University of Padova

Department of Mathematics

Master Thesis in Data Science

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

Supervisor
Alberto Testolin
University of Padova

Co-supervisor

Master Candidate
Antoni Valls Cifre

Academic Year
2023-2024

To Nico and Alícia.

Abstract

Social contagion shapes how information spreads through networks, influencing critical processes ranging from the dissemination of news and job opportunities to the propagation of social movements and potentially misleading content. The structure of social networks plays a pivotal role in determining the reach and impact of these diffusion processes, with network topology critically modulating how individuals access and interpret information.

Within this context, link prediction (LP) emerges as a crucial mechanism for understanding and potentially manipulating network structures. The primary goal of LP is to determine whether two nodes in a network are likely to form a connection, thereby potentially reshaping the network's information transmission capabilities. While numerous LP methods have been proposed in the literature, along with various methodologies, biases research, and evaluation approaches, the relationship between LP and social diffusion processes remains less thoroughly explored, particularly concerning Graph Neural Network (GNN)-based LP algorithms.

In this study, we systematically analyze four distinct GNN-based LP models to investigate how predicted network structures influence social contagion dynamics. Our research employs six diverse datasets, characterized by comprehensive node-level centrality measures and graph-level topological metrics. By leveraging this methodological variability, we aim to provide a nuanced understanding of how network characteristics correlate with social diffusion and how they are modulated by different LP models. We model social contagion using both simple and complex contagion frameworks through epidemic modeling techniques.

Our findings reveal that LP models consistently reshape network structures in ways that significantly influence contagion dynamics. By introducing structural shortcuts or targeting hub nodes, these models enhance information diffusion, particularly in denser networks with high average degrees and clustering coefficients. Additionally, we observe that the impact of LP varies between simple and complex contagion processes, with attention-based models like Graph Transformers facilitating broader propagation and Graph Convolutional Networks (GCNs) forming localized clusters under specific conditions. Critically, measures such as Complex Path Centrality and node degree emerge as key predictors of contagion susceptibility, highlighting the intricate interplay between network topology and social diffusion behavior.

This work contributes a comprehensive overview of GNN-based LP methods, network and node characterization, and social contagion modeling, taking an initial step toward bridging existing literature gaps and advancing the understanding of LP's impact on social dynamics.

Contents

Listing of figures

Listing of tables

Listing of acronyms

AUC-ROC . . . . . Area Under the Receiver Operating Characteristic Curve

CV . . . . . Computer Vision

CNN . . . . . Convolutional Neural Networks

DP . . . . . Dot-Product Attention

FN . . . . . False Negative

FP . . . . . False Positive

FPR . . . . . False Positive Rate

GCN . . . . . Graph Convolutional Network

GN . . . . . Graph Network

GAT . . . . . Graph Attention Network

GNN . . . . . Graph Neural Network

GO . . . . . Original GAT Attention

LLE . . . . . Locally Linear Embedding

LP . . . . . Link Prediction

MLP . . . . . Multi-Layer Perceptron

MPNN . . . . . Message-Passing Neural Network

MX . . . . . Mixed GO and DP Attention

NLNN . . . . . Neural Logic Network

NLP . . . . . Neural Language Processing

PyG . . . . . `PyTorch Geometric` library

ReLU . . . . . Rectified Linear Unit

RNN . . . . . Recurrent Neural Network

SI . . . . . Susceptible-Infected

SuperGAT . . . . . Supervised Graph Attention Network

TN . . . . . True Negative

TP . . . . . True Positive

TPR . . . . . True Positive Rate

VCMPR@k . . . . Vertex-Centric Max Precision Recall at k

1 Introduction

Social media has become an integral part of daily life worldwide, influencing how people connect, communicate, and consume information. In Italy alone, there were 42.80 million social media users as of January 2024, representing 72.8% of the population [3]. Italian users spend an average of over 23 hours per month on platforms such as Facebook, X (formerly Twitter), and LinkedIn [4]. This extensive adoption underscores the central role that social media now plays in modern society, not only as a tool for interaction but also as a dominant channel for accessing and sharing information.

In today's digital age, people's personalities, interests, and even political views are significantly shaped by the content they encounter on social media. Traditional news sources are increasingly being supplanted by social media networks as primary channels for current events and breaking news. This fundamental shift has radically restructured the relationship between journalism, media organizations, and their audiences, creating a dynamic and often problematic information ecosystem [5].

Media outlets have rapidly adapted to this new landscape, developing strategies that prioritize algorithmic engagement over traditional journalistic principles. The rise of clickbait techniques has become a defining characteristic of modern digital journalism, where headlines and content are meticulously crafted to maximize algorithmic visibility and user interaction. Journalists and media organizations now design content with social media algorithms in mind, employing strategies like sensationalist headlines, emotionally charged language, and provocative framing to increase shares, comments, and follows [5, 6].

These algorithmic adaptations have profound implications for information dissemination. News organizations increasingly prioritize content that generates immediate emotional responses and viral potential over depth, nuance, and factual comprehensiveness. The metrics of success have shifted from journalistic integrity to engagement rates, likes, and algorithmic recommendation.

Similarly, politicians have become particularly adept at manipulating these algorithmic dynamics. By crafting provocative, emotionally charged statements designed to trigger immediate reactions, they can instantly dominate media cycles and amplify their messaging. Social media platforms enable political figures to bypass traditional media gatekeepers, directly communicating with followers through carefully curated narratives. This direct communication allows for rapid response to global events, but also creates an environment where inflammatory rhetoric and polarizing statements can quickly gain unprecedented visibility and spread [7].

This phenomenon is not uniform across ideological lines. Research shows that right-leaning content enjoys structural and algorithmic advantages on social media, amplifying its visibility and engagement relative to left-leaning content. This "amplification of the right" stems from factors like greater audience susceptibility to moralized content and algorithmic curation favoring partisan narratives. For instance, during movements like Black Lives Matter, even in progressive networks, right-leaning outlets dominated the discourse, illustrating how social media often favors polarizing and extreme content [8].

The 2024 U.S. presidential elections provide a contemporary example of this phenomenon. Donald Trump, in tandem with Elon Musk's amplified presence on X, leveraged the platform to dominate public attention. Trump's campaign used hashtags like #TooBigToRig and #StopTheSteal to frame narratives around election integrity, while Musk's algorithmic influence and live discussions helped spread conspiracy theories and reinforce echo chambers. Together, their strategy overwhelmed the information space, fostering polarization and shaping public discourse in ways that mirrored the algorithmic incentives of social media platforms [9].

This algorithmic logic incentivize extreme, divisive, and simplified narratives over balanced, complex reporting. As a result, the boundary between news, opinion, and pure entertainment becomes increasingly blurred, challenging traditional notions of journalistic objectivity and public information consumption. Media organizations, politicians, and individual users are now participants in a complex ecosystem where algorithmic logic increasingly mediates social understanding and public discourse.

Instead of receiving an unbiased and comprehensive view of the world, users encounter information largely filtered through the perspectives of individuals, organizations, and influencers

within their networks. This networked structure influences not only what they see but also how they interpret and form opinions. The position individuals hold within these networks — and the connections they forge — directly shapes their information exposure to information and, ultimately, their social capital [10, 11].

The impact of social media structures is in the spotlight today. The design of major platforms and the algorithms that power their content have far-reaching effects, shaping public opinion and even impacting national election outcomes [12, 13]. These systems determine the information users encounter, often amplifying certain voices while suppressing others, fostering echo chambers [14] and marginalizing minority perspectives [15]. As a result, social media's role in shaping political views and behaviors has emerged as a central concern in both public discourse and academic research.

Social networks continuously evolve, shaped by both organic growth and algorithmic interventions. At the heart of many online platforms lies link prediction (LP), also known as link recommendation, a crucial technology that reshapes these networks by recommending connections through features like "People You May Know" on LinkedIn or "Suggested Friends" on Facebook. Operating as a predictive task, LP determines whether a connection exists or will form between network nodes by analyzing various factors, including neighbor similarities, network topology, and node features [16, 17, 18]. While its applications extend beyond social platforms — from uncovering protein interactions in biological networks [19] to predicting citations in academic networks [20] — LP has become foundational for modern recommender systems and network analysis.

Previous research has extensively studied network structures and their effect on social contagion — the complex process through which ideas, behaviors, or information propagate across networks [21, 22, 23]. Drawing parallels from epidemiology, social contagion is often modeled using epidemic spreading frameworks, where information or behaviors spread like diseases through a population [24]. These models typically distinguish between two fundamental types of contagion processes: simple contagion, where spread occurs through direct contact with a single source (like sharing news, viral memes, or basic information), and complex contagion, where adoption requires social reinforcement from multiple sources (like adopting new behaviors, beliefs, or technologies, where individuals need substantial social proof or validation before changing their state). These processes fundamentally shape how information flows, opinions form, and behaviors spread across social networks [22, 25].

The intersection of LP algorithms and social contagion presents a fascinating yet underexplored research frontier. When LP algorithms modify network structures by suggesting new

connections, they may inadvertently create or remove pathways for information flow, potentially altering the dynamics of both simple and complex contagion processes. For instance, by connecting previously distant communities, LP algorithms might accelerate the spread of information across diverse groups. Conversely, by strengthening existing clusters, they could intensify echo chambers and contribute to opinion polarization. At both the local (node) and global (graph) levels, these algorithmic interventions might fundamentally reshape network vulnerability to misinformation, alter the speed of information diffusion, and influence the formation of community structures.

This study aims to take the first steps in addressing this gap by exploring how different LP algorithms influence social contagion processes across different network structures. While traditional LP methods rely on heuristic approaches like common neighbors or path-based metrics, we focus exclusively on Graph Neural Network-based LP algorithms, which have emerged as the state-of-the-art approach in recent years [26, 27, 28, 29].

Graph Neural Networks (GNNs) are a class of deep learning models specifically designed to operate on graph-structured data [30]. GNNs have demonstrated superior performance by leveraging their ability to simultaneously process both graph topology and rich node/edge feature information through their message-passing architecture and learned representations This enables them to effectively model relationships and dependencies, making them highly suitable for tasks like LP. Their flexibility in incorporating node and edge features further enhances their capability to understand complex network dynamics. At the core of these GNN-based LP methods is the use of node embeddings — low-dimensional vector representations of nodes that capture information about their local and global neighborhood structures—. Node embeddings are generated through dimensionality reduction techniques, condensing high-dimensional graph data into compact, informative vectors that can then be leveraged for tasks such as node classification, clustering, and LP [2, 31, 32].

This research aims to take an initial step toward bridging gaps in the existing literature and advancing the understanding of LP and its impact on social dynamics. While the focus remains on static LP and link reconstruction — rather than dynamic network evolution — this study will explore the structural changes induced by LP models at both graph and node levels. By examining these structural impacts, the research seeks to uncover how LP algorithms shape the underlying network in ways that influence social contagion and information diffusion processes.

To this end, we will employ a variety of state-of-the-art GNN architectures for LP tasks, including Graph Convolutional Networks (GCNs) [33], Graph Attention Networks (GATs)

[34], SuperGAT [35], and GraphTransformers [36]. These models offer diverse approaches to capturing graph topology and node feature interactions. GCNs leverage spectral graph theory to aggregate neighborhood information, while GATs introduce attention mechanisms to weight the importance of different neighbors. SuperGAT builds on GAT by incorporating self-supervised learning signals for enhanced edge-level attention, and GraphTransformers extend the transformer paradigm to graphs, enabling the modeling of long-range dependencies and more complex structural relationships. Together, these models provide a comprehensive toolkit for investigating how different LP mechanisms influence the underlying network structure.

This investigation will delve into the effects of LP algorithms at two complementary scales:

Graph-Level Analysis:

- Comprehensive exploration of network topologies using advanced metrics: average degree, clustering coefficient, and Gini coefficient for centrality measures.
- Rigorous performance evaluation using AUC-ROC scores across different LP models.
- In-depth analysis of LP-induced structural changes and their impact on social contagion processes.
- Comparative assessment of diffusion dynamics, including contagion stabilization time, total infection size, and infection propagation rate.
- Comparative analysis between predicted and real network contagion patterns.

Node-Level Analysis:

- Comprehensive node's structural assessment using centrality measures such as degree centrality, eigenvector centrality, diffusion centrality, and complex path centrality.
- Evaluation of LP performance using VCMPR@k scores, a local measure used in LP tasks involving node embeddings in graphs [37].
- Detailed examination of nodes' roles in contagion dynamics, including node's vulnerability and recency.
- Comparative analysis between node's predicted and real neighbours contagion metrics.

By integrating these node-level and graph-level analyses, this research provides a comprehensive framework for understanding the dual-scale impact of LP algorithms on network structures and their subsequent influence on contagion processes.

The thesis is organized as follows. Chapter 2 presents a comprehensive review of the related literature on the main topics of this thesis. Chapter 3 introduces GNNs, their mathematical formalisms, and the specific GNN models that will be employed for our LP task. Chapter 4 details the LP task, its common frameworks, and evaluation metrics. Chapter 5 explores social contagion dynamics and their prevalent modeling approaches. The research results are presented in Chapter 7, with concluding remarks provided in Chapter 8.

2
Related Work

Earlier groundwork on processing structured data using neural networks was laid by Sperduti et al. [38], which introduced the generalized recursive neuron. This concept extended neural networks to classify structured patterns, like trees and logic terms, overcoming the limitations of feature-based methods. These foundational advances in learning algorithms for structured data gradually paved the way for graph neural networks (GNNs), an emerging field that would soon attract significant research attention.

As GNNs emerged, researchers began providing comprehensive reviews and frameworks to systematize this evolving domain. Bronstein et al. [39] introduced the term geometric deep learning and provided a foundational overview of deep learning techniques applied to non-Euclidean domains, such as graphs and manifolds. While this pioneering work represents the first comprehensive survey on GNNs, its primary focus is on convolutional GNNs. Similarly, Hamilton et al. [2] synthesized graph representation learning techniques, introducing GNN formalism and emphasizing solutions to network embedding problems, although with coverage limited to a subset of GNN models. Battaglia et al. [40] introduced the graph networks (GN) framework, which defines a class of functions specifically designed for relational reasoning over graph-structured data. This framework generalizes and extends several established approaches, including GNNs, message-passing neural networks (MPNNs), and neural logic networks (NLNNs) [41, 42, 43]. By providing a unified perspective, the GN framework enables the construction of complex architectures from simple and modular building blocks, offering a robust foundation for relational reasoning in graph-based representations. Lee et al. [44] pro-

vided a focused review on GNNs employing attention mechanisms, presenting a partial survey of this specialized subset of models. In contrast, Wu et al. [1] offered a comprehensive survey of GNNs, leveraging abundant resources to analyze existing limitations and proposing a new taxonomy.

Beyond general overviews, significant research also addresses the limitations and challenges inherent in GNNs. Alon et al. [45] identified the bottleneck problem, describing how GNNs struggle to propagate information across long paths in a graph, which adversely affects message aggregation between distant nodes. Shchur et al. [46] highlighted issues in experimental setups, particularly the common practice of relying on a single train/validation/test split. Their findings demonstrated that a simple Graph Convolutional Network (GCN) can outperform more complex GNN architectures when consistent hyperparameter tuning and training procedures are applied, and results are averaged across multiple data splits. Finally, Dong et al. [47] investigated how GNNs may produce biased outcomes against certain demographic subgroups. Their work underscores the importance to develop effective and fair GNN models that mitigate the influence of biased structures in the input network, ensuring these do not serve as a significant source of bias.

The issue of bias in LP models has also garnered considerable attention in recent research, particularly regarding fairness in predicting connections within social networks. Studies have shown that LP algorithms often exhibit biases, such as favoring intra-group connections or amplifying the visibility of certain groups over others. For example, accuracy disparity in LP models, as discussed by Li et al. [48], reveals that inter-group links are often predicted with less accuracy than intra-group links. This research identifies imbalanced link densities as a root cause, proposing a mitigation method (FAIR-LP) to equalize link distributions while preserving network structure.

The homophily principle [49] also plays a crucial role in LP bias, as highlighted by Masrour et al.[50], who show that LP algorithms often reinforce filter bubbles by favoring connections between similar nodes. Karimi et al. [15] further demonstrate that this effect increases network segregation, reducing exposure to diverse perspectives and potentially disadvantaging minority groups by limiting their opportunities to connect with majority groups or access new information. Approaches to mitigate this issue include introducing fairness criteria that enhance link diversity, promoting more heterogeneous network structures.

Further, research on degree bias shows that LP models often favor high-degree nodes, which skews results and can amplify "rich-get-richer" effects. As noted by Aiyappa et al.[51], this bias can lead algorithms to overemphasize node degree, resulting in deceptively high performance

scores. They argue that the standard LP benchmark has a substantial inherent bias that disproportionately rewards models exploiting node degree information. This degree bias stems from edge sampling in performance evaluations: when edges are randomly sampled from a graph, a node with k edges is k times more likely to be chosen than a node with only one edge ($k = 1$). In contrast, negative edge samples are taken from randomly selected unconnected node pairs, lacking this degree-based skew. To address this, they propose a degree-corrected task for a more balanced and fair link prediction evaluation.

