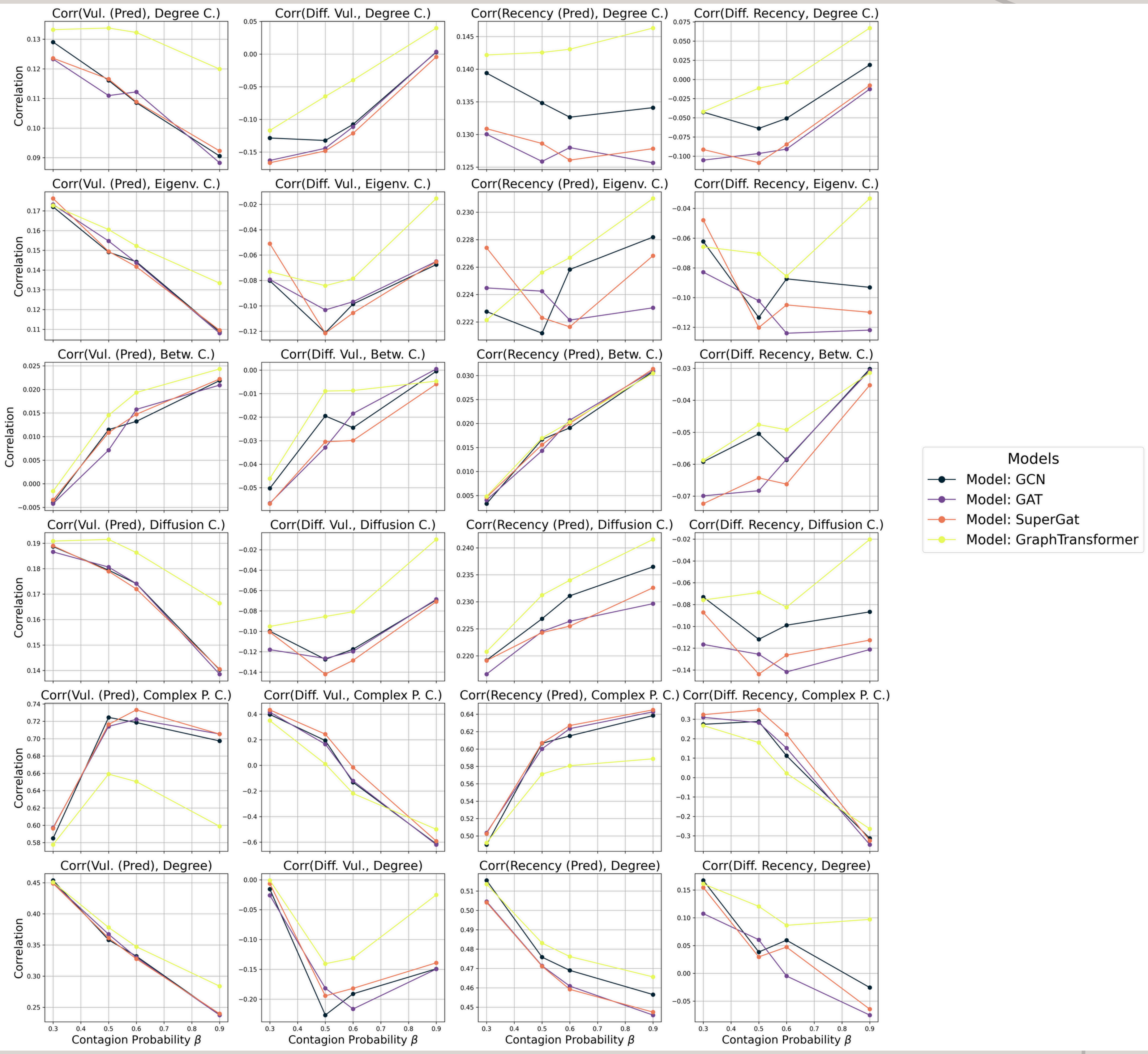
Among the node features analyzed, *Complex Path Centrality* **and** *Degree* **emerge as the most influential factors** in determining a node's susceptibility to the contagion process.



Simple Contagion
Node-Level Correlations

Correlation between *Vulnerability*/*Recency* and most centrality measures decreases as the probability β increases.