Meanwhile, Subramonian et al. [52] dives into within-group fairness and reveals that some LP methods, especially GCNs, have a built-in bias toward high-degree nodes. Their work introduces new fairness metrics to assess these effects within groups, aiming to mitigate degree-based disparities.

While extensive research has investigated the bias effects of LP models and the influence of link recommendation algorithms on dynamics such as polarization [53] and network centralities [54], the relationship between LP and diffusion processes remains less understood. Existing literature has primarily focused on accurately estimating classification performance for missing links [16, 55], with limited attention to the impact of LP on a network's spreading capacity. An initial study by Weng et al. [56] provided evidence from a meme dataset that information diffusion plays a significant role in shaping network evolution. Similarly, Li et al. [57], using data from a micro-blogging platform, demonstrated that information diffusion influences the creation of new links. Their analysis concluded that incorporating diffusion processes as a feature in link recommendation algorithms yields better results than relying solely on topological properties. However, neither study evaluated specific LP strategies nor proposed a general framework for assessing or characterizing the networks that evolve under these dynamics.

In an early study, Vega et al. [58] examined how traditional LP algorithms — such as SimRank [59], PageRank [60], Common Neighbors [16] , and Adamic-Adar [61] — affect various diffusion processes, including epidemic, information, and rumor spread, in dynamic networks. They calculated each network's initial spreading capacity for different diffusion models and generated evolved versions by adding edges recommended by each LP method, also keeping an eye on the structural properties of the LP-evolved networks. Their findings suggest that adding new edges does not always enhance spreading capacity; LP methods that maintain or lower network complexity tend to achieve better spreading results. Changes in structural properties like the number of triangles, modularity, and assortativity did not consistently correlate with improved diffusion. However, Vega et al.'s work is limited to traditional heuristics and

does not investigate the effects of GNN-based LP models, leaving a gap in understanding how more advanced models might influence diffusion dynamics.

Research on social contagion has grown significantly in recent years, driven by the increasing reliance on social media and our interconnected roles within these artificial social networks. Mønsted et al. [62] provided experimental evidence that the complex contagion model better describes observed information diffusion behavior on Twitter compared to simple contagion. Their study employed 'social bots' to orchestrate coordinated attempts at spreading information. Sassine et al. [63] explored how network structure influences the speed and reach of social contagions, highlighting the pivotal role of topology in shaping diffusion processes. Banerjee et al. [23] investigated how the network positions of initial information recipients affect the diffusion of new products. To study this, a microfinance institution entered 43 villages in India, offering microfinance loans and collecting detailed network data through household surveys. Their findings introduced diffusion centrality, a novel measure of a node's effectiveness as an injection point for diffusion processes. Jackson et al. [11] further advanced the field by providing a typology of social capital, breaking it into seven distinct forms, each tied to different node characteristics and centrality measures. Guilbeault et al. [64] expanded on this by introducing measures such as complex path length and complex-path centrality, improving the identification of key individuals and network structures critical for complex contagion.

Despite these advancements, significant gaps remain in understanding how LP algorithms interact with social contagion processes. While prior studies have laid a strong foundation for understanding the interplay between network structure, centrality, and contagion, the influence of GNN-based LP on the spread of information is unexplored.

3

Graph Neural Networks

Researchers have developed neural networks that operate on graph-structured data for over a decade [1]. Unlike traditional neural networks that process structured data such as images — Convolutional Neural Networks (CNNs) [65] — or sequences — Recurrent Neural Networks (RNNs) [66] —, Graph Neural Networks (GNNs) [30] work directly with graphs, defined as $\mathcal{G}(\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = u_1, u_2, \ldots, u_n$ represents the set of vertices or nodes, and $\mathcal{E}$ denotes the set of edges connecting these nodes. GNNs process a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ along with node features $\mathbf{X} \in \mathcal{R}^{d \times |\mathcal{V}|}$ to generate node embeddings $\mathbf{z}_u$.

Early methods for node embedding involve projecting nodes onto a lower-dimensional space by estimating the graph's intrinsic dimensionality using spectral techniques on the adjacency matrix. Examples include approaches like Locally Linear Embedding (LLE) [67] and Isomap [68]. However, a key limitation of these early techniques was their inability to scale well to large networks. Later, more robust statistical models emerged, where node parameters are derived by optimizing a global objective function to preserve graph structure. The underlying principle is that similar nodes in the original graph should remain close in the lower-dimensional space. This notion of graph proximity is often defined by neighboring nodes, reflecting the homophily principle [49], where connected entities tend to share characteristics.

Random walk-based methods [69, 31] leverage this concept of neighboring connections, as information and labels propagate along the network, leading connected nodes to be positioned closely in the latent space. Going a step further, GNNs have proven particularly powerful in this context, as they learn node embeddings dynamically during training. This allows GNNs

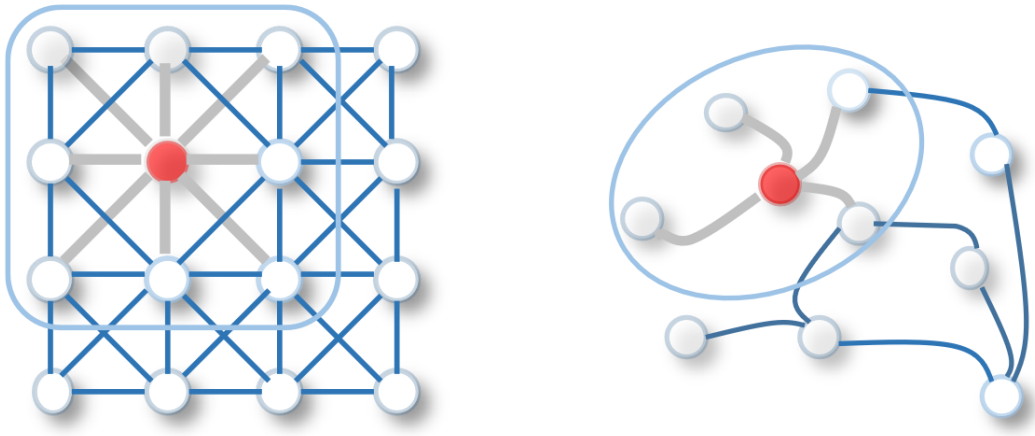

Figure 3.1: (*left*) 2D Convolution: Images can be represented as a regular grid in the Euclidean space. Analogous to a graph, each pixel is taken as a node where neighbors are determined by the filter size. The 2D convolution takes the weighted average of pixel values of the red node along with its neighbors, which are ordered and have a fixed size.
(*right*) Graph Convolution: To get a hidden representation of the red node, a simple graph convolutional operation involves taking the average value of the node features of the red node along with its neighbors. Unlike image data, the neighbors of a node in a graph are unordered and variable in size.
Source: Adapted from Wu et al. [1].

to capture complex, non-linear relationships inherent in graph data.

Interestingly, images can also be interpreted as graphs, where pixels are treated as nodes and edges connect neighboring pixels based on spatial proximity. In this framework, convolutional operations, fundamental to CNNs, can be generalized to graphs. In standard CNNs, the convolution operation aggregates features from a pixel's local neighborhood (e.g., a 3×3 kernel) using predefined spatial relationships. Similarly, in GNNs, convolution is redefined as a message-passing mechanism, where features of a node are updated by aggregating information from its neighbors according to the graph structure [2]. As shown in Figure 3.1, an image can be viewed as a specific instance of a graph, where pixels are connected by adjacent pixels. . Analogous to 2D convolution, graph convolutions can be performed by taking the weighted average of a node's neighborhood information.

The cornerstone of GNN architecture is neural message passing, where nodes exchange and update vector messages through neural networks. The intuition behind message passing is that a node's characteristics (or features) are influenced by its neighboring nodes in a graph. Therefore, during training, a GNN learns how to optimally aggregate and propagate information from each node's neighbors, updating the node's embeddings to capture the underlying structure of the graph. This generalization allows GNNs to extend convolutional principles to irreg-

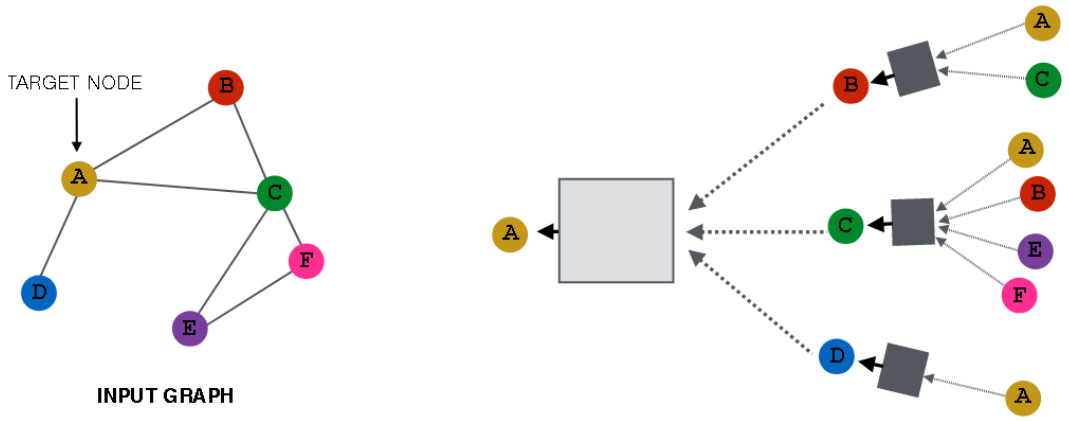

Figure 3.2: Illustration of a single node aggregating messages from its local neighborhood. Each neighboring node contributes information that has been aggregated from its own respective neighborhood, creating a recursive flow of information. This visualization demonstrates a two-layer structure of the message-passing model.
Source: Adapted from Hamilton et al. [2].

ular, non-Euclidean domains, enabling their application to a wide range of problems beyond grid-like data.

3.1 Message Passing Process

The GNNs message-passing framework follows an intuitive progression: 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. Specifically, after the first iteration ($k = 1$), every node embedding contains information from its 1-hop neighborhood, *i.e.*, every node embedding contains information about the features of its immediate graph neighbors; after the second iteration ($k = 2$) every node embedding contains information from its 2-hop neighborhood (see Figure 3.2); and in general, after k iterations every node embedding contains information about its k-hop neighborhood [2]. Figure 3.3 illustrates a basic GNN architecture.

The message-passing process can be broken down into several stages that occur across multiple layers of the GNN:

1. Initialization: Each node starts with its initial feature representation $X_i \in \mathcal{R}^d$, derived from domain-specific attributes (such as user profiles in social networks or atomic properties in molecular graphs).

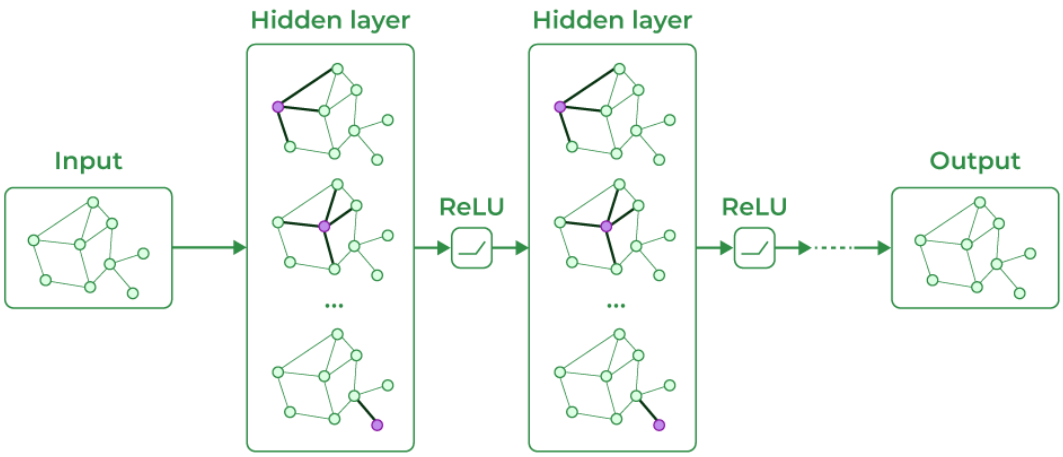

Figure 3.3: Overview of a basic GNN architecture. The input graph structure and features are processed through multiple hidden layers, where message-passing occurs. Each hidden layer aggregates information from neighboring nodes, applies a ReLU activation function, and propagates updated node representations to subsequent layers, culminating in the output graph representation.

2. Aggregation: At each layer of the network, a node collects (aggregates) information from its neighboring nodes. This aggregation step combines the features of the neighboring nodes, allowing the target node to update its representation based on its local neighborhood.

3. Update: After aggregation, a transformation step — typically a learnable function, like a linear layer followed by a non-linear activation function — is applied to the aggregated information. This transforms the node's features into a new representation that incorporates the neighborhood's influence. As layers stack, nodes progressively gather information from increasingly larger neighborhoods, capturing more of the global structure of the graph.

This process can be formally expressed as:

$$\mathbf{h}_u^{(k)} = \text{UPDATE}^{(k-1)}\left(\mathbf{h}_u^{(k-1)}, \text{AGGREGATE}^{(k-1)}\left(\left\{\mathbf{h}_v^{(k-1)}, \forall v \in \mathcal{N}(u)\right\}\right)\right) \tag{3.1}$$

$$= \text{UPDATE}^{(k-1)}\left(\mathbf{h}_u^{(k-1)}, \mathbf{m}_{\mathcal{N}(u)}^{(k-1)}\right), \tag{3.2}$$

where $\mathbf{h}_u^{(k)}$ represents the embedding of node $u \in \mathcal{V}$ at iteration k, $\mathcal{N}(u)$ denotes u's neighborhood, $\mathbf{m}_{\mathcal{N}(u)}$ represents the aggregated message from $\mathcal{N}(u)$, and UPDATE and AGGREGATE are neural networks [2].

After running K iterations of message passing — K layers or K Graph Neural blocks [40] —, the final node embeddings are defined as:

$$\mathbf{z}_u = \mathbf{h}_u^{(K)}, \forall u \in \mathcal{V}\,. \tag{3.3}$$

Notably, since the AGGREGATE function operates on sets, these GNNs are inherently permutation equivariant, which is essencial as there is no natural ordering of node's neighbours [2].

Through this iterative process, GNNs capture both local interactions and global patterns, enabling nodes to develop rich, contextual representations that reflect their position and relationships within the graph structure

3.2 Learning in GNNs

GNNs address two fundamental learning challenges:

1. Aggregation Learning: The GNN needs to learn how to effectively aggregate information from neighboring nodes. A good aggregation method should preserve useful information from the neighbors while filtering out noise, enabling each node to update its representation in a meaningful way.

2. Representation Learning: The model must learn node embeddings that serve downstream tasks effectively. For example, in node classification, the learned embeddings should enable the model to distinguish between different node classes. In link prediction tasks the embeddings should capture the likelihood of edges (relationships) forming between pairs of nodes.

The power of GNNs lies in their ability to learn representations that encode both node-specific features and neighborhood structural properties, making them powerful tools for diverse graph-based applications.

3.3 The Basic GNN

The standard GNN message-passing mechanism is defined as [70, 30]:

$$\mathbf{h}_u^{(k)} = \sigma\left(\mathbf{W}_{\text{self}}^{(k)}\mathbf{h}_u^{(k-1)} + \mathbf{W}_{\text{neigh}}^{(k)}\sum_{v\in\mathcal{N}(u)}\mathbf{h}_v^{(k-1)} + \mathbf{b}^{(k)}\right), \tag{3.4}$$

where $\mathbf{W}_{\text{self}}^{(k)}, \mathbf{W}_{\text{neigh}}^{(k)} \in \mathbb{R}^{d^{(k)} \times d^{(k-1)}}$ are trainable parameter matrices, σ denotes an element wise non-linearity, such as a $\tanh$ or ReLU function, and $\mathbf{b}^{(k)} \in \mathbb{R}^{d^{(k)}}$ is the bias term. The graph-level equation is written as follows:

$$\mathbf{H}^{(k)} = \sigma\left(\mathbf{H}^{(k-1)}\mathbf{W}_{\text{self}}^{(k)} + \mathbf{A}\mathbf{H}^{(k-1)}\mathbf{W}_{\text{neigh}}^{(k)}\right), \tag{3.5}$$

where $\mathbf{H}^{(k)} \in \mathbb{R}^{|V| \times d}$ denotes the matrix of node embeddings at layer k and $\mathbf{A}$ is the adjacency matrix. The bias term has been omitted for simplicity.

In a basic GNN framework, message passing resembles operations in a traditional multi-layer perceptron (MLP), involving linear transformations followed by an elementwise non-linear activation.