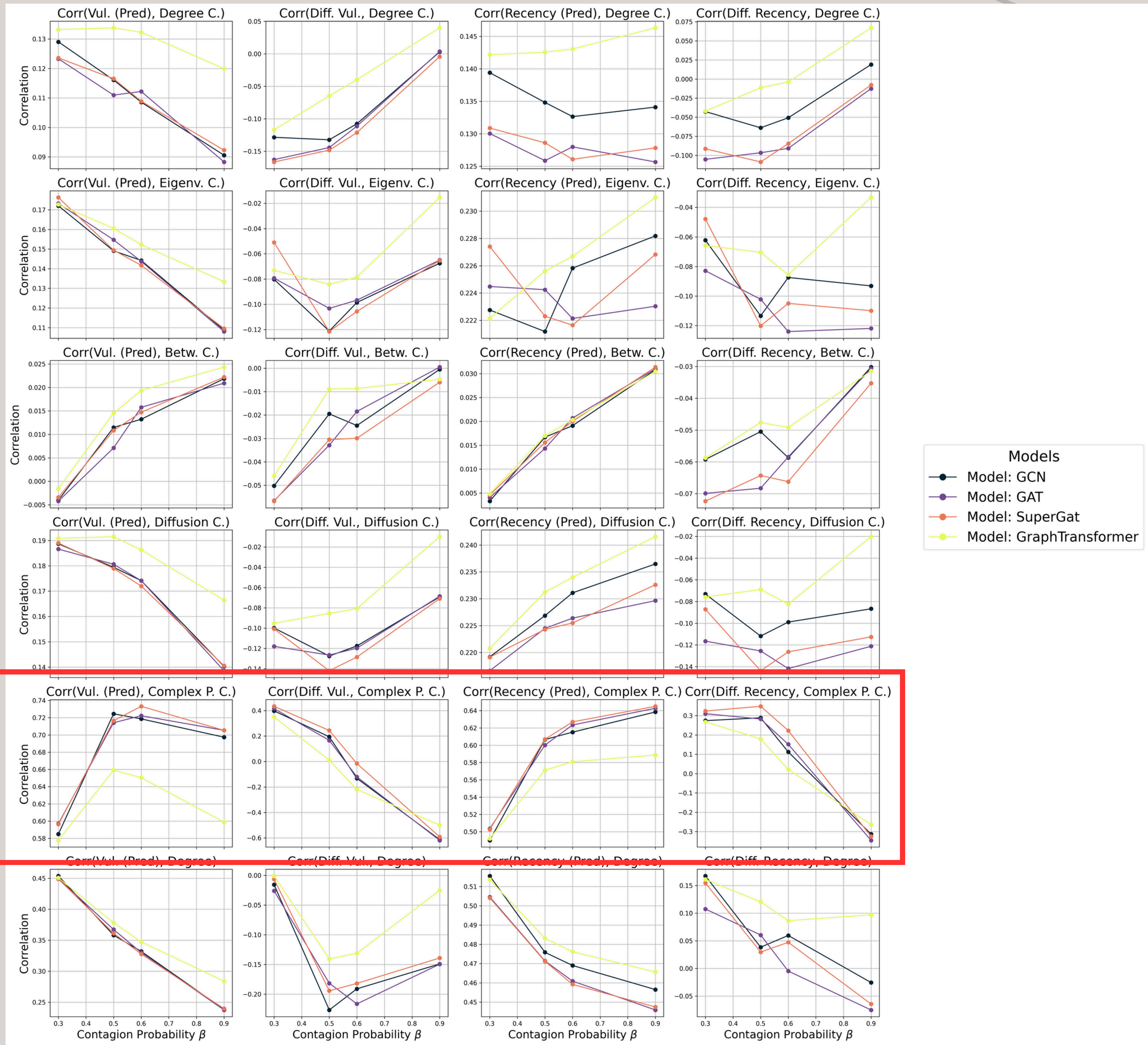
This observation aligns with the phenomenon discussed in the previous section: As the probability of contagion rises, nodes become more susceptible to infection, diminishing the effect of neighborhood characteristics, connection quality, or the number of neighbors on infection likelihood.



Simple Contagion

Node-Level Correlations

Notably, the correlation of *Complex Path Centrality* intensifies as the probability increases, reaching stability around β = 0.5. This effect is particularly attenuated in the Graph Transformer model.