Often, the input graph is simplified by adding self-loops, eliminating the need for an explicit update step:

$$\mathbf{h}_u^{(k)} = \text{AGGREGATE}\left(\{\mathbf{h}_v^{(k-1)}, \forall v \in \mathcal{N}(u) \cup \{u\}\}\right), \tag{3.6}$$

where aggregation now includes both the neighbors $\mathcal{N}(u)$ and the node u itself. This approach can help reduce overfitting but restricts the GNN's expressiveness, as the information coming from the node's neighbours cannot be differentiated from the information from the node itself. For basic GNNs, incorporating self-loops is equivalent to sharing parameters between the $\mathbf{W}_{\text{self}}$ and $\mathbf{W}_{\text{neigh}}$ matrices, which gives the following graph-level update:

$$\mathbf{H}^{(k)} = \sigma\left(\left(\mathbf{I} + \mathbf{A}\right)\mathbf{H}^{(k-1)}\mathbf{W}^{(k)}\right). \tag{3.7}$$

This foundational framework sets the stage for exploring specialized GNN architectures, including Graph Convolutional Networks (GCNs) [33], Graph Attention Networks (GATs) [34], SuperGATs [35], and GraphTransformers [71, 36], each implementing distinct message-passing mechanisms.

3.4 Graph Convolutional Networks (GCNs)

In Eq.3.4, the aggregation operator of the basic GNN simply takes the sum of the neighbor embeddings:

$$\mathbf{m}_{\mathcal{N}}(u) = \text{AGGREGATE}^{(k)}\left(\{\mathbf{h}_v^{(k-1)}, \forall v \in \mathcal{N}(u)\}\right) = \sum_{v \in \mathcal{N}(u)} h_v\,. \tag{3.8}$$

This approach is unstable and highly sensitive. A straightforward solution is to sum the embeddings of neighboring nodes and then normalize it by the degree of the target node:

$$\mathbf{m}_{\mathcal{N}}(u) = \frac{\sum_{v \in \mathcal{N}(u)} h_v}{|\mathcal{N}(u)|}. \tag{3.9}$$

In 2016, Kipf et al. [33] 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:

$$\mathbf{m}_{\mathcal{N}}(u) = \sum_{v \in \mathcal{N}(u)} \frac{h_v}{\sqrt{|\mathcal{N}(u)||\mathcal{N}(v)|}}. \tag{3.10}$$

This adjustment became a defining feature of GCNs, as it ensures that each neighbor's contribution is weighted according to the connectivity patterns of the graph. GCNs became one of the most influential GNN models, combining this symmetric-normalized aggregation operation with self-loop updates. The GCN message passing function is defined as follows [2]:

$$\mathbf{h}_u^{(k)} = \sigma\left(\mathbf{W}^{(k)} \sum_{v \in \mathcal{N}(u) \cup \{u\}} \frac{h_v}{\sqrt{|\mathcal{N}(u)||\mathcal{N}(v)|}}\right). \tag{3.11}$$

In GCNs, this aggregation mechanism is coupled with a feature transformation via the weight matrix $\mathbf{W}^{(k)}$ and a non-linear activation function σ, such as ReLU, to improve model expressivity. The key aspect of the GCN approach is that we can build powerful models by stacking very simple graph convolutional layers. A basic GCN layer is defined in Kipf et al. [33] as:

$$\mathbf{H}^{(k)} = \sigma\left(\tilde{\mathbf{A}}\mathbf{H}^{(k-1)}\mathbf{W}^{(k)}\right), \tag{3.12}$$

where $\tilde{\mathbf{A}} = (\mathbf{D} + \mathbf{I})^{-\frac{1}{2}}(\mathbf{I} + \mathbf{A})(\mathbf{D} + \mathbf{I})^{-\frac{1}{2}}$ is the normalized variant of the adjacency matrix (with self-loops), and $\mathbf{D}$ is the diagonal node degree matrix.

By iterating over this propagation rule, GCNs enable each node to incorporate information from increasingly larger neighborhoods, allowing deeper insights into the graph structure. Thus, the concept of message-passing can be interpreted as a simple form of graph convolution, enhanced by the integration of trainable weights and non-linear activation functions [2]. This approach balances simplicity and effectiveness, making GCNs one of the most widely used architectures in the field of graph neural networks.

3.5 Graph Attention Networks (GATs)

Real-world graphs frequently contain noisy connections between unrelated nodes, which can lead GNNs to learn less effective representations [35]. Although normalizing neighborhood information can improve GNN performance, more sophisticated methods exist to further refine the aggregation process. Instead of simply summing or averaging neighboring embeddings, an effective approach is to apply an attention mechanism, as initially popularized by Bahdanau et al. in neural machine translation [72]. The attention mechanism assigns unique weights to each neighbor, determining their relative importance during aggregation. Veličković et al. introduced this approach to GNNs through their Graph Attention Network (GAT) model [34], where the aggregation step becomes a weighted sum of neighbor embeddings:

$$\mathbf{m}_{\mathcal{N}}(u) = \sum_{v \in \mathcal{N}(u)} \alpha_{u,v} h_v \,. \tag{3.13}$$

Then, the message-passing mechanism operator is:

$$h_u^{(k)} = \alpha_{u,u}^{(k)} \mathbf{W}^{(k)} h_u^{(k-1)} + \mathbf{W}^{(k)} \sum_{v \in \mathcal{N}(u)} \alpha_{u,v}^{(k)} h_v^{(k-1)} \,. \tag{3.14}$$

Here $\alpha_{u,v}$ represents the attention weight for neighbor $v \in \mathcal{N}(u)$ when aggregating information for node u. In the original GAT model, these attention weights are calculated as:

$$\alpha_{u,v}^{(k)} = \frac{\exp\left(\text{LeakyReLU}\left(\mathbf{a}^{(k)\top}[\mathbf{W}^{(k)}\mathbf{h}_u^{(k-1)} \oplus \mathbf{W}^{(k)}\mathbf{h}_v^{(k-1)}]\right)\right)}{\sum_{v' \in \mathcal{N}(u)\cup\{u\}} \exp\left(\text{LeakyReLU}\left(\mathbf{a}^{(k)\top}[\mathbf{W}^{(k)}\mathbf{h}_u^{(k-1)} \oplus \mathbf{W}^{(k)}\mathbf{h}_{v'}^{(k-1)}]\right)\right)}, \tag{3.15}$$

where $\mathbf{a}$ is a learnable attention vector, $\mathbf{W}$ is a trainable matrix, and $\oplus$ denotes the concatenation operation [2, 34]. This setup enables the GAT to give different levels of influence to each neighbor based on their relevance to the target node, in other words, the degree of importance of each of the neighbors to represent the center node. Thus graph attention results in a flexible and adaptive aggregation process.

To further enhance the effectiveness of GATs, especially in noisy graphs, Supervised Graph Attention Networks (SuperGAT) introduce a mechanism that supervises the attention process. Proposed by Kim Et. Al. [35], SuperGAT builds on the GAT model by using supervision signals to help the model prioritize informative edges and down-weight irrelevant connections.

SuperGAT modifies the attention mechanism by adding an auxiliary task that penalizes attention weights for edges that are likely to be noisy or uninformative, while boosting weights for edges aligned with label information or prior knowledge, guiding attention with the presence or absence of an edge between a node pair. This supervisory signal adjusts the attention scores $\alpha_{u,v}$, refining the aggregation process to prioritize relevant neighborhood information.

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 — referred to as GO attention in the paper — and the dot-product attention (DP). The GO attention computes attention coefficients using a single-layer feed-forward network parameterized by a learnable attention vector $\mathbf{a}$ [34], as defined in Eq.3.15. Meanwhile, the DP attention employs a dot-product operation between node feature vectors [73, 74]. Mathematically, these attention mechanisms are expressed as follows:

$$e_{u,v}^{\mathrm{GO}} = \mathbf{a}^{(k)\top}[\mathbf{W}^{(k)}\mathbf{h}_u^{(k-1)} \oplus \mathbf{W}^{(k)}\mathbf{h}_v^{(k-1)}]\,, \tag{3.16}$$

$$e_{u,v}^{\mathrm{DP}} = \left(\mathbf{W}^{(k)}\mathbf{h}_u^{(k-1)}\right)^{\top} \cdot \mathbf{W}^{(k)}\mathbf{h}_v^{(k-1)}\,, \tag{3.17}$$

$$e_{u,v}^{\mathrm{MX}} = e_{u,v}^{\mathrm{GO}} \cdot \sigma(e_{u,v}^{\mathrm{DP}})\,. \tag{3.18}$$

The combined attention $\alpha_{u,v}^{\mathrm{MX}}$ is then computed as:

$$\alpha_{u,v}^{(k)} = \frac{\exp\left(\mathrm{LeakyReLU}\left(e_{u,v}^{\mathrm{MX}}\right)\right)}{\sum_{v' \in \mathcal{N}(u)\cup\{u\}} \exp\left(\mathrm{LeakyReLU}\left(e_{u,v'}^{\mathrm{MX}}\right)\right)}. \tag{3.19}$$

Additionally, SuperGAT employs a self-supervised task of LP to enhance the learning of attention weights. In this task, the attention values are used to predict the likelihood $\phi_{u,v}^{\mathrm{MX}}$ of an edge existing between two nodes:

$$\phi_{u,v}^{\mathrm{MX}} = \sigma\left(\left(\mathbf{W}^{(k)}\mathbf{h}_u^{(k-1)}\right)^{\top} \cdot \mathbf{W}^{(k)}\mathbf{h}_v^{(k-1)}\right)\,. \tag{3.20}$$

In Kim et al. [35], SuperGAT demonstrates significant improvements in robustness compared to traditional GATs, particularly in noisy and complex graph environments. By leveraging a hybrid attention mechanism and self-supervised tasks, SuperGAT is able to capture more nuanced relationships in the graph, resulting in highly accurate and expressive node representations. These improvements make it a promising model for applications requiring robust graph-

based learning, such as social network analysis, recommendation systems, and knowledge graph completion.

3.6 Graph Transformer

In recent years, the graph transformer has emerged as a powerful and versatile approach for graph learning, garnering significant interest across both academic and industry sectors [75]. Graph transformer research is inspired by the success of transformers in natural language processing (NLP) [73] and computer vision (CV) [76], combined with the established strengths of GNNs. By integrating graph-specific inductive biases — such as inherent assumptions about structural relationships and properties — graph transformers offer a robust framework to process complex graph data effectively. Furthermore, they can adapt to dynamic and heterogeneous graphs, leveraging both node and edge features and attributes.

The literature on graph transformers is extensive, addressing various approaches to applying transformers to graph-structured data [71]. In this project, we leverage the multi-head attention mechanism proposed by Shi et al. [36]. This approach adapts the traditional self-attention mechanism for graph data, aligning with the principles of the GAT [34], which restricts nodes to attend primarily to their local neighbors. Shi et al. extend this concept by adopting the vanilla multi-head attention framework from the original transformer architecture [73] and customizing it for graph learning tasks.

Formally, the message-passing step is:

$$h_u^{(k)} = \mathbf{W}_1^{(k)} h_u^{(k-1)} + \mathbf{W}_2^{(k)} \sum_{v \in \mathcal{N}(u)} \alpha_{u,v}^{(k)} h_v^{(k-1)} , \tag{3.21}$$

where the attention coefficients $\alpha_{u,v}$ are computed using a multi-head dot-product attention mechanism:

$$\alpha_{u,v} = \text{softmax}\left(\frac{(\mathbf{W}_3^{(k)} \mathbf{h}_u^{(k-1)})^\top (\mathbf{W}_4^{(k)} \mathbf{h}_v^{(k-1)})}{\sqrt{d}} \right), \tag{3.22}$$

where $\mathbf{W}_1$, $\mathbf{W}_2$, $\mathbf{W}_3$ and $\mathbf{W}_4$ are learnable weight matrices, and d is a scaling factor based on the dimensionality of the hidden layer.

This multi-head attention mechanism enables the model to capture a wide range of relational patterns by attending to different aspects of neighborhood information across multiple heads. This approach facilitates the model's ability to discern complex relationships and depen-

dencies within the graph, setting a new standard for high-performance graph-based learning tasks.

In the next chapter, we delve into one of the most impactful applications of GNNs: Link Prediction.

4

Link Prediction

The primary goal of link prediction (LP) is to determine whether two nodes in a network are likely to form a connection [16]. Illustrated in Figure 4.1, for undirected, unweighted graphs, the LP problem seeks to identify missing edges in a partial or incomplete version of the graph, denoted as $\mathcal{G}'$, which is a subset of the complete graph $\mathcal{G}$.

Given the prevalence of networks across various domains, LP has numerous applications. For example, in criminal networks [77, 78], LP aids law enforcement by analyzing relationships and interactions to uncover illicit activities, such as drug trafficking or money laundering, and identifying connections between individuals involved. In social networks [61, 79], LP facilitates the discovery of potential connections, helping users find people they may know but have not yet connected with. Similarly, in recommender systems [80, 81], LP predicts new items, products, or services for users based on their preferences and actions, enhancing customer satisfaction and driving sales. Lastly, in biological networks, particularly for predicting protein interactions [82, 83], LP algorithms infer new interactions between proteins based on existing data, enabling researchers to propose hypotheses about the roles of previously unknown proteins.

As discussed in the Related Work — Chapter 2 — LP algorithms are not only essential for these applications but also influence network properties such as social dynamics, connectivity, and information diffusion [58]. Previous studies, for example, have highlighted concerns around biases in LP models, which may inadvertently create "filter bubbles" [15] or reinforce high-degree nodes, potentially leading to a "rich-get-richer" effect [51, 52].

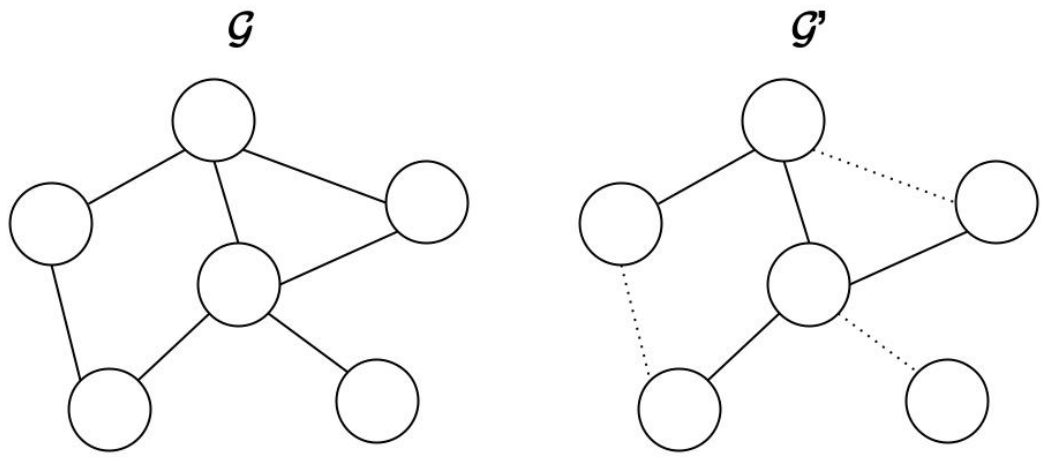

Figure 4.1: Illustration of the link prediction problem: the left side depicts the complete graph $\mathcal{G}$, while the right side shows its incomplete version, $\mathcal{G}'$, where the model will be applied to predict the missing edges.

Existing LP methods can be categorized into three main types: heuristic, latent-feature, and content-based approaches [84]. Heuristic methods predict link likelihoods by calculating similarity scores between node pairs. These approaches are further divided based on the scope of the information they use: local, quasi-local, and global indices [85]. Local indices rely on the immediate neighborhood of nodes, considering paths of length less than two. Prominent examples include the Common Neighbors metric [16], the Adamic-Adar Index [61], the Preferential Attachment Index [86], and the Jaccard Index [87]. Global indices, on the other hand, utilize information from the entire network, considering paths longer than two. Notable examples include the Katz Index [88], Random Walk-based methods [89], and SimRank [59]. Quasi-local indices strike a balance between the two, combining the efficiency of local methods with the broader perspective of global ones. These include techniques like the Local Random Walk [90] and the Local Path Index [91], which account for paths up to a distance of two.

Latent-feature methods, on the other hand, compute low-dimensional latent representations of nodes, typically obtained by factorizing a matrix derived from the network, such as the adjacency matrix or the Laplacian matrix. These latent features are not directly observable and must be learned through optimization processes. Unlike explicit node features, where each dimension corresponds to a specific, interpretable property, the dimensions of latent features lack interpretability; their meaning is not inherently understood.

One of the most widely used latent-feature approaches is matrix factorization [92], originat-

ing from the recommender systems literature. Prominent network embedding techniques such as DeepWalk [31], LINE [32], and node2vec [69] are also categorized as latent-feature methods, as they implicitly perform matrix factorization [84]. These methods incorporate global network properties and long-range effects into the learned node representations. Since the optimization involves all node pairs, the final embedding of a node is influenced by every other node within the same connected component of the graph. However, latent-feature methods exhibit certain limitations. They fail to capture structural similarities between nodes [84]; for instance, two nodes with identical neighborhood structures may not be assigned similar embeddings. Additionally, these methods often require extremely high-dimensional embeddings to represent even simple heuristics effectively [17], which can lead to worse performance compared to heuristic methods in certain scenarios. Both heuristic and latent-feature methods rely on the network's existing structure to infer potential or missing links.