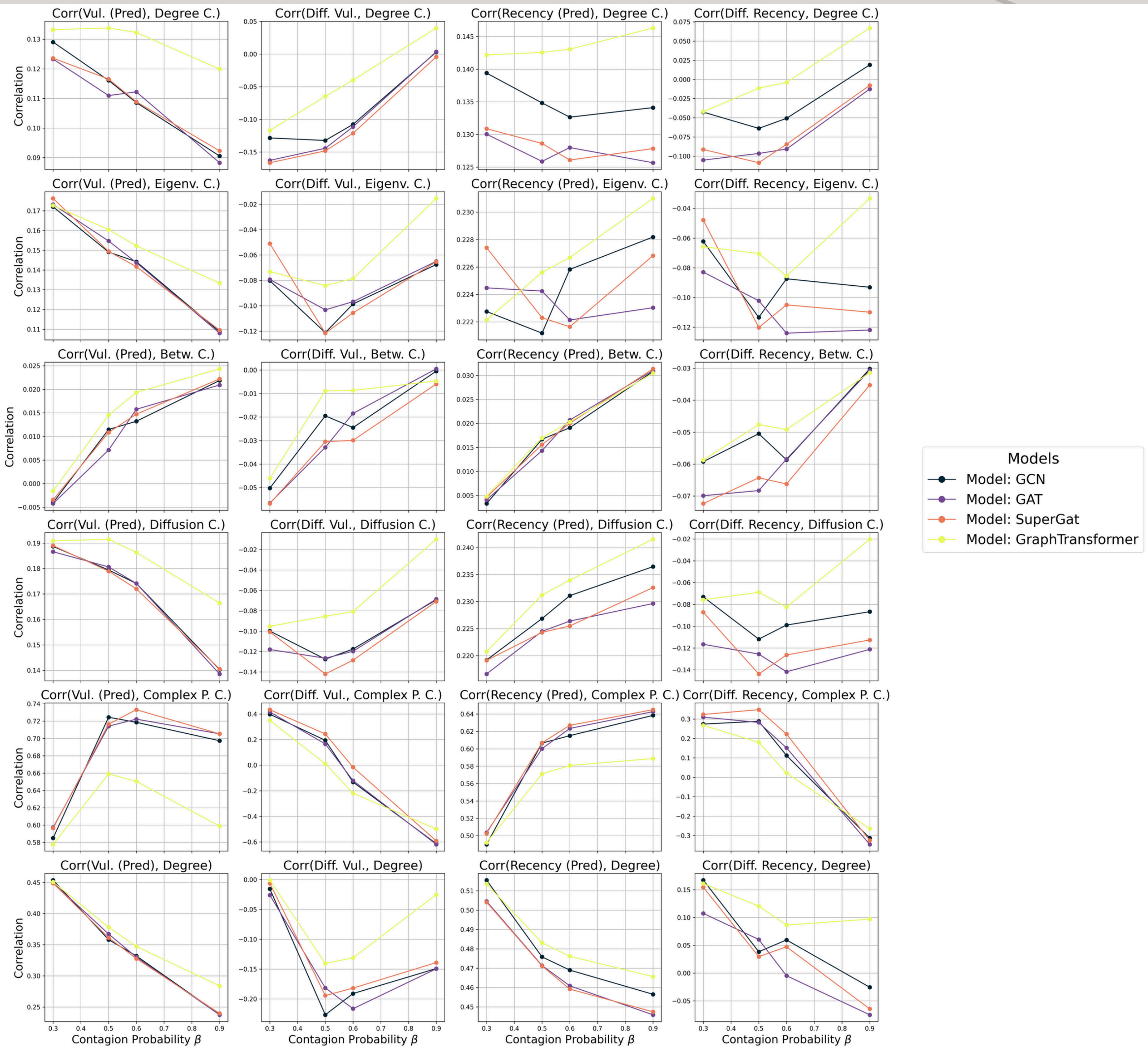


Simple Contagion

Node-Level Correlations

The Graph Transformer model uniquely reduces the correlation between predicted and actual node *Vulnerability* and *Recency*, pushing these correlations closer to zero at lower contagion probabilities.

Its correlation patterns consistently differ from those of other models.