In contrast, content-based methods leverage explicit node attributes or features, rather than network topology alone, to make predictions [84]. Combining these node features with the network structure has been shown to enhance prediction accuracy [17, 18]. Recently, Graph Neural Networks (GNNs) have emerged as powerful tools for LP, integrating information from both graph topology and node or edge features. GNNs typically outperform traditional approaches [26, 27, 28, 29].

GNNs provide a powerful framework for handling multiple classification tasks in graphs, including graph, node, and edge classification. LP is specifically an edge classification task, where the GNN learns to predict the presence or absence of an edge between pairs of nodes. In other words, the model determines if an edge $E(i, j) = 1$ (exists) or $E(i, j) = 0$ (does not exist) between nodes (i, j), effectively addressing LP as a binary classification problem at the edge level.

Within GNN-based LP, two main paradigms have gained popularity: node-based and subgraph-based methods. Node-based approaches learn node representations through GNNs and then aggregate pairs of these representations to construct link representations. Examples include the Graph Convolutional Network (GCN) [33], the Graph Attention Network (GAT) [34], the SuperGAT [35], and the GraphTransformer [71, 36], each implementing distinct message-passing mechanisms. Subgraph-based approaches, alternatively, extract local subgraphs around each target link and apply a graph-level GNN (using pooling) to these subgraphs to create link representations. A prominent example of this method is SEAL [28]. Our experimentation will only cover node-based models.

It is worth to point out the importance of negative sampling in GNN-based LP models. A

widely-used LP benchmark evaluates methods by their ability to classify pairs of nodes as either connected or unconnected [51]. The connected node pairs are randomly sampled from existing edges as the hidden positive set, and an equal number of node pairs are randomly chosen from unconnected node pairs, which are far more common because of the sparsity of edges in graphs [93]. By including a balanced mix of positive and negative samples during training, negative sampling helps the model to differentiate real connections from random node pairs, thus enhancing the predictive power of LP models and avoiding overfitting to the training data.

4.1 Framework and Metrics

The commonly benchmark for LP consists in the following procedure. First, a subset of edges from the edge set E is randomly selected as positive examples, representing a fraction of the total edges. Then, an equal number of non-connected node pairs are randomly chosen from the node set V as negative examples. Any negative edges that create loops or overlap with either positive or test edges are resampled to avoid duplication. After preparing the dataset, a LP method assigns a score s_{ij} to each node pair (i, j), where higher scores indicate a stronger likelihood of an edge existing between the nodes. The method's predictive performance is then evaluated by calculating the Area Under the Receiver (AUC-ROC), which reflects the probability of assigning a higher score to positive edges over negative ones. While there are alternative approaches that might vary in negative sampling or use different metrics, this particular framework remains widely accepted in the field [51, 94, 95, 96, 97, 98].

The Area Under the Receiver (AUC-ROC) is the area under the Receiver Operating Characteristic Curve (ROC) [99], which is a measure of the model's ability to distinguish between positive and negative links. The ROC curve plots the true positive rate (TPR) — also called recall — against the false positive rate (FPR) at various threshold levels, effectively summarizing the trade-off between sensitivity and specificity across different decision boundaries. TPR and FPR can be computed using the following formulas:

$$\text{TPR} = \frac{TP}{TP + FN}\,, \tag{4.1}$$

$$\text{FPR} = \frac{FP}{FP + TN}\,. \tag{4.2}$$

An AUC-ROC score of 0.5 indicates no discriminative power, equivalent to random guessing, while a score of 1.0 represents perfect discrimination between positive and negative links.

In the context of LP, a higher AUC-ROC reflects the model's ability to correctly rank true links (existing edges) higher than false links (non-existent edges). This metric is particularly valuable because it is threshold-independent, meaning it evaluates the model's overall ranking performance rather than its accuracy at a specific threshold, making it robust across diverse datasets and applications.

The AUC-ROC score is commonly used as a primary evaluation metric for training LP models. However, as highlighted in [37, 51], relying solely on AUC-ROC may not provide a complete picture of a model's performance. To address this, an additional local measure called VCMPR@k is often computed after model training. According to [37], the widely-used AUC-ROC metric for LP using node embeddings can be misleading, as it does not account for sparse ground truths effectively. Their findings suggest that low-dimensional embeddings, often evaluated with AUC-ROC, struggle to capture sparse relationships when similarity is based on dot products, leading to inflated AUC scores that do not reflect true predictive quality.

Given that LP ground truth is typically sparse, they propose a vertex-centric measure of performance, called the VCMPR@k. For each pair of vertices (i, j), the model computes a score based on which it predicts an edge. For a given vertex i of nonzero degree d_i, all other vertices j are ranked in decreasing order of their scores, and pairs from E_{train} are removed. From this ordered list, the top k scores are selected and $t_i(k)$ is defined as the number of ground truth edges within these top k predictors. VCMPR@k differs from AUC-ROC in that it is a local metric, evaluated individually for each vertex, rather than a global metric.

For a given vertex i with nonzero degree d_i, the metric is defined in Aiyappa et al. [51] as follows:

$$\text{VCMPR@}k \text{ for vertex } i = \frac{t_i(k)}{max(k, d_i^{\text{test}})}, \tag{4.3}$$

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. This formulation balances between precision@k $\left(\frac{t_i(k)}{k}\right)$ and recall@k $\left(\frac{t_i(k)}{d_i^{\text{test}}}\right)$.

It is worth noting that the objective of this work is not to perfectly optimize LP performance as measured by VCMPR@k or AUC-ROC. Instead, we aim to understand how these models impact social diffusion processes on the resulting networks. Thus, our focus is on the effects of LP models in this context, rather than maximizing predictive accuracy.

5

Social Contagion Dynamics

Most collective behaviors spread through social contact, making social networks crucial for understanding a wide range of phenomena. From the emergence and reinforcement of social norms to the widespread adoption of technological innovations and the growth of social movements, these behaviors often propagate as "social contagions" across the connections between individuals. The pathways and mechanisms of these contagions are strongly influenced by the underlying network structure.

Studies of social diffusion dynamics have consistently shown that the topology of a social network—its arrangement of nodes and edges—plays a pivotal role in shaping how collective behaviors emerge and spread [64]. Key structural features, such as network density, clustering, centralities, and the presence of weak or strong ties, can determine the speed, reach, and stability of these behaviors. For instance, tightly clustered networks may facilitate the adoption of social norms through reinforcement, while networks with diverse bridges between communities may enable innovations to diffuse more broadly [63]. These insights highlight the critical interplay between network structure and the dynamics of social contagions, offering valuable tools for predicting and influencing collective behavior.

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 and complex contagion. These types differ in the mechanisms by which a contagion event occurs:

- Simple Contagion: A single interaction between a susceptible and an infected node is sufficient for the contagion to propagate. This type of contagion is commonly used to model phenomena such as infectious disease spread.
- Complex Contagion: Multiple reinforcing interactions are required for a contagion event. This is more representative of social behaviors or innovations, where peer influence and thresholds play a crucial role.

For simple contagions, the transmission mechanism operates at the level of individual connections, with each interaction having an independent probability of causing transmission. In contrast, complex contagions rely on cumulative exposure: a node becomes "infected" only when a sufficient proportion of its neighbors are already infected. This distinction highlights the divergent requirements for propagation in different contexts. For example, while simple contagions thrive in random or sparsely connected networks with long ties, complex contagions require clustered networks that facilitate reinforcement through local interactions [22, 25].

The consequences of these dynamics are profound. In small-world networks, simple contagions accelerate due to the presence of shortcuts that reduce path lengths. Conversely, complex contagions often stall in such networks because long ties do not provide the reinforcement necessary for transmission. For complex contagions to succeed, clustered structures with wide bridges—multiple paths of interaction—are essential [25]. As a result, the interplay between network topology and contagion type shapes not only the speed and reach of diffusion but also its vulnerability to disruptions [63].

To simplify the analysis, the Susceptible-Infected (SI) framework is commonly adopted, where each node in the network exists in one of two states: susceptible (S) or infected (I). In this model, infected nodes do not recover, making it particularly suitable for studying the unidirectional spread of phenomena. The processes are analyzed in discrete time, with variations arising solely from the mechanisms governing the transition from the susceptible to the infected state:

- Simple Contagion: Each susceptible node can independently be infected by its infected neighbors with a probability β per unit time.
- Complex Contagion: Modeled as a threshold process, a susceptible node u becomes infected when the fraction of its infected neighbors exceeds its predefined threshold θ^u. The thresholds for nodes are drawn from a truncated normal distribution with specified mean μ_θ and standard deviation σ_θ.

These two processes capture the core differences between simple and complex contagion. The following section delve into the details of the SI model. Figure 5.1 illustrates the fundamental mechanisms of simple and threshold contagion.

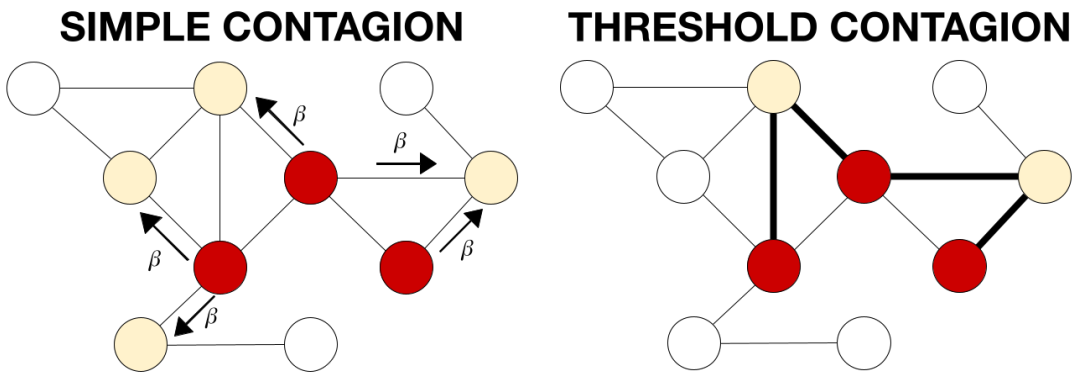

Figure 5.1: Mechanisms of contagion: (left) simple contagion propagates through pairwise interactions with probability β per unit time for each edge; (right) complex contagion occurs when the fraction of infected neighbors exceeds a threshold θ (here $\theta = 0.5$).

5.1 Susceptible-Infectious Model

The SI (Susceptible-Infectious) model is one of the simplest frameworks for studying contagion processes on networks. In this model, each node in the network exists in one of two states: Susceptible (S) or Infected (I). The model assumes that once a node becomes infected, it remains in that state indefinitely, meaning there is no recovery or removal mechanism. The dynamics of the SI model are entirely governed by the interactions between susceptible and infected nodes, making it an ideal framework for modeling the early stages of an outbreak or processes without recovery. The network's degree distribution, clustering coefficient, and average path length are critical factors that determine the speed and reach of the contagion.

In simple contagion the probability of a susceptible node u becoming infected at time t can be expressed as:

$$P(\text{node } u \text{ becomes infected at time } t) = 1 - \prod_{v \in \mathcal{N}(u)} (1 - \beta \delta_v(t-1)), \tag{5.1}$$

where $\mathcal{N}(u)$ denotes the set of neighbors of node u, $\delta_v(t-1)$ is an indicator function that equals 1 if node v is infected at time $t-1$, and 0 otherwise, and β is the transmission probability.

In threshold contagion, the infection probability is defined as:

$$P(\text{node } u \text{ becomes infected at time } t) = \begin{cases} 1 & \text{if } \sum_{v \in \mathcal{N}(u)} \delta_v(t-1) \geq \theta^u \\ 0 & \text{if } \sum_{v \in \mathcal{N}(u)} \delta_v(t-1) < \theta^u \end{cases}, \quad (5.2)$$

where θ^u is the threshold value of node u, representing the minimum number of infected neighbors required for u to become infected.

In the next chapter, we will present the datasets and methods employed in this study. This includes a detailed description of the data used to model and analyze the contagion processes, as well as the methodologies and algorithms implemented to study the dynamics and evaluate the outcomes of the proposed frameworks. These foundational elements are crucial for understanding the results and insights discussed in the subsequent sections.

6 Data and Methodology

This chapter provides a comprehensive overview of the foundational components used in this study, including the datasets, methodologies, and analytical frameworks. First, the datasets employed for experimentation are introduced, detailing their key characteristics and relevance to the study. This section also formalizes the graph-level topological metrics and node-level centralities, which are essential for characterizing the structural properties of the data. Next, the link prediction (LP) framework is described, outlining its role in modeling and analyzing network connectivity. The chapter then presents the social contagion framework, emphasizing the mechanisms and assumptions driving the diffusion dynamics. Finally, the metrics derived from these frameworks are introduced, providing the tools to analyze and interpret the relationship between LP and the dynamics of social contagion.

6.1 Data

We conducted link prediction (LP) experiments using six diverse datasets sourced from the PyTorch Geometric (PyG) library [100]. Table 6.1 provides a comprehensive overview of the key statistical characteristics for each dataset. These datasets were carefully selected to represent a wide range of network structures and feature compositions, enabling a robust evaluation of our proposed method.

- Cora dataset [101, 102]: This citation network comprises 2,708 machine learning papers categorized into seven classes. Each paper (node) is represented by a binary word

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

Table 6.1: Overview of nodes, edges, and feature dimensions for the datasets used in experiments.

vector indicating the absence/presence of the corresponding word from the dictionary. The dictionary consists of 1433 unique words. Edges indicate citation relationships. The dataset was collected by parsing research papers categorized by topic and processing their abstracts into a bag-of-words representation, creating an undirected graph for semi-supervised learning and node classification.

- CiteSeer dataset [102, 103]: Similar to Cora, this citation network comprises 3,327 research papers from the CiteSeer digital library, categorized into six classes. Nodes represent papers, with features as word vectors, and edges indicate citation links.

- Facebook dataset [104, 105]: This dataset includes anonymized data of users' ego networks on Facebook. Profile and network data were collected from 10 ego-networks, encompassing 193 circles and 4,039 users. The authors developed a custom Facebook application and surveyed ten users, who manually identified all social circles their friends belonged to. On average, each user identified 19 circles within their ego-network, with an average circle size of 22 friends. These circles typically represented social groups such as university peers, sports teams, family members, and others.

- Wikipedia dataset [105]: This network captures the link structure of Wikipedia pages, forming a network of articles and their references. It includes thousands of nodes, where each node represents a Wikipedia article, and edges denote hyperlinks between articles. The dataset enables analysis of information flow and topic clustering within Wikipedia.

- Twitch ES dataset [106]: This is a network of Twitch users (gamers) in Spain, where nodes correspond to individual gamers and edges signify followerships between them. Node features include embeddings that capture the types of games played by each user. The primary task associated with this dataset is to predict whether a user streams mature content, making it useful for research on content classification and community detection within gaming networks.

- LastFMAsia dataset [107]: The LastFMAsia graph represents a social network of users from various Asian countries, such as the Philippines, Malaysia, and Singapore. Nodes correspond to users of the music streaming service LastFM, while edges represent friendships between them. This dataset was collected in March 2020 using the LastFM API. The associated classification task involves predicting a user's home country based on their position in the social network and the artists they like.

For a comprehensive exploratory analysis, we present the metrics used to describe network structures at both the node and graph levels in the following sections. In all cases we use the undirected, unweighted version of the networks.

6.1.1 Node Centrality Measures

Beyond simply analyzing a node's degree, we can gain substantial insights by examining various centrality measures that capture a node's influence, connectivity, and strategic position within the network. Centrality metrics help reveal how integral each node is to network structure and function—whether it acts as a hub, a bridge, or a well-connected influencer. Each of the following centrality measures provides unique information about a node's role and significance in facilitating network dynamics, such as the flow of information or the spread of influence.

We focus on the following centrality measures:

- Degree centrality: Measures the importance of a node based on its number of direct connections, with a higher degree indicating greater influence within the network [108]. Mathematically, the degree centrality of a node u is:

$$\mathrm{DC}(u_i) = \frac{1}{N-1} \sum_{j=1}^{N} \alpha_{i,j} \,, \tag{6.1}$$

  where N is the number of vertices, and $\alpha_{i,j} = 1$, if there is a direct link between u_i and u_j such that $u_i \neq u_j$, or $\alpha_{i,j} = 0$ if there is no connection of $i = j$.

- Betweenness centrality: Evaluates the significance of a node by counting how often it serves as a bridge along the shortest path between two other nodes. This reflects its role in controlling information flow across the network [108]. Mathematically, the betweenness centrality of a node u is defined as the sum of the fraction of all-pairs shortest paths that pass through u:

$$\mathrm{BC}(u) = \sum_{s,t \in V} \frac{\sigma(s,t|u)}{\sigma(s,t)} \tag{6.2}$$

where V is the set of nodes, $\sigma(s,t)$ is the number of shortest (s,t)-paths, and $\sigma(s,t|u)$ is the number of those paths passing through some node u other than s and t. If $s=t$, $\sigma(s,t)=1$, and if $u \in \{s,t\}$, $\sigma(s,t|u)=0$ [109].

- Eigenvector centrality: Determines a node's centrality by considering not only its direct connections but also the centrality of those connected nodes. Nodes linked to highly connected nodes receive higher scores, emphasizing influence in the network [108]. Eigenvector centrality for a node i is the i-th element of a left eigenvector associated with the eigenvalue λ of maximum modulus that is positive. Such an eigenvector x is defined up to a multiplicative constant by the equation

$$\lambda x^T = x^T A, \tag{6.3}$$

where A is the adjacency matrix of the graph $\mathcal{G}$. Using the properties of matrix multiplication (specifically, the row-column product), we can express each component of this equation individually

$$\lambda x_i = \sum_{j \to i} x_j, \tag{6.4}$$

where the summation is over the predecessors of i. Thus, the eigenvector centrality of i is obtained by adding the eigenvector centralities of its predecessors, multiplied by λ.

- Complex-path centrality [64]: Traditional centrality measures often use simple path lengths, which may overlook the structural features most effective for spreading complex contagions. To address this, Guilbeault et al. [64] introduce measures of complex path length and complex-path centrality, significantly enhancing the ability to identify network structures and key individuals for complex contagion. The complex-path centrality of a node i is the average length of complex paths originating from its neighborhood, *i.e.*, nodes that are at a distance of 1 from i. Here, a complex path between node i and node j is the sequence of neighborhoods through which a complex contagion must traverse to travel from the neighborhood of node i, $N(i)$, until reaching j, where $i,j \in V$. For the contagion threshold parameter T, representing the minimum fraction of activated peers required for a node to adopt the contagion, we have used $T=0.5$.

  Mathematically, the complex path length ($\mathrm{PL}_{C_{ij}}$ between $N(i)$ and node j is expressed as:

$$\mathrm{PL}_{C_{ij}} = |\phi(\mathrm{GEO}_{\mathrm{CP}_{ij}})|, \tag{6.5}$$

where $\phi(\mathrm{GEO}_{\mathrm{CP}_{ij}})$ represents the vertex sequence in the geodesic between node i and node j within CP_{ij}, which also identifies the shortest complex path within CP_{ij}. Here, CP_{ij} denotes the induced subgraph formed by nodes activated during the complex con-

tagion spread from $N(i)$ to node j, which contains the set of possible complex paths between $N(i)$ and node j.

The complex path centrality of node i is then calculated as the average complex path length, formally expressed as:

$$\mathrm{PL}_{C_i} = \frac{1}{n - |V(N(i))|} \cdot \sum_{i \neq j} \mathrm{PL}_{C_{ij}} \,, \tag{6.6}$$

where n represents the total number of nodes in the network, and $|V(N(i))|$ is the number of neighbors of node i.

- Diffusion centrality [11, 23]: This measure relates to Information Capital, *i.e.*, the ability to acquire valuable information and/or to spread it to other people who can use it through social connections. It is calculated as the sum, for all nodes j, of the expected number of times j will receive information originating from node i over T periods. Mathematically, the diffusion centrality of a node i in a network with an adjacency matrix $\mathbf{g}$, passing probability q, and iterations T, is the i-th entry of the vector

$$\mathrm{DC}(\mathbf{g}; q, T) = \left[\sum_{t=1}^{T} (q\mathbf{g})^t \right] \cdot \mathbb{1} \,. \tag{6.7}$$

The probability q is often selected as the inverse of the first eigenvalue of the adjacency matrix, $\lambda_1(\mathbf{g})$ [23]. In our analysis we set $T = 10$.

6.1.2 Graph Topological Measures

To describe the structural characteristics and properties of the network, we focus on a set of graph topological measures. These measures provide insights into the connectivity, clustering tendencies, and distribution of influence across nodes within the network. By analyzing these properties, we can better understand how the network is organized and how information, influence, or resources may flow through it.

- Average Degree: This measure calculates the average number of connections (edges) each node has within the graph. It provides a sense of the network's overall connectivity and the typical level of interaction between nodes.
- Global Clustering Coefficient: This metric quantifies the average local clustering coefficient of the nodes, measuring the tendency of nodes in a graph to form tightly-knit

groups or clusters. The local clustering coefficient for undirected graphs can be defined as:

$$C_i = \frac{2E_i}{k_i(k_i - 1)}, \tag{6.8}$$

where k_i is the number of neighbours of node i, and E_i is the number of actual connections among these k_i.

The global clustering coefficient indicates how well a node's neighbors are interconnected, highlighting the presence of local connections or "cliques" within the network. A higher global clustering coefficient suggests a greater prevalence of these local clusters [110].

- Gini Coefficient: This measure quantifies inequality in the distribution of a given centrality measure, *e.g.*, degree centrality, betweenness centrality, across nodes in the network. It provides insights into whether centrality is evenly spread or concentrated among a few influential nodes [111]. A low Gini Coefficient (close to 0) indicates that most nodes have similar centrality values, suggesting that the importance or influence is relatively evenly spread across the network. This indicates a decentralized or egalitarian structure, where no single node has significantly more influence than the others. On the other hand a high Gini Coefficient (close to 1) suggests that a small subset of nodes has significantly higher centrality than others,implying a concentration of influence or control within the network. In this case, a few nodes dominate in terms of their position in the network (*e.g.*, acting as hubs or bridges), while the majority of nodes have low centrality.

6.2 Methodology

Building upon our comprehensive dataset analysis presented earlier, this study delves into the intricate interplay between link prediction (LP) models and social contagion processes. By leveraging insights from the dataset's structural properties, the methodology is structured around the following key objectives:

- The impact of LP models: Investigating how different LP models influence the difference in social diffusion parameters between the real and predicted networks. Also how variation in the depth of the neural networks or the contagion probability correlate to diffusion parameters.
- Node characteristics and contagion dynamics: Analyzing how intrinsic node properties, such as centralities or the degree, correlate with their progression in the contagion process.

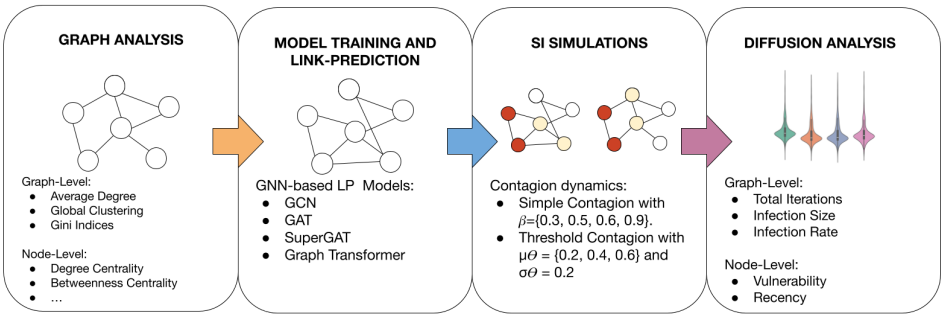

Figure 6.1: Experimental workflow for network modeling and diffusion analysis, encompassing graph analysis, model training, SI simulations, and diffusion metric evaluation.

- Graph topologies and social diffusion parameters: Studying how structural graph-level properties (*e.g.*, clustering coefficients, degree distributions, or Gini indices) relate to observed diffusion behaviors.

- Other exploratory analyses: Delving into additional factors influencing the interplay between network prediction and contagion diffusion.

To systematically address these objectives, we divided the analysis into two levels:

1. Graph-Level Study: Focused on understanding global structural properties and their relationships with diffusion metrics across entire networks.

2. Node-Level Study: Concentrated on individual nodes, exploring how their characteristics influence their role and performance in the contagion process.

This two-pronged approach allows us to uncover insights that span from overarching network structures to granular node-level dynamics, providing a comprehensive understanding of the interplay between LP, diffusion, and network topology.

6.2.1 Experimental Framework

Our experimental pipeline, depicted in Fig. 6.1, for each combination of *model-dataset-contagion process*, proceeds as follows:

1. Graph Analysis: We begin characterizing each dataset at both graph and node levels by computing the graph-level topologies and node-level centralities outlined in Sections 6.1.2 and 6.1.1, respectively. This analysis provides foundational insights into the structural and local properties of the datasets.

2. Model Training and Link Prediction (LP): Drawing insights from Shchur et al.'s comprehensive analysis [46], our experimental design employs a robust training methodology. Specifically, we train ten independent model iterations using different random train-validation-test splits, consistently maintaining a 60-10-30 data allocation strategy. The experimental configuration encompasses two primary Graph Neural Network (GNN) architectures:

   - Two-layer models featuring progressive hidden dimensions of 128 and 64 neurons.
   - Three-layer models with hierarchical hidden dimensions of 128, 64, and 64 neurons.

   Our training protocol leverages the Adam optimizer with a fixed learning rate of 0.01, optimizing model performance through the Binary Cross-Entropy with Logits loss function (BCEWithLogitsLoss). For each trained model, we systematically generate predicted and ground-truth network representations based on the held-out test set, enabling comprehensive performance evaluation.

   While the AUC-ROC serves as our primary evaluation metric for the LP models, recent literature [51, 37] cautions against its exclusive use. AUC-ROC, although widely adopted, can mask critical nuances in model performance by providing an aggregate measure that may not capture local network characteristics. The AUC-ROC score serves as the main metric for assessing the performance of LP models during training. Nevertheless, as emphasized in [51, 37], depending exclusively on AUC-ROC can overlook important aspects of a model's behavior. To provide a more nuanced evaluation, the VCMPR@k local measure, defined in Eq. 4.3, is also calculated after training the models. For this metric, k is chosen to be the average vertex degree of the test set.

3. SI Simulations: We conduct one hundred social contagion simulations for each network, employing a comprehensive Susceptible-Infected (SI) model — presented in Section 5.1 — that captures the nuanced dynamics of disease transmission across different network structures.

   The simulation framework follows a structured protocol:

   (a) Initialization:

   - Set all nodes to the susceptible state (S).
   - For simple contagion: Randomly select one node to be initially infected (I).

- For complex contagion: following Guilbeault et al.'s approach [64], the initialization process begins by randomly selecting 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. Specifically, the algorithm determines the number of neighbors to infect as the fraction of the node's total neighbor that would equal the threshold. This method captures the essence of complex contagion, where individuals require substantial social proof or validation before adopting new behaviors, beliefs, or technologies — a phenomenon typically observed in group dynamics and community-driven changes.

(b) Infection Dynamics: At each discrete time step, implement infection propagation:

- For simple contagion: Each infected node attempts to infect its susceptible neighbors with a fixed probability $\beta \in \{0.3, 0.5, 0.6, 0.9\}$.
- For 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 $\sigma_\theta = 0.2$.

(c) State Transition:

- Update the infection status of all nodes based on the specific contagion rules.
- Newly infected nodes are added to the infection pool.

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

Each network undergoes one hundred independent simulations to ensure robust and statistically meaningful results. This approach allows for a comprehensive exploration of infection dynamics across different network structures and contagion parameters.

4. Social Contagion Analysis: To analyze the contagion dynamics across these simulations, we measure the following 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:

$$\textit{Infection Rate} = \frac{\textit{Infection Size} \times \text{Number of Nodes}}{\textit{Iterations}} \tag{6.9}$$

- *Vulnerability* of node u: The proportion of simulations in which node u becomes infected, defined as:

$$\textit{Vulnerability}(u) = \frac{\text{Number of simulations where } u \text{ is infected}}{\text{Total number of simulations}}. \tag{6.10}$$

- *Recency* of node u: This metric quantifies how quickly a node typically becomes infected during the contagion process. It is defined as:

$$\textit{Recency}(u) = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{\text{timestep}_i(u) + 1}, \tag{6.11}$$

  where N is the total number of simulations, and $\text{timestep}_i(u)$ denotes the iteration at which node u is infected in the i-th simulation. A high *Recency* value indicates that the node is generally infected early in the contagion process, often signifying its importance in the contagion process and its high susceptibility to being infected. Conversely, a low *Recency* value suggests that the node is typically infected later, indicating a more peripheral role in the contagion dynamics.

This methodology provides a rigorous foundation for conducting an extensive analysis of the interplay between model architectures, dataset topologies, node-level centralities, and the dynamics of social contagion processes. In the following section, we delve into a detailed analysis of our experimental results, analyzing the datasets, exploring the impact of LP models on the diffusion patterns, the relationship between node characteristics and their infection susceptibility, and other relevant factors.

7 Results

In this chapter, we present the results of a comprehensive exploratory analysis aimed at understanding the interplay between network structure, link prediction (LP) models, and social contagion dynamics. By leveraging the metrics introduced in Section 6.2, we examine both node-level and graph-level characteristics to uncover patterns and relationships that influence social contagion diffusion behaviors. The analysis explores how GNN-based LP models affect diffusion outcomes, the role of structural properties in shaping social diffusion processes, and how node's position in the network correlate to their contagion susceptibility, among others. These results offer important perspectives on the core questions of the study, serving as an initial step toward bridging gaps in the existing literature and enhancing our understanding of LP and its impact on social dynamics.

7.1 Initial Exploratory Analysis

We start this chapter by exploring the structural characteristics of our datasets, focusing on node-level centrality measures and graph-level metrics. These analyses aim to uncover the underlying distribution patterns and topological properties that define each network, providing a foundation for understanding their influence on diffusion dynamics. Figure 7.1 presents violin plots of various centrality measures across datasets, offering a detailed view of their distributions. The observed skewness in these metrics, where most nodes have low centrality values and a few exhibit significantly higher values, underscores the inherent structural inequalities

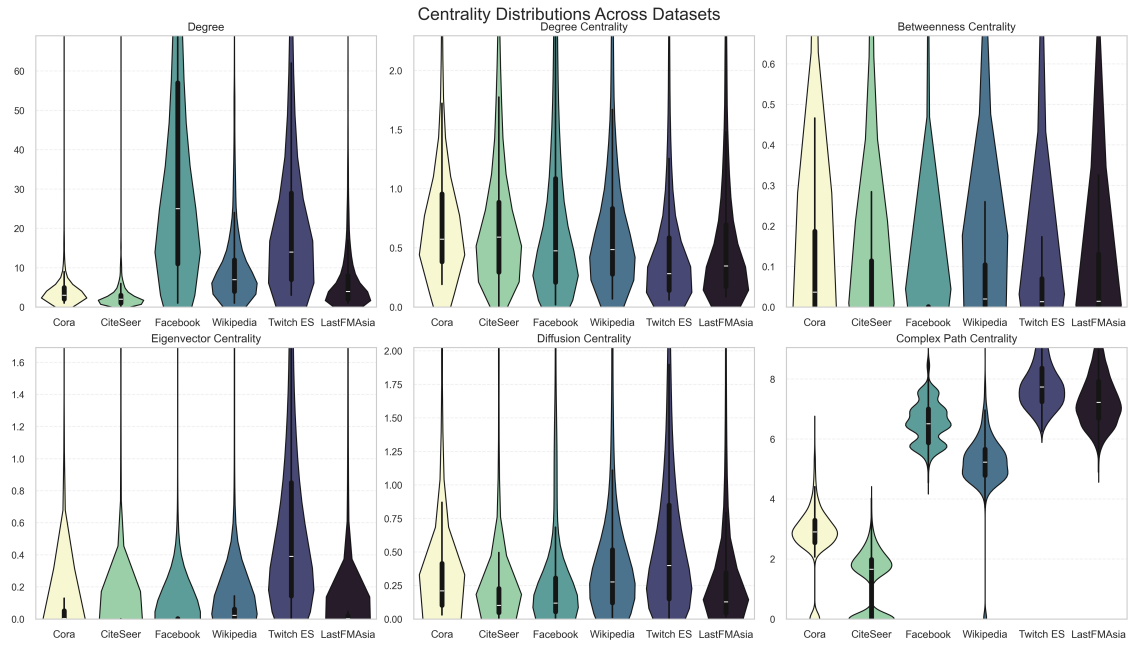

Figure 7.1: Distribution of network degrees and centrality measures across datasets, visualized through violin plots. Y-axes are truncated at the 95th percentile to highlight the core distribution patterns while excluding extreme outliers.

in these networks. Notably, *Facebook* and *Twitch ES* stand out as networks with higher degree distributions. Among the datasets, *Twitch ES* demonstrates significantly higher *Eigenvector Centrality*, reflecting its unique network dynamics. Furthermore, the distributions of *Complex Path Centrality* are particularly diverse, with lower values observed in *Cora* and *CiteSeer* and higher values in *Twitch ES* and *LastFMAsia*. This underscores the distinctive role of *Complex Path Centrality* in capturing network properties that differ from other centrality measures.