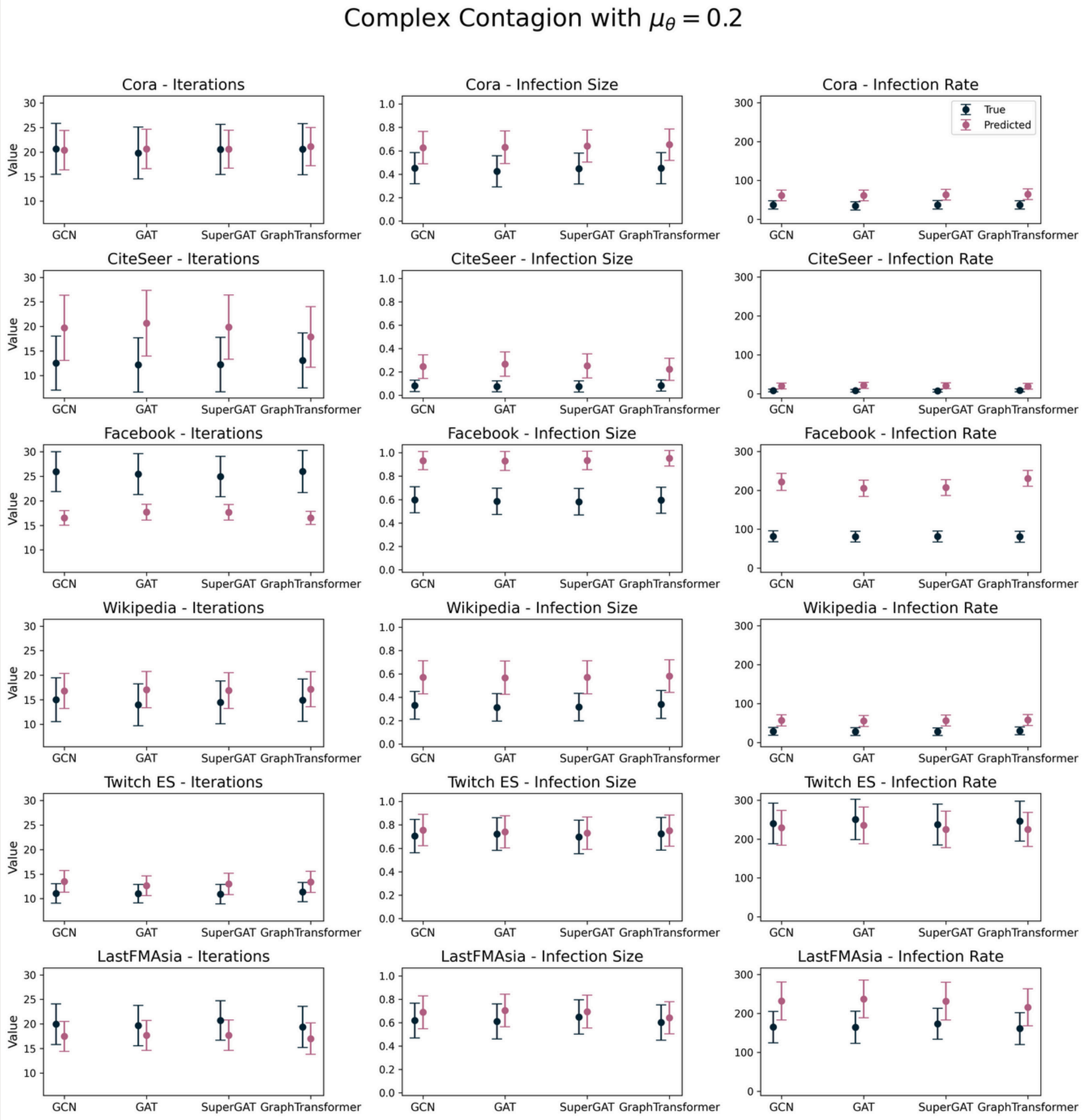


Complex Contagion

Graph-Level

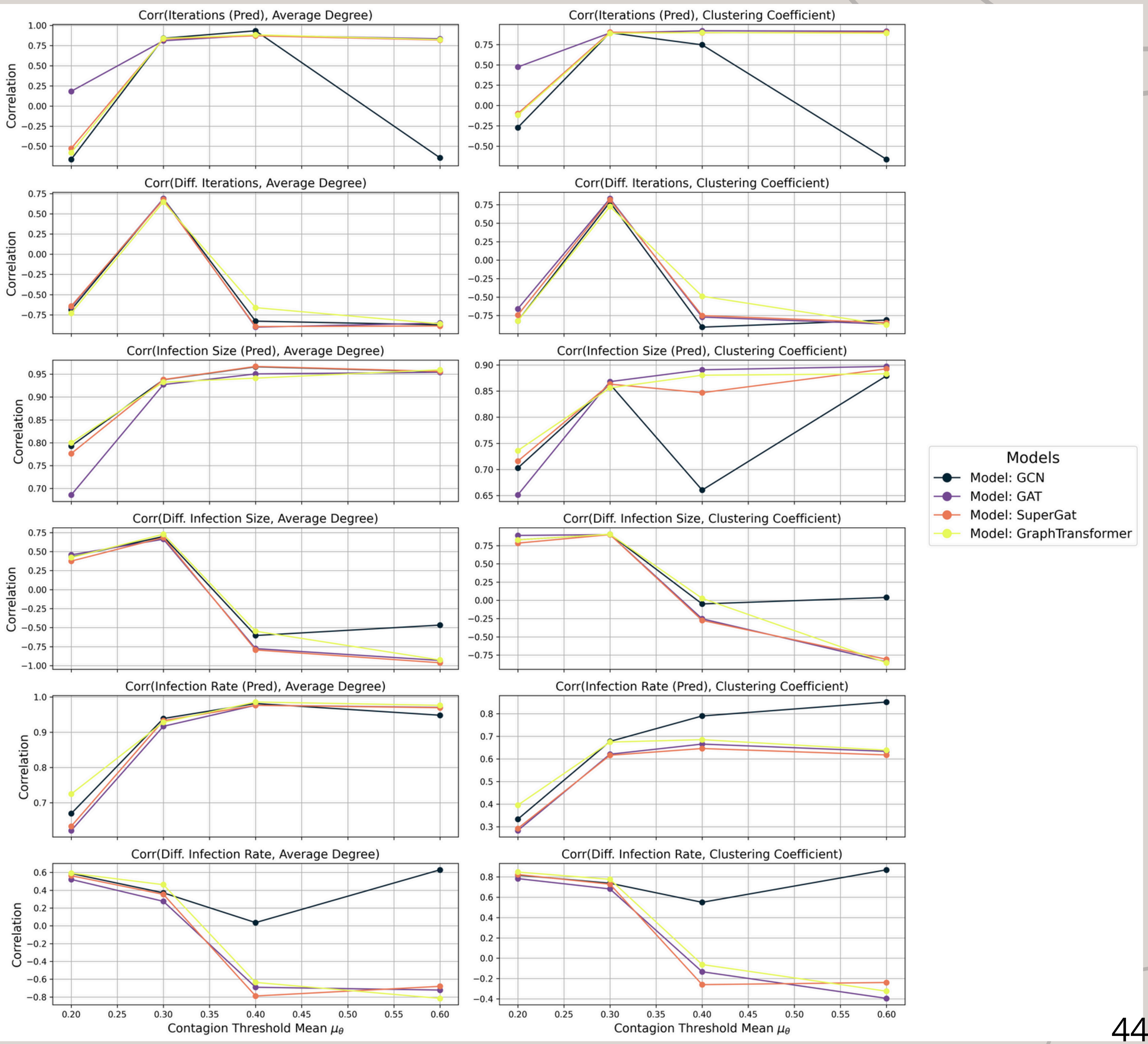
The trends align with those observed in the Simple Contagion model, revealing a consistent pattern across both contagion scenarios:

- **LP algorithms tend to generate network structures that enhance the efficiency of information diffusion.**



Complex Contagion

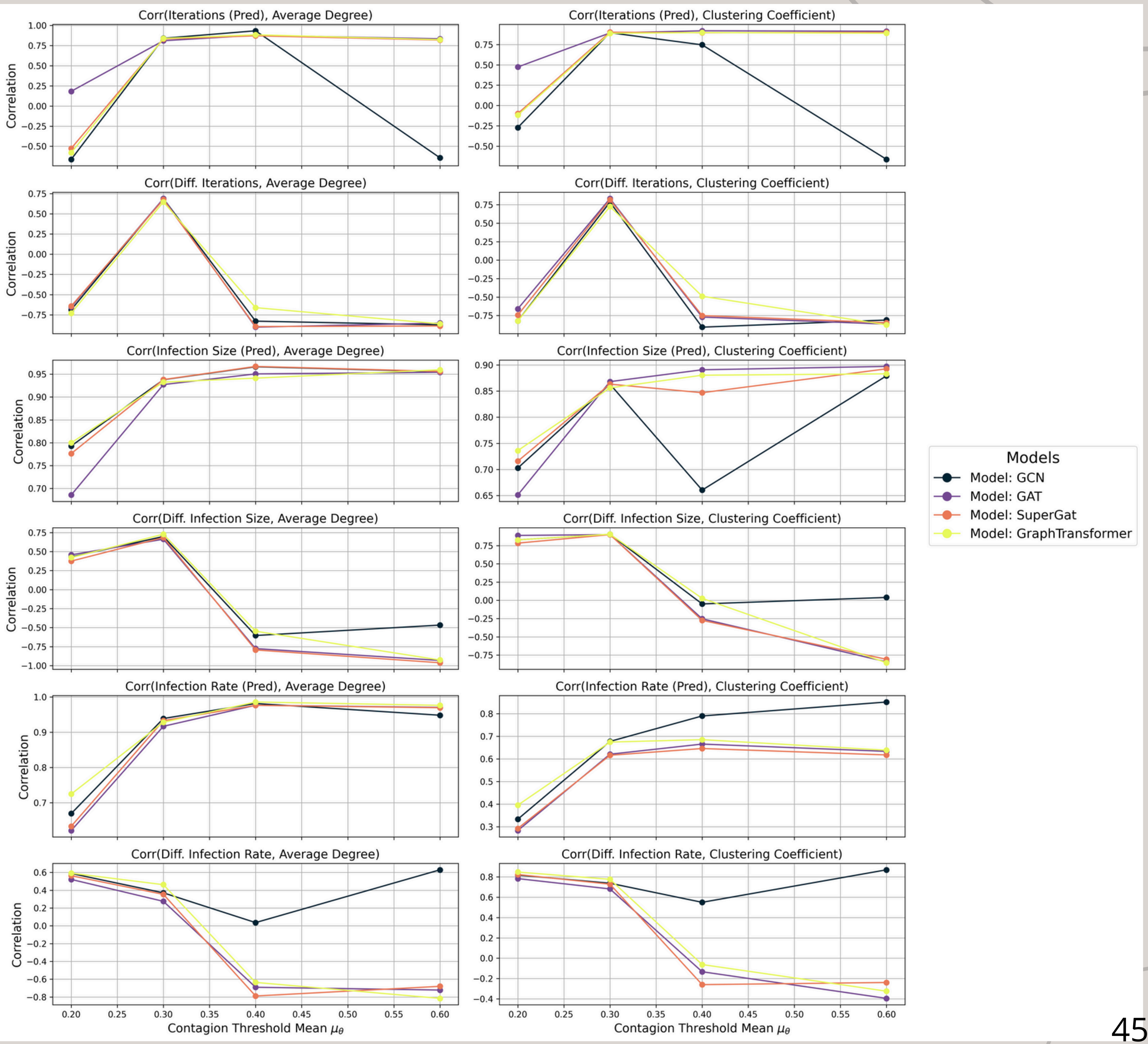
The correlation between contagion metrics and network connectivity features strengthens as the contagion threshold mean (μθ) increases, showing the same phenomenon seen in Simple Contagion.



Complex Contagion

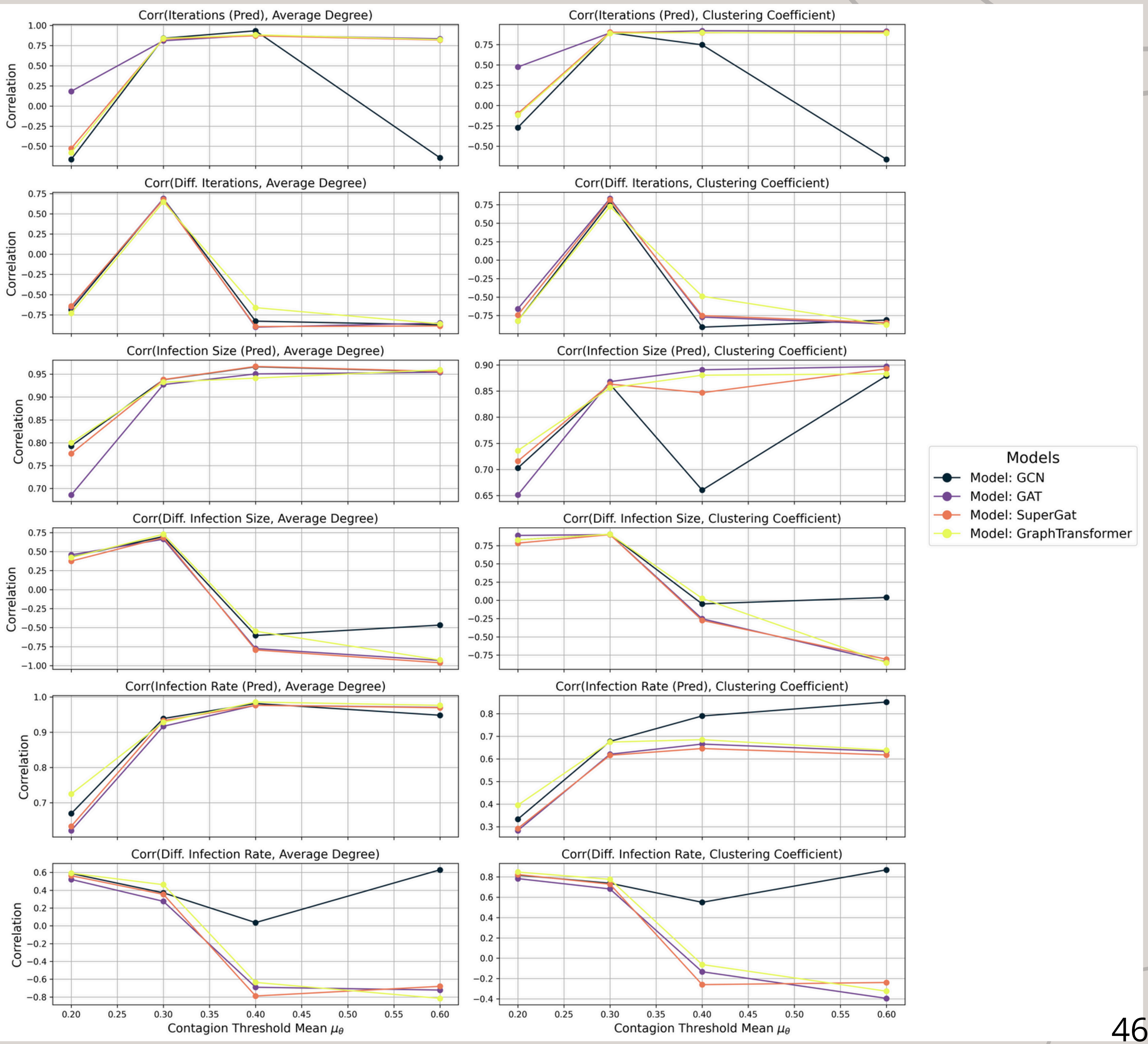
A notable divergence emerges for high thresholds in the GCN model, where its correlations with network connectivity metrics deviate sharply from other models. For example, at high thresholds, GCN shows a strong negative correlation between *Iterations* and connectivity, unlike the positive correlations in other models.