Additionally, we examine key graph-level measures, as shown in Figure 7.2, including *Average Degree*, *Clustering Coefficient*, and *Gini Coefficients* for various centrality measures. These metrics reveal significant variations across datasets, reflecting differences in connectivity, clustering tendencies, and inequalities in the distribution of centralities. *Facebook*, *Wikipedia*, and *Cora* stand out with higher *Clustering Coefficients*, while *Facebook* and *Twitch ES* exhibit the highest *Average Degree*, with *CiteSeer* having the lowest.

Regarding *Gini Coefficients*, most datasets exhibit similar values, except for *CiteSeer*, which shows notably higher *Gini Complex Path Centrality* compared to others, where it is generally low. Conversely, *Twitch ES* stands out with the lowest *Gini Eigenvector Centrality*. *Facebook* further distinguishes itself with the highest values for both *Gini Betweenness Centrality* and *Gini Eigenvector Centrality*, emphasizing its unique structural inequalities.

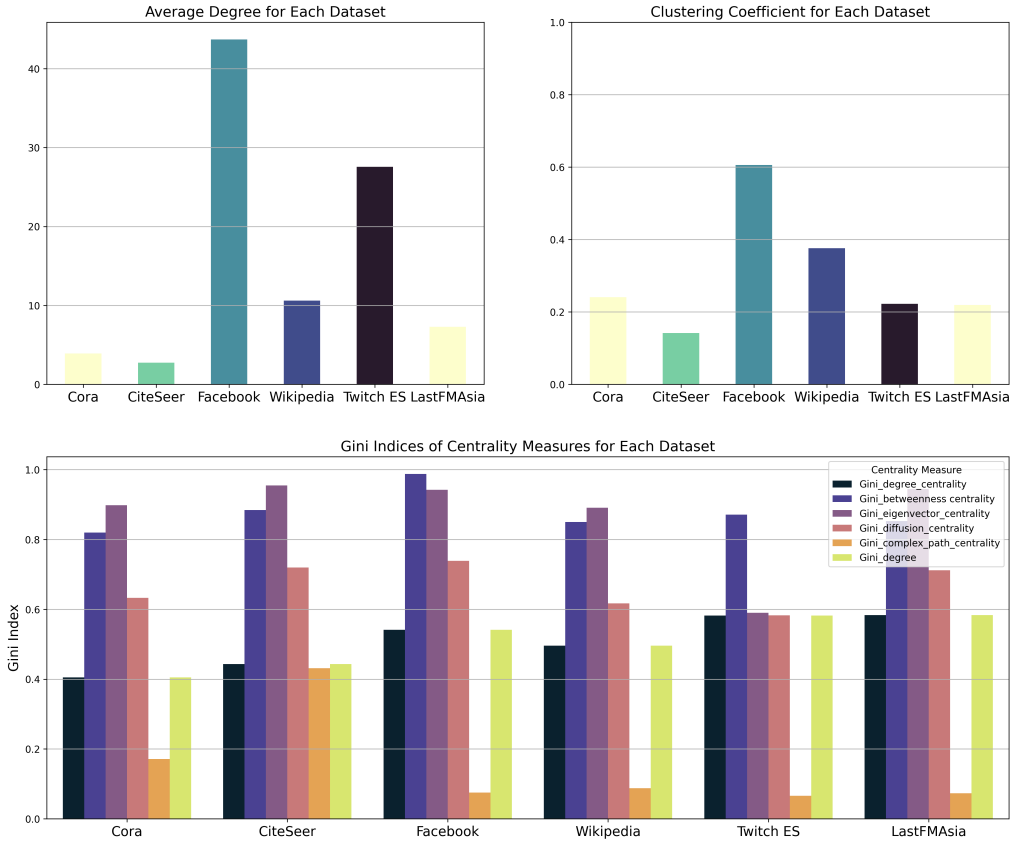

Figure 7.2: Graph-level measures for each dataset. The top two plots display the *Average Degree* and *Clustering Coefficient* for each dataset, while the larger plot below highlights the *Gini Coefficients* for various centrality measures, revealing the diversity in distribution inequality across networks.

7.2 Performance Evaluation

Now, we briefly introduce the link prediction (LP) evaluation metrics, discussed in detail in Section 4.1, for our models and datasets.

Figure 7.3 presents the bar plots of the AUC-ROC scores for each model across the datasets. *Facebook* and *Wikipedia* stand out with the highest AUC-ROC scores, demonstrating strong performance in accurately predicting links, while *CiteSeer* and *Twitch ES* exhibit the lowest overall scores, suggesting greater challenges for LP in these networks.

Among the LP models, GCN-based architectures consistently outperform the others across almost all datasets, underscoring their effectiveness in leveraging graph structure for LP tasks. The notable exception is *LastFMAsia*, where the attention-based mechanism of the GAT model

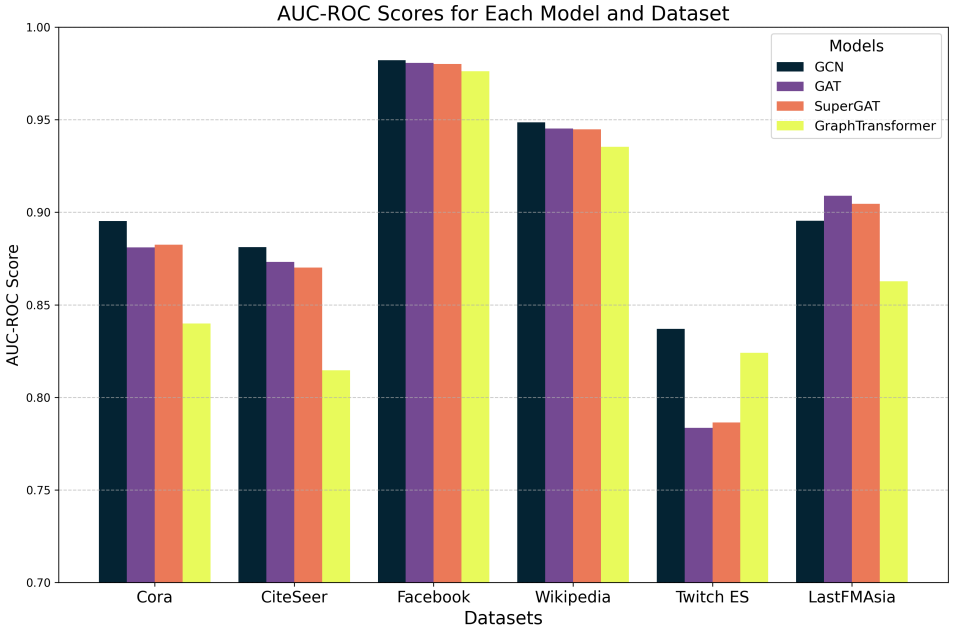

Figure 7.3: Barplots of the AUC-ROC score for each model and dataset.

surpasses other approaches.

These findings not only emphasize the variability in model performance across datasets but also underline the importance of selecting suitable LP algorithms tailored to the specific characteristics of the graph.

While the AUC-ROC score is commonly used to evaluate LP models, it may not fully capture a model's performance, particularly when ground truths are sparse [37, 51]. Low-dimensional embeddings often evaluated by AUC-ROC can lead to inflated scores, as they fail to effectively capture sparse relationships. To address this, the VCMPR@k measure is used as a local performance metric, providing a more accurate evaluation in sparse settings.

Figure 7.4 shows that VCMPR@k distributions are skewed, though not excessively, aligning with the observations of Menand et al. [37]. Overall, the various LP models produce comparable distributions. Interestingly, the *LastFMAsia* dataset exhibits the lowest VCMPR@k scores, a result that contrasts with its relatively strong AUC-ROC scores. Another notable finding is the performance of the GCN model on the *Twitch ES* network, which displays a distinct highly left-skewed distribution compared to other models. This divergence is particularly striking given that the GCN model achieved the highest AUC-ROC score in Figure 7.3. These findings underscore the nuanced differences between graph-level and node-level evaluation metrics.

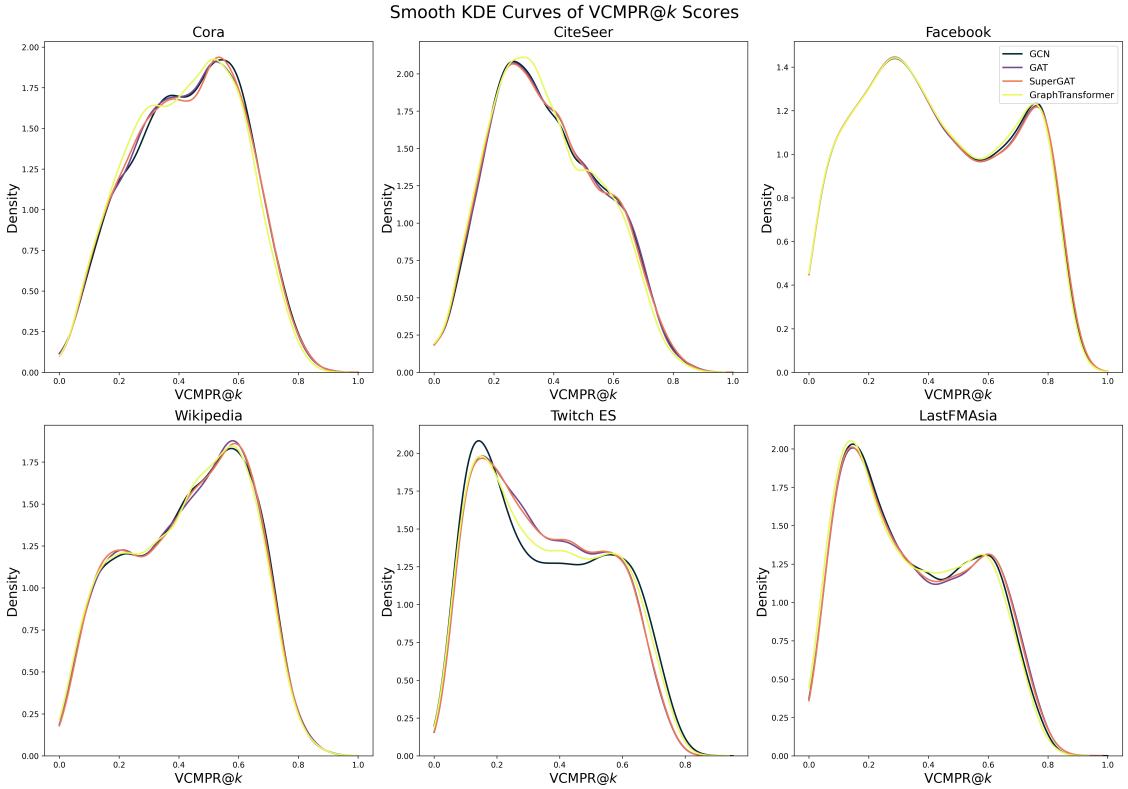

Figure 7.4: Distribution of VCMPR@k values across multiple LP models on various graph datasets. Kernel Density Estimation (KDE) curves reveal the performance variability and distributional characteristics of different graph neural network approaches.

7.3 Results

In this section, we present the results and visualizations that serve as an initial step toward uncovering the critical relationships between LP models and diffusion processes, as outlined previously on this chapter. We begin by examining the simple contagion scenario, followed by an exploration of the complex contagion process.

7.3.1 Simple Contagion

Graph - Level Analysis

First, we examine how the contagion dynamics — specifically the parameters *Iterations*, *Infection Size*, and *Infection Rate* — vary across the four different GNN models discussed in Chapter 3 in the simple contagion context.

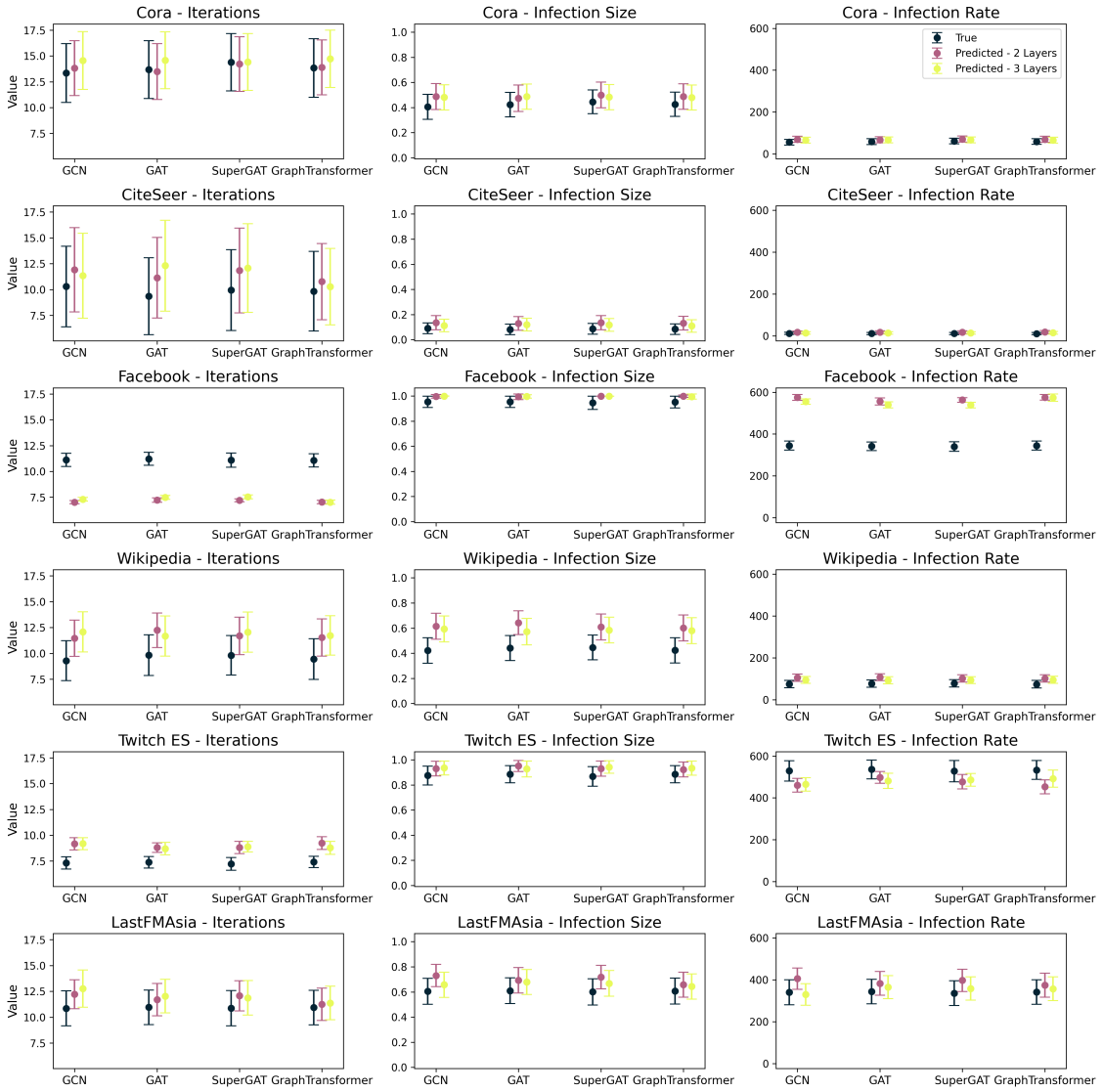

Figure 7.5: Comparison of contagion metrics — *Iterations*, *Infection Size*, and *Infection Rate* — for real and predicted networks across different two-layer and three-layer models. The contagion setup assumes a simple contagion process with $\beta = 0.5$. Metrics are averaged over ten model versions, each evaluated with 100 simulations (as described in Section 6.2.1). Error bars represent the standard error. Results indicate that predicted networks consistently facilitate contagion more effectively than real networks.