The GCN model's structural changes under high contagion thresholds may hinder diffusion by over-relying on highly connected nodes or forming bottlenecks. Densely connected clusters quickly saturate or isolate uninfected nodes, limiting complex propagation. GCNs local smoothing may emphasize cohesive substructures, reducing iterations needed for stabilization.



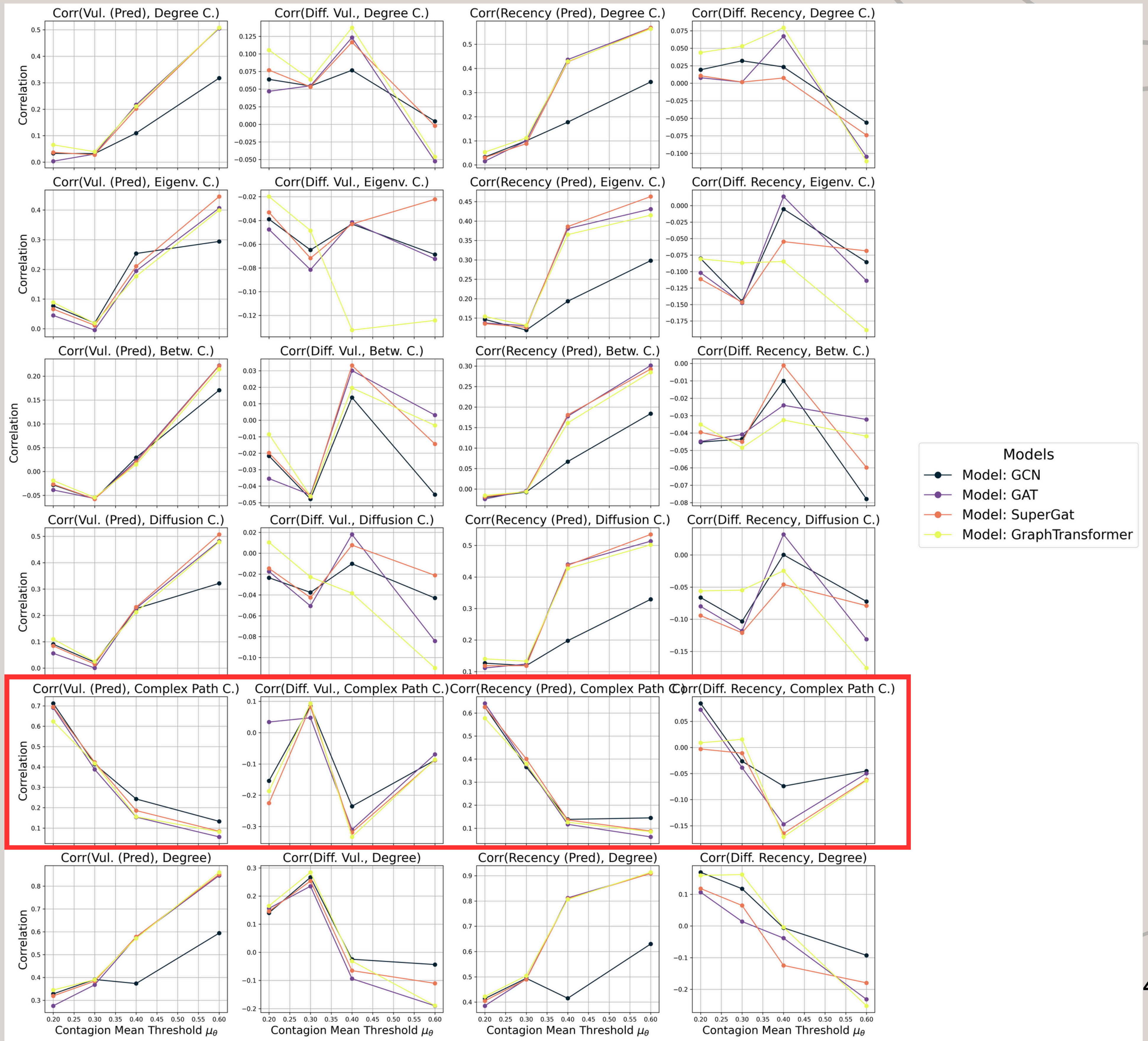
Complex Contagion

In contrast, the positive correlations seen in other models imply that **higher connectivity in their predicted networks extends the stabilization process**. This might happen because these models introduce structures, such as additional bridges between clusters or redundant pathways, which allow the complex contagion to spread more widely before stabilization.



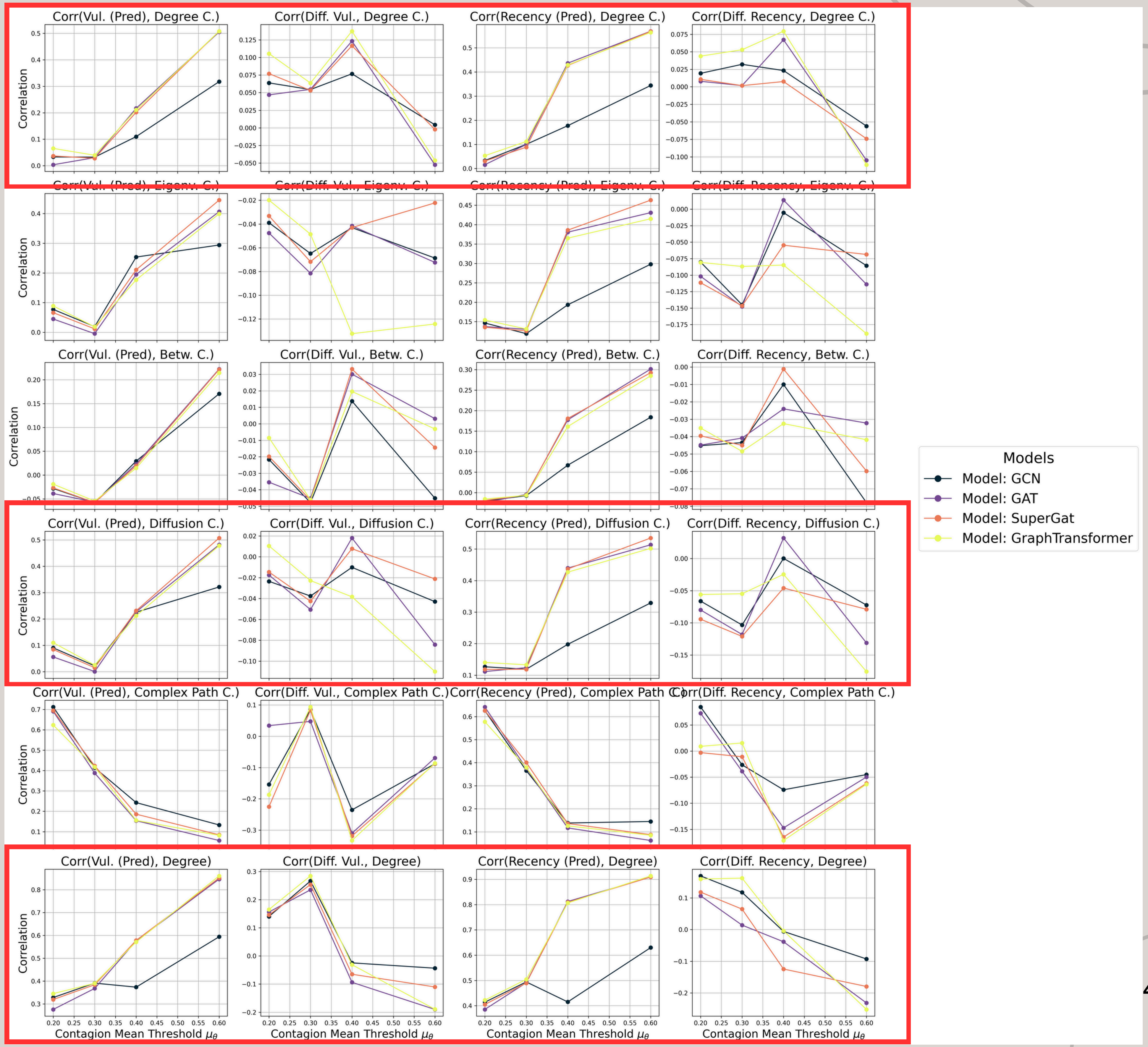
Complex Contagion

At lower contagion thresholds, *Complex Path Centrality* emerge as the most influential factor in determining a node's susceptibility to contagion, reflecting its **ability to capture unique structural features that remain significant even in high-contagion regimes.**



Complex Contagion

At higher contagion thresholds, *Degree, Degree Centrality*, and *Diffusion Centrality* become the dominant predictors.



Conclusions

- Novel Approach to study the LP effects in Social Contagion:
  - By systematically examining the interplay between edge prediction and contagion dynamics, **we addressed a critical gap in existing literature**, bridging advanced machine learning techniques with complex network behavior.
  - We provided a dual perspective (graph-level and node-level analysis) which, combined with our focus on both simple and complex contagion scenarios, offers a **comprehensive understanding of information and behavior spread**.
- Key Methodological Innovations:
  - Employed **state-of-the-art GNN architectures for LP**, including GCN, GAT, SuperGAT and GraphTransformer.
  - Used both classical performance metrics like AUC-ROC and innovative evaluation techniques, such as the Vertex-Centric Max Precision Recall at *k* (VCMPR@*k*).
  - Explored the dependency of correlations and LP effects across four different LP models and six datasets, in both simple and complex contagion scenarios, with varying model depths and contagion probabilities, running **more than 100,000 SI simulations**.