The results depicted in Figure 7.5 highlight key distinctions between the contagion dynamics in real and predicted networks. Predicted networks generally exhibit higher values for all contagion metrics — *Iterations*, *Infection Size*, and *Infection Rate* — compared to their real counterparts across all datasets, with the exception of the *Facebook* dataset for the *Iterations*

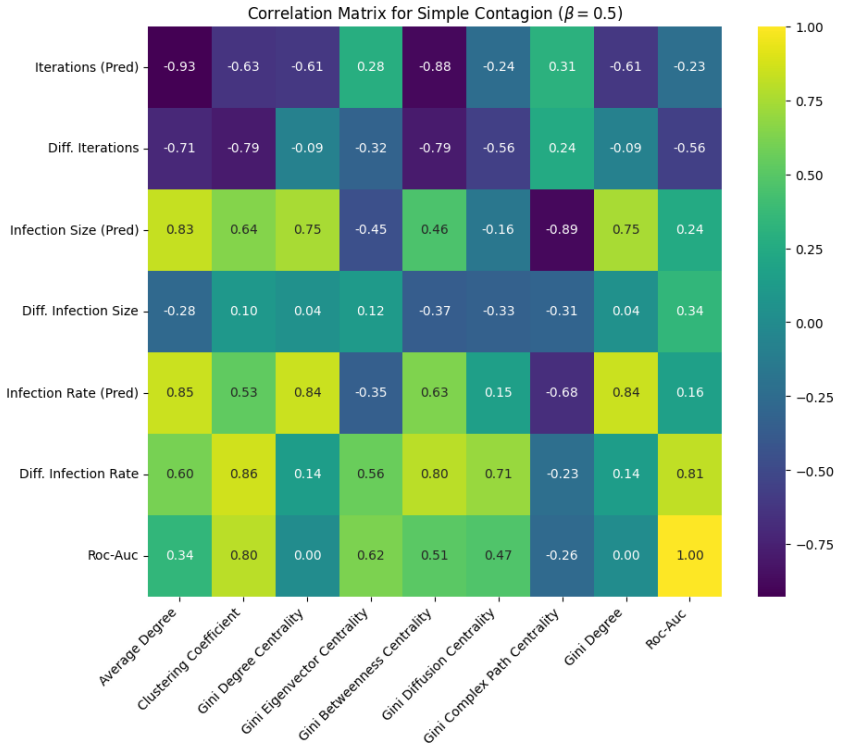

Figure 7.6: Correlation matrix between graph-level topological measures and contagion metrics for simple contagion processes ($\beta = 0.5$). The matrix illustrates relationships between graph-level properties and key contagion dynamics metrics (iterations, infection size, and infection rate), including the ROC-AUC score.

metric, and the *Twitch ES* dataset for *Infection Size*. Notably, *Facebook*, which has the highest average degree and clustering coefficient (see Figure 7.2), displays a different behavior, potentially due to the influence of its denser structure on diffusion dynamics. Within less than ten iterations, the spread of information rapidly stabilizes, effectively reaching the entire population. This trend suggests that LP algorithms tend to generate network structures that facilitate simple information diffusion, likely by introducing shortcuts that reduce the average path length [25]. Additionally, in this figure, the contagion metrics exhibit minimal variation across different models and model depths, suggesting that the structural changes introduced by various LP algorithms lead to largely consistent diffusion patterns.

To understand the distinct behavior of the *Facebook* and *Twitch ES* networks we analyzed the Pearson correlation matrix between the graph topological measures (described in Section 6.1.2) and the contagion metrics. *

*We calculated the differences between predicted and true networks' contagion metrics as: diff = pred − true.

Figure 7.6 reveals several significant relationships. First, the difference in stabilization iterations between predicted and real networks (*Diff. Iterations*) shows strong negative correlations with *Average Degree*, *Clustering Coefficient*, and *Gini Betweenness Centrality*. This explains why the *Facebook* dataset, which exhibits the highest values for these metrics (Figure 7.2), demonstrates inverse behavior to other datasets (Figure 7.5).

Second, *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. This finding is exemplified by the *CiteSeer* network, which has the highest *Gini Complex Path Centrality* and, consequently, the lowest *Infection Size*.

Finally, we observe that *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* network exemplifies this phenomenon, with its high average degree and low clustering coefficient driving an increased *Infection Rate* due to the presence of such structural configurations.

To better understand how models influence the differences in diffusion metrics between predicted and real networks, we present a visualization of these relationships in Figure 7.7. The figure reveals that different models lead to varying discrepancies in social contagion outcomes across networks. For instance, networks with a larger *Average Degree*, such as *Facebook* and *Twitch ES* (see Table 6.1), exhibit smaller variations between models in diffusion differences compared to smaller networks.

Interestingly, a higher *Gini Degree* correlates with a more pronounced effect of LP models on the resulting *Infection Size* and *Infection Rate*. This observation implies that datasets characterized by an average degree driven by a few highly connected nodes — *e.g.*, the *Twitch ES* dataset — are more susceptible to the impact of LP models. The mechanism behind this relationship likely stems from how LP algorithms, which often create shortcuts, can significantly influence diffusion when the prediction involves these highly connected nodes. Such shortcuts could alter the network's structure in ways that amplify or diminish contagion processes, particularly when they create new connections to or between hub nodes. Additionally, we observe a strong negative correlation between *Diff. Iterations* and network features such as *Average Degree* and *Clustering Coefficient*, reinforcing earlier findings about the relationship between network density and diffusion speed.

However, while these structural patterns are evident, the reasons why certain LP models

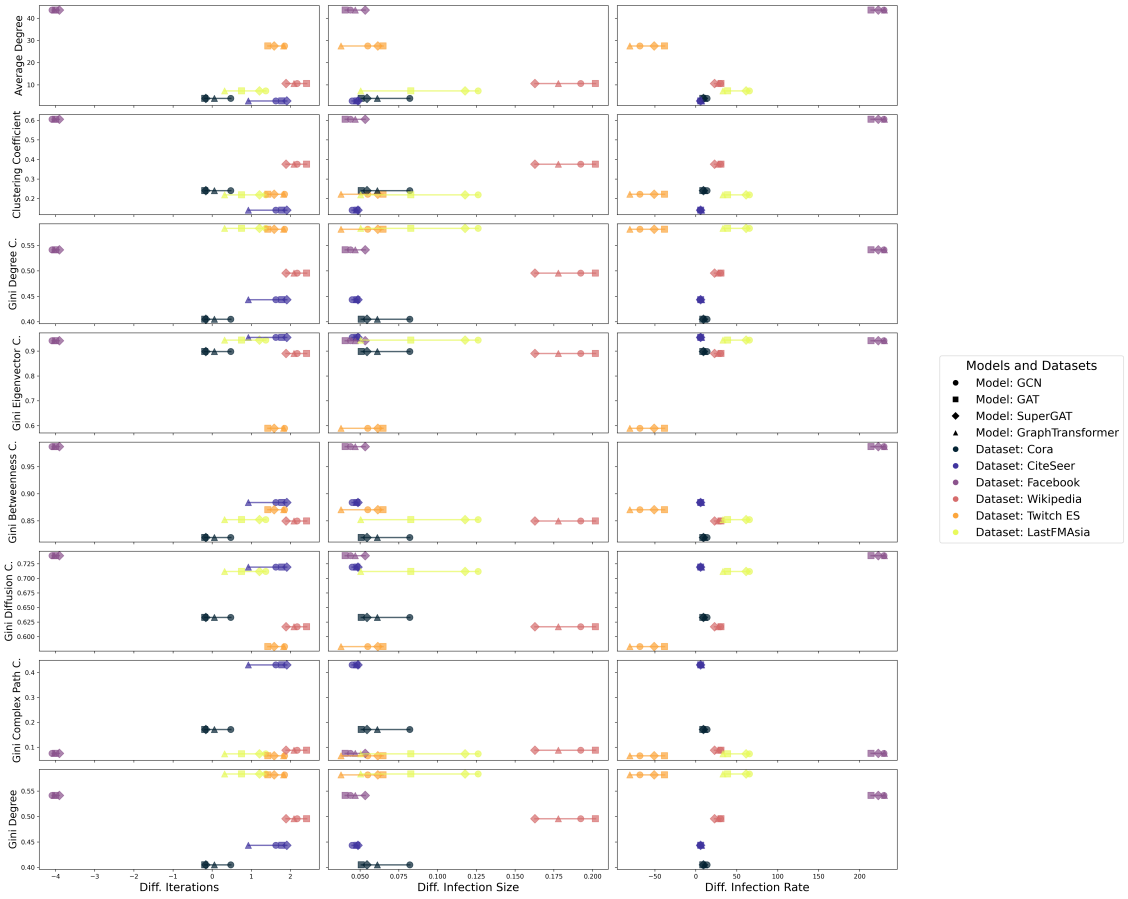

Figure 7.7: Graph topological metrics versus differences in diffusion metrics between predicted and true networks for simple contagion processes ($\beta = 0.5$) in two-layer models. Each dataset is represented by a distinct color and each model by a different marker.

increase specific diffusion metrics compared to others remain less clear. The complexity of these relationships suggests that model performance might depend on how well they preserve or modify critical network features that influence diffusion dynamics.

To explore this even deeper, we analyze how some of the correlations from Figure 7.6 vary among contagion probabilities and models, seeking to uncover whether these relationships are consistent across different diffusion scenarios or if they exhibit model-specific characteristics. Figure 7.8 shows how the correlations between the diffusion metrics and the dataset topological features vary across the contagion probability β. Notably, we observe some possible linear evolutions in the correlations with the *Average Degree* and the *Clustering Coefficient*. These patterns suggest a strong dependency of the correlation values on the contagion probability. These correlations are computed using aggregated data from all models.

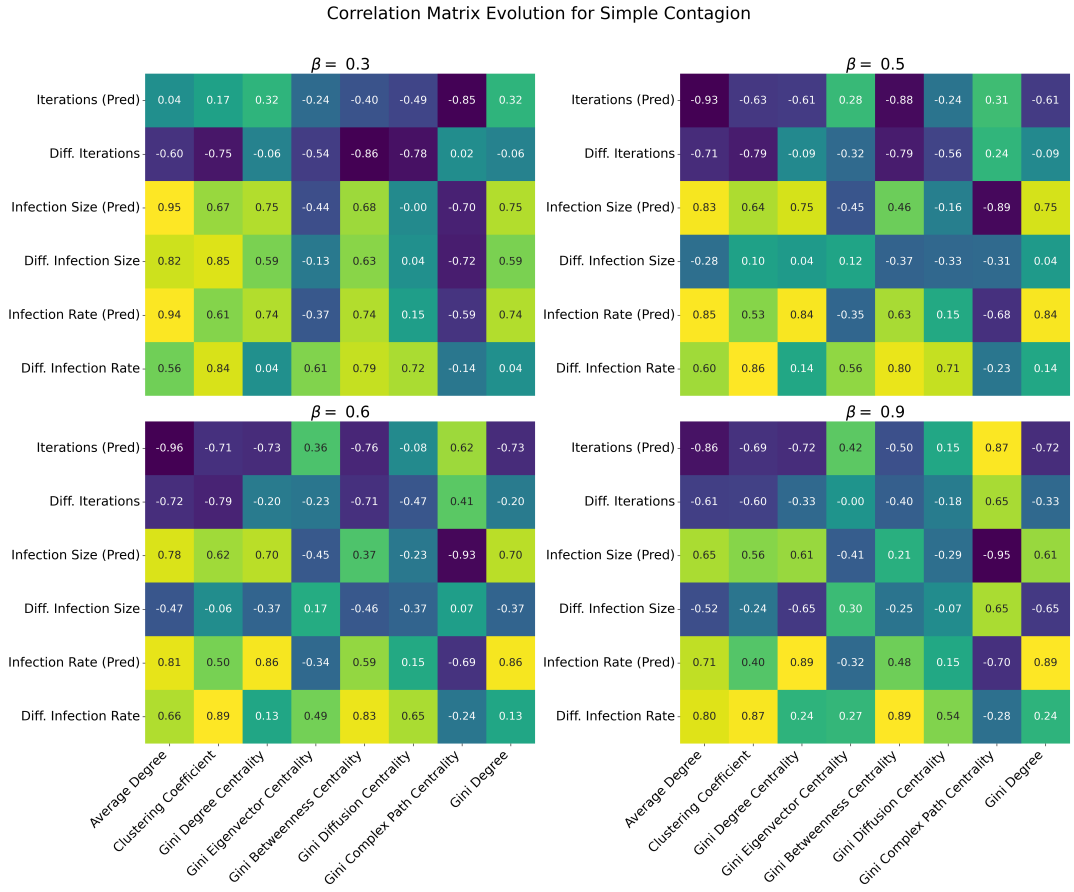

Figure 7.8: Correlation matrix between graph-level topological measures and contagion metrics for simple contagion processes on a range of contagion probabilities: $\beta \in \{0.3, 0.5, 0.6, 0.9\}$. The matrix is computed using simulations from all models.

To further dissect these relationships, we examine how the network's structural properties influence these correlation patterns. Figure 7.9 presents a detailed analysis of correlation dynamics across different diffusion metrics and contagion probabilities β, with specific attention to model dependencies. Each subplot focuses on the correlation between a specific diffusion metric and either the *Average Degree* or *Clustering Coefficient*. The models are color-coded to highlight their distinct behavioral patterns, enabling direct comparison of their evolutionary trajectories.

The trends reveal several noteworthy insights into the relationship between diffusion metrics and network topological features across contagion probabilities and models. Correlations with the *Average Degree* display relatively linear trends, particularly for metrics such as *Infection Size (Pred)* and *Infection Rate (Pred)*. However, for correlations with the *Clustering Coefficient*, the

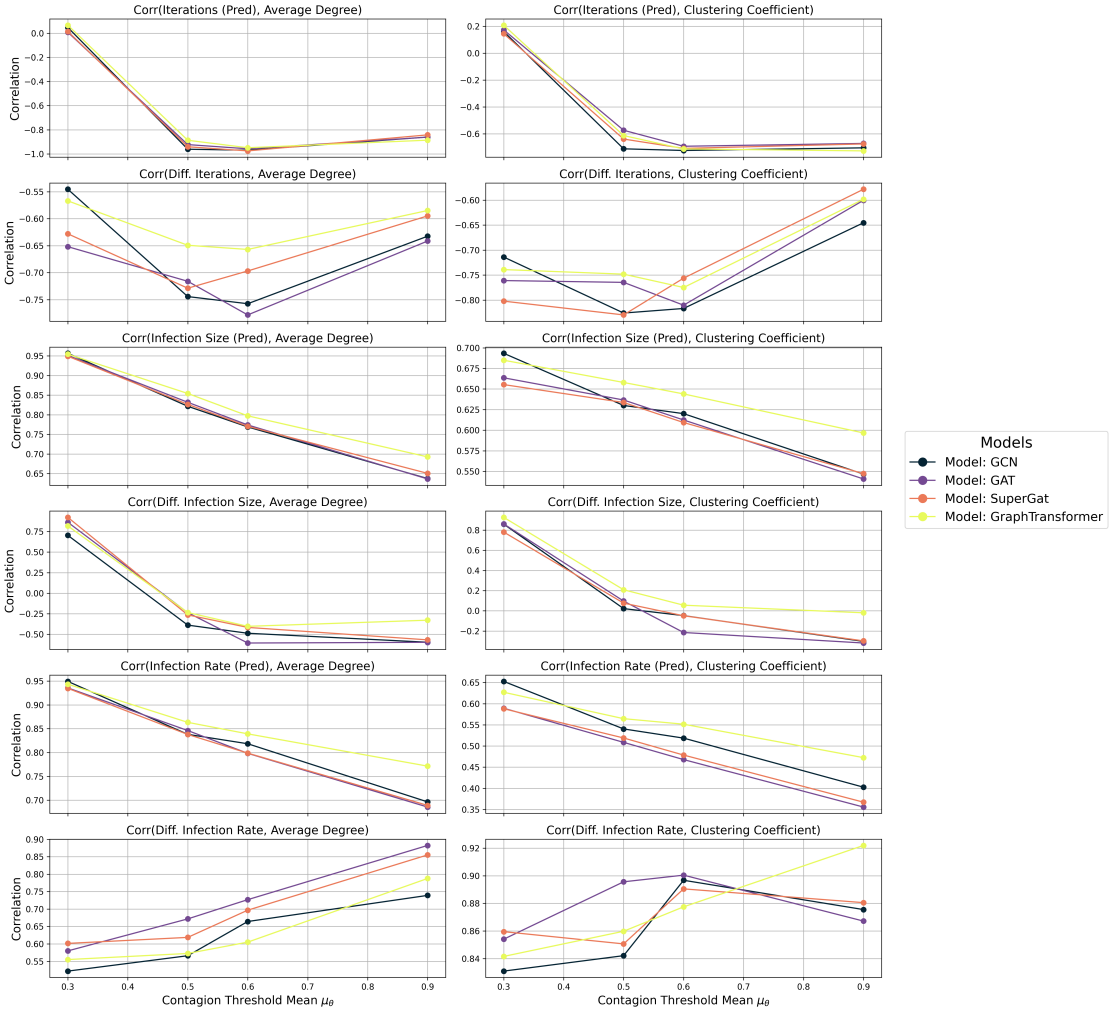

Figure 7.9: Evolution of correlations between diffusion metrics and topological features (*Average Degree*, *Clustering Coefficient*) across contagion probabilities $\beta \in \{0.3, 0.5, 0.6, 0.9\}$ on simple contagion simulations for different models. Lines are color-coded to indicate model types.

behavior across models become more variable, especially for *Diff. Infection Rate*.

Notably, the contagion probability β plays a crucial role in shaping these correlations. As β increases, the correlation between *Iterations (Pred)* and both the *Average Degree* and *Clustering Coefficient* becomes increasingly negative. Meanwhile, the correlation of *Infection Size (Pred)* decreases, and the correlation of *Infection Rate (Pred)* strengthens positively. This highlights the significant dependence of contagion spread on node connectivity.

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 *Infec-*

tion Size and *Infection Rate*. Additionally, a strong negative correlation between topological features and the number of iterations required for contagion stabilization is observed even at relatively low β values. This suggests that these features are critical for accelerating the stabilization of contagion.