- Contagion Key Findings:
  - Link Prediction models introduce structural shortcuts, reducing average path lengths and enhancing social contagion.
  - Contagion metrics exhibited robustness across models and network depths.
  - Denser networks with high *Average Degree* and *Clustering Coefficient* exhibited larger contagion spreads. More uniform distributions of *Complex Path Centrality* amplified LP effects.
  - Networks with high connectivity driven by a few highly connected nodes, experienced faster contagion spread.
  - The contagion probability modulated infection dynamics, with higher probabilities diminishing the influence of network topology by making nodes uniformly susceptible.

- *Complex Path Centrality* and *Degree* emerged as pivotal in determining node's contagion susceptibility, with *Complex Path Centrality* distinguishing itself as uniquely insightful for capturing the relationship between network topology and contagion behavior.
- In Simple Contagion: Graph Transformers, leveraging global attention, provided smoother, more stable diffusion patterns. GCN, GAT and SuperGAT present similar behaviors.
- In Complex Contagion: While many trends from simple contagion persisted, complex contagion exhibited greater variability between LP models: GCNs, under high contagion thresholds, tended to form localized clusters, either saturating or isolating nodes, in contrast to attention-based models, which facilitated broader propagation via network bridges.

- Future Research Directions:
  - Explore temporal and dynamic networks.
  - Incorporate comparisons with simpler or hybrid LP algorithms.
  - Expand to multiplex networks, where nodes have multiple types of connections.
  - Develop interpretability techniques for GNN-based LP models.
  - Apply findings to real-world scenarios like misinformation mitigation.

References

[1] WeAreSocial, Meltwater. Digital 2024: Italy. [Online]. Available: https://datareportal.com/reports/digital-2024-italy

[2] M. S. Jeffrey Kuiken, Anne Schuth and M. Marx, "Effective headlines of newspaper articles in a digital environment," Digital Journalism, vol. 5, no. 10, pp. 1300–1314, 2017. [Online]. Available: https://doi.org/10.1080/21670811.2017.1279978

[3] News in an Online World: The Need for an " Automatic Crap Detector ", vol. 6?10, 10 2015.

[4] S. González-Bailón, V. D'Andrea, D. Freelon, and M. De Domenico, "The advantage of the right in social media news sharing," PNAS Nexus, vol. 1, 07 2022

[5] T. Graham. (2024) Elon musk's flood of us election tweets may look chaotic – my data reveals an alarming strategy. Accessed: 2024-12-03. [Online]. Available: https : / / theconversation . com elon-musks-flood-of-us-election-tweets-may-look-chaotic-my-data-reveals-an-alarming strategy-243021

[6] L. Terren and R. Borge, "Echo chambers on social media: a systematic review of the literature," 03 2021.

[7] F. Karimi, M. Génois, C. Wagner, P. Singer, and M. Strohmaier, "Homophily influences ranking of minorities in social networks," Scientific Reports, vol. 8, 07 2018.

References

[8] A. Bashardoust, H. C. Beilinson, S. A. Friedler, J. Ma, J. Rousseau, C. E. Scheidegger, B. D. Sullivan, N. Ulzii-Orshikh, and S. Venkatasubramanian, “Information access representations and social capital in networks,” 2023. [Online]. Available: https://arxiv.org/abs/2010.12611

[9] W. L. Hamilton, “Graph representation learning,” Synthesis Lectures on Artificial Intelligence and Machine Learning, vol. 14, no. 3, pp. 1–159.

[10] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” 2017. [Online]. Available: https://arxiv.org/abs/1609.02907

[11] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, “Graph attention networks,” 2018. [Online]. Available: https://arxiv.org/abs/1710.10903

[12] D. Kim and A. Oh, “How to find your friendly neighborhood: Graph attention design with self-supervision,” in International Conference on Learning Representations, 2021. [Online]. Available: https://openreview.net/forum?id=Wi5KUNlqWty

References

[13] Y. Shi, Z. Huang, S. Feng, H. Zhong, W. Wang, and Y. Sun, “Masked label prediction: Unified message passing model for semi-supervised classification,” 2021. [Online]. Available: https://arxiv.org/abs/2009.03509

[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, “Attention is all you need,” in Advances in Neural Information Processing Systems, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/ file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[15] F. Karimi, M. Génois, C. Wagner, P. Singer, and M. Strohmaier, “Homophily influences ranking of minorities in social networks,” Scientific Reports, vol. 8, 07 2018.

[16] R. Aiyappa, X. Wang, M. Kim, O. C. Seckin, J. Yoon, Y.-Y. Ahn, and S. Kojaku, “Implicit degree bias in the link prediction task,” 2024. [Online]. Available: https: //arxiv.org/abs/2405.14985

References

[17] N. Menand and C. Seshadhri, “Link prediction using low-dimensional node embeddings: The measurement problem,” Proceedings of the National Academy of Sciences, vol. 121, no. 8, p. e2312527121, 2024. [Online]. Available: https://www.pnas.org/doi/abs/10.1073/pnas.2312527121

Thanks for your attention!

Antoni Valls Cifre

antoni.vallscifre@gmail.com

Dec2024