Model-specific characteristics also emerge from these trends. Models such as GCN, GAT, and SuperGAT exhibit similar sensitivities to both *Average Degree* and *Clustering Coefficient*, often following comparable trajectories. In contrast, the GraphTransformer demonstrates distinct deviations, leveraging its global attention mechanism. This approach sometimes reduces correlations, particularly for differential diffusion metrics like *Diff. Infection Size* and *Diff. Infection Rate*, while at other times increases correlations, notably for *Infection Size (Pred)* and *Infection Rate (Pred)*. These patterns likely highlight the model's capacity to effectively integrate global structural information.

Finally, a clear distinction emerges between "pred" metrics and their "diff" counterparts. The former generally exhibit smoother and more consistent correlation trends across β, while the latter amplify structural effects ,accentuating the modifications induced by LP on the network and their interaction with contagion probability. This highlights the role of β as a magnifier of structural changes introduced by LP algorithms.

Node-Level Analysis

We now examine how specific node properties correlate with their behavior in the simple diffusion process. This analysis focuses on the relationships between node centrality measures (introduced in Section 6.1.1) and two key diffusion metrics: *Vulnerability* and *Recency* (detailed in Section 6.2.1). We investigate how these relationships are modulated by both the contagion probability β and the various LP algorithms.

Figure 7.10 reveals how nodes participate in the simple contagion process across different datasets and how their roles are affected by LP-generated edges. With $\beta = 0.5$, most datasets exhibit a common pattern: the majority of nodes show high *Vulnerability* and low *Recency*. *CiteSeer* presents a notable exception to this pattern, displaying lower overall susceptibility to contagion. This distinctive behavior aligns with our previous observations of *CiteSeer*'s lower *Complex Path Centrality* and *Degree* distributions, which correlate with reduced *Infection Size* (as shown in Fig. 7.1).

The LP models introduce subtle but significant modifications to these distributions, generally enhancing nodes' susceptibility to contagion. This observation supports our earlier findings regarding the impact of LP algorithms on graph-level diffusion characteristics, as discussed

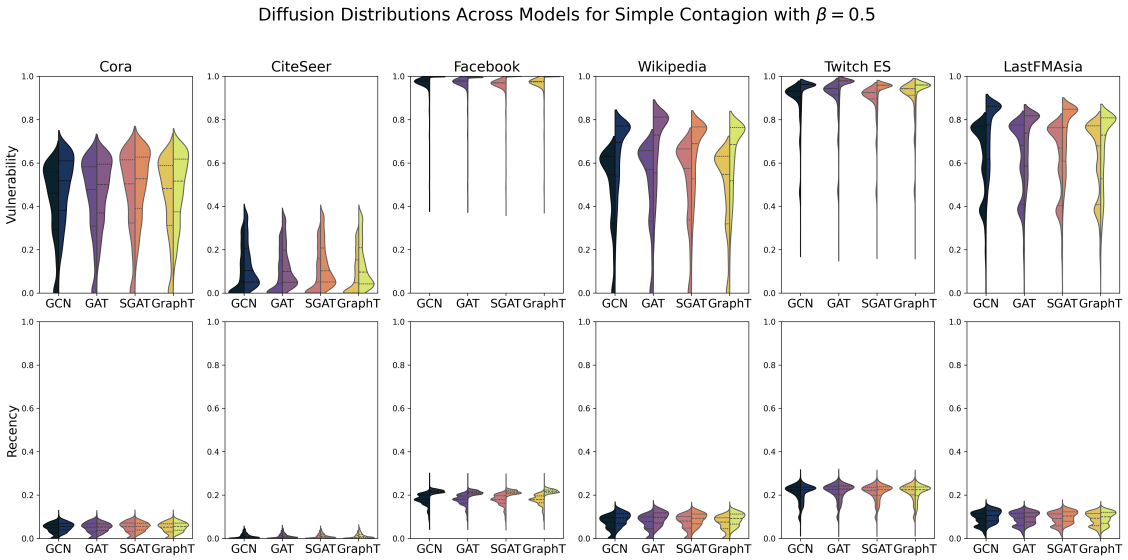

Figure 7.10: Distribution analysis of *Vulnerability* and *Recency* metrics in Simple Contagion processes ($\beta = 0.5$) across different datasets and LP models. Each violin plot is split to show the distribution in true networks (left side) versus predicted networks (right side).

in Section 7.3.1.

Figure 7.11 illustrates the evolution of correlations between node characteristics and contagion metrics across different contagion probabilities β. Unlike the graph-level correlations observed in Figure 7.8, 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.

Extending the analytical framework established in Section 7.3.1, the subsequent analysis delves deeper into how different model architectures modify network parameters that potentially impact contagion spread. Figure 7.12 shows that correlations tend to change monotonically, sometimes accentuated by specific models.

The correlation between *Vulnerability* 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.

However, two centrality measures exhibit unique behavior: *Betweenness Centrality* and *Complex Path Centrality*. The correlation with *Betweenness Centrality* increases slightly for most models, though minimally and potentially attributable to noise. In contrast, *Complex Path*

Figure 7.11: Correlation matrices between node-level centrality measures and contagion metrics for simple contagion processes on a range of contagion probabilities: $\beta \in \{0.3, 0.5, 0.6, 0.9\}$. The matrices is computed using simulations from all models. They also include the correlations with the *VCMPR@k* score.

Centrality's correlation strengthens as the probability increases, stabilizing around $\beta = 0.5$ — an effect particularly pronounced in the Graph Transformer model. This pattern reflects the *Complex Path Centrality* definition, which relates to the average length of complex paths in a node's neighborhood. A node's positional context is crucial in determining infection probability during an epidemic. However, at high contagion probabilities, the node's network position becomes less significant compared to its inherent susceptibility, as the infection can more readily reach remote nodes.

Recency-related correlations demonstrate a similar phenomenon. As contagion probability increases, the number of iterations required for a node to become infected becomes increasingly influenced by its centrality measures, with this correlation plateauing at certain contagion probabilities.

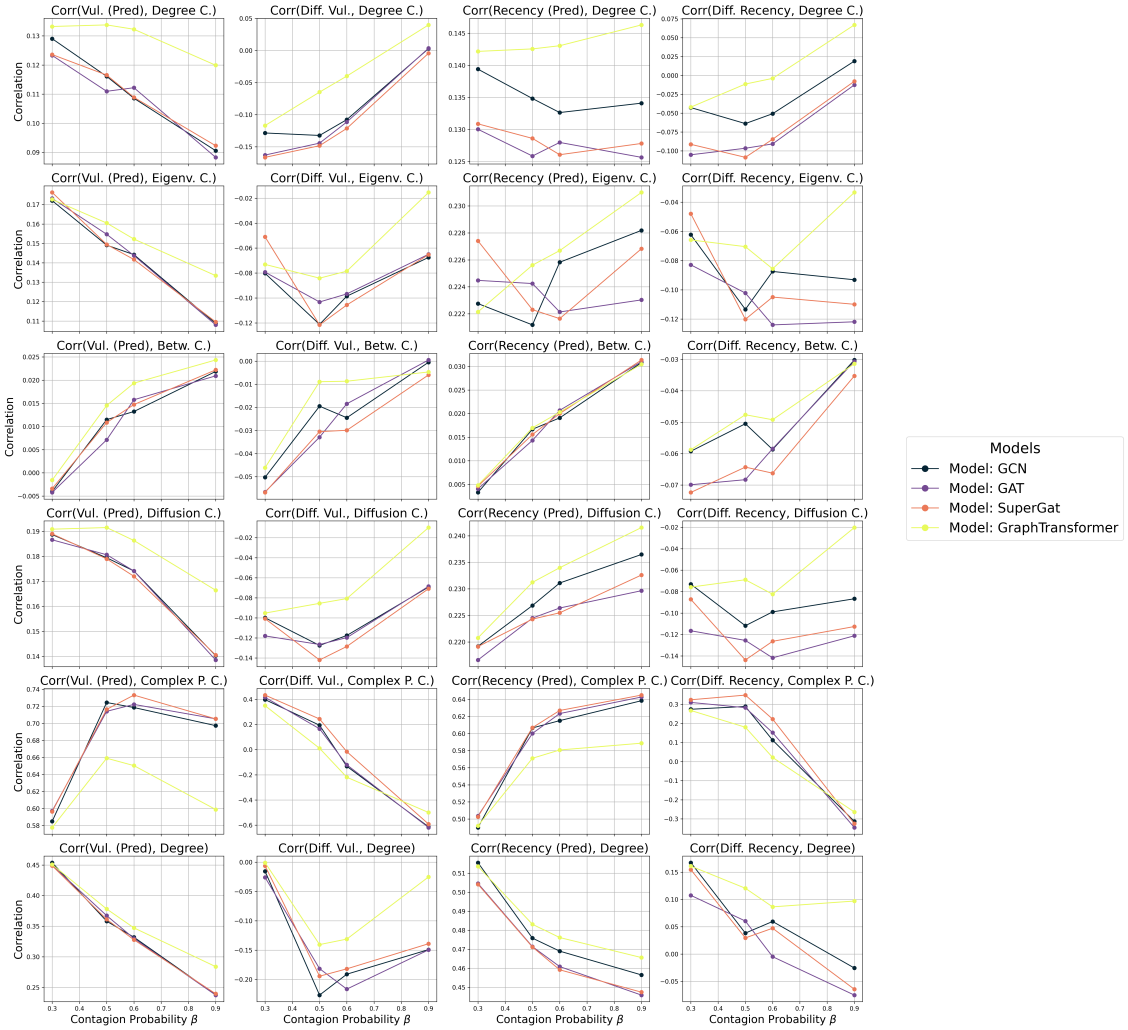

Figure 7.12: Evolution of correlations between node diffusion metrics and centrality features across contagion probabilities $\beta \in \{0.3, 0.5, 0.6, 0.9\}$ on simple contagion simulations for different models. Lines are color-coded to indicate model types.

The different model architectures exhibit distinct effects on the relationships between the predicted and actual network node properties, as well as their correlations with centrality measures. Among these, the Graph Transformer model stands out for its unique behavior. Specifically, it dampens the correlation between the differences in predicted and actual node *Vulnerability* and *Recency*, causing these correlations to approach zero at lower contagion probability values compared to other models. Furthermore, the Graph Transformer consistently exhibits correlation values that differ from those of other models, which often display similar trends. For instance, in its correlation with *Complex Path Centrality*, networks generated from edges

predicted by the Graph Transformer maintain a lower correlation even at high contagion probabilities, distinguishing its behavior from that of the other architectures.

7.3.2 Complex Contagion

Now, we present the results of the complex contagion simulations. As previously explained, in the complex contagion model, a susceptible node becomes infected only when a sufficient number of its neighbors are already infected. Node thresholds are drawn from a truncated normal distribution, with the mean threshold μ_θ taking values in the set $\{0.2, 0.3, 0.4, 0.6\}$, and the standard error set to $\sigma_\theta = 0.2$. This threshold-based mechanism introduces a more nuanced contagion process compared to the simple model, where adoption requires social reinforcement from multiple sources. In this context, individuals need substantial social proof or validation — such as adopting new behaviors, beliefs, or technologies — before changing their state. This captures scenarios where a single influence is not enough to induce change, and a critical mass of infected neighbors is necessary for a node to adopt the infection.

Graph-Level Analysis

As with the simple contagion analysis, we begin by examining how contagion dynamics vary across the four GNN models discussed in Chapter 3.

The results, shown in Figure 7.13, reveal significant differences between the contagion dynamics in real and predicted networks. Overall, the trends are similar to those observed in the simple contagion model. In both contagion scenarios, a consistent pattern emerges: LP algorithms often produce network structures that facilitate more efficient information diffusion. Notably, as seen with simple contagion, the *Facebook* dataset exhibits a unique behavior. In the predicted network, all nodes are reached — *i.e.*, *Infection Size* $= 1$ — in fewer iterations compared to the real network, which fails to reach all individuals. This difference stems from the increased contagion rate in the LP-affected network, a phenomenon also observed in the simple contagion analysis (Figure 7.5).

Figure 7.14 depicts the the Pearson correlation matrix between the graph topological measures and the contagion metrics for $\mu_\theta = 0.2$. Similar trends to those observed in simple contagion emerge: the disparity in stabilization iterations between predicted and real networks (*Diff. Iterations*) exhibits strong negative correlations with *Average Degree*, *Clustering Coefficient*, and *Gini Betweenness Centrality*. Meanwhile, *Infection Size* correlates positively with *Average Degree* and *Clustering Coefficient* but negatively with *Gini Complex Path Centrality*.

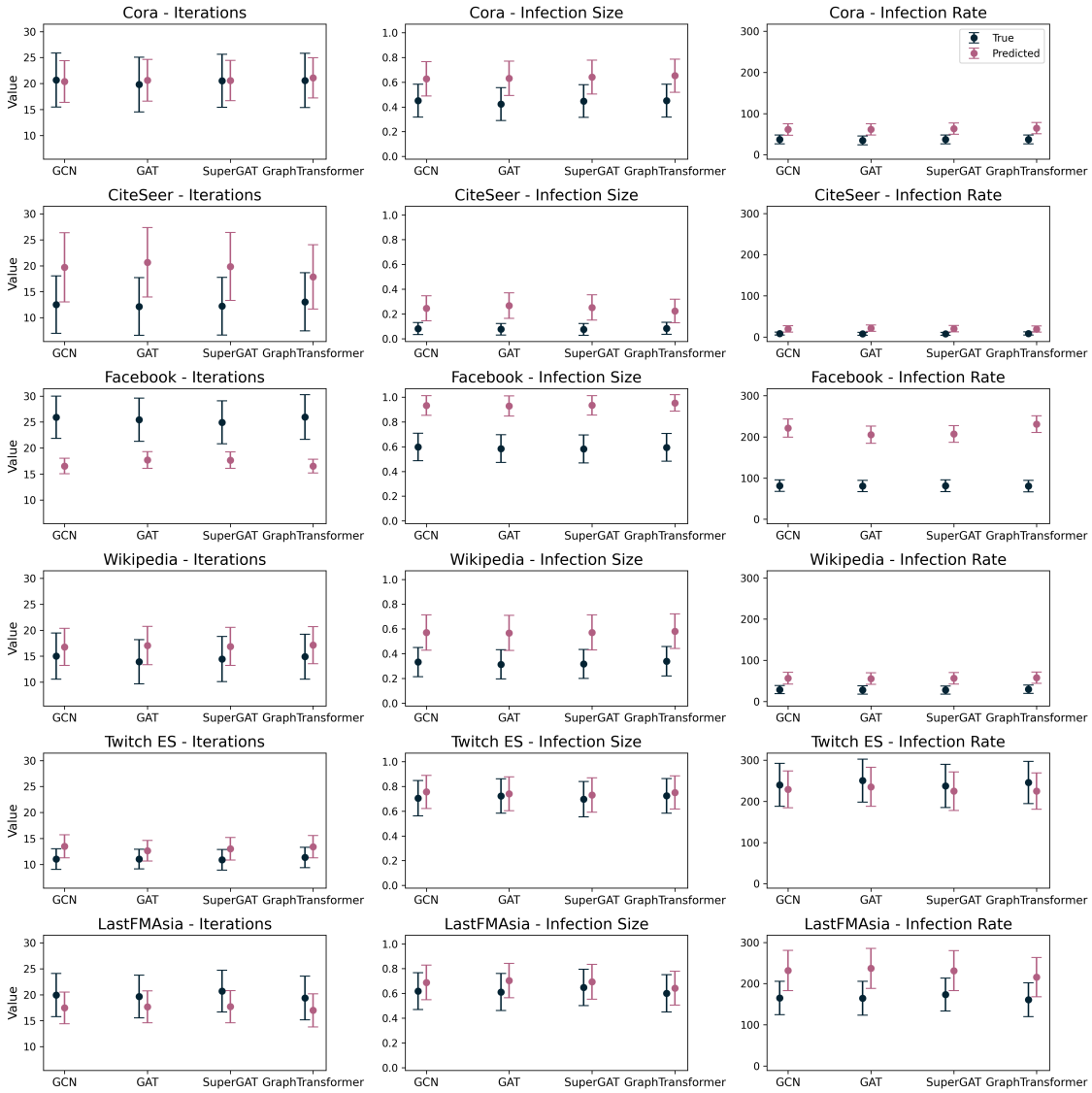

Figure 7.13: Comparison of contagion metrics — Iterations, Infection Size, and Infection Rate — for real and predicted networks across different two-layer models. The contagion setup assumes a complex contagion process with $\mu_\theta = 0.2$ and $\sigma_\theta = 0.2$ Metrics are averaged over ten model versions, each evaluated with 100 simulations (as described in Section 6.2.1). Error bars represent the standard error.

The presence of shortcuts reshapes the network's structure, potentially enhancing or hindering contagion, particularly by linking or reinforcing hub nodes. Lastly, *Infection Rate* shows a positive correlation with both *Average Degree*, *Gini Degree Centrality* and *Gini Degree*.

Using the same framework as the simple contagion analysis, we now examine how the correlations from Figure 7.14 vary across threshold means and models.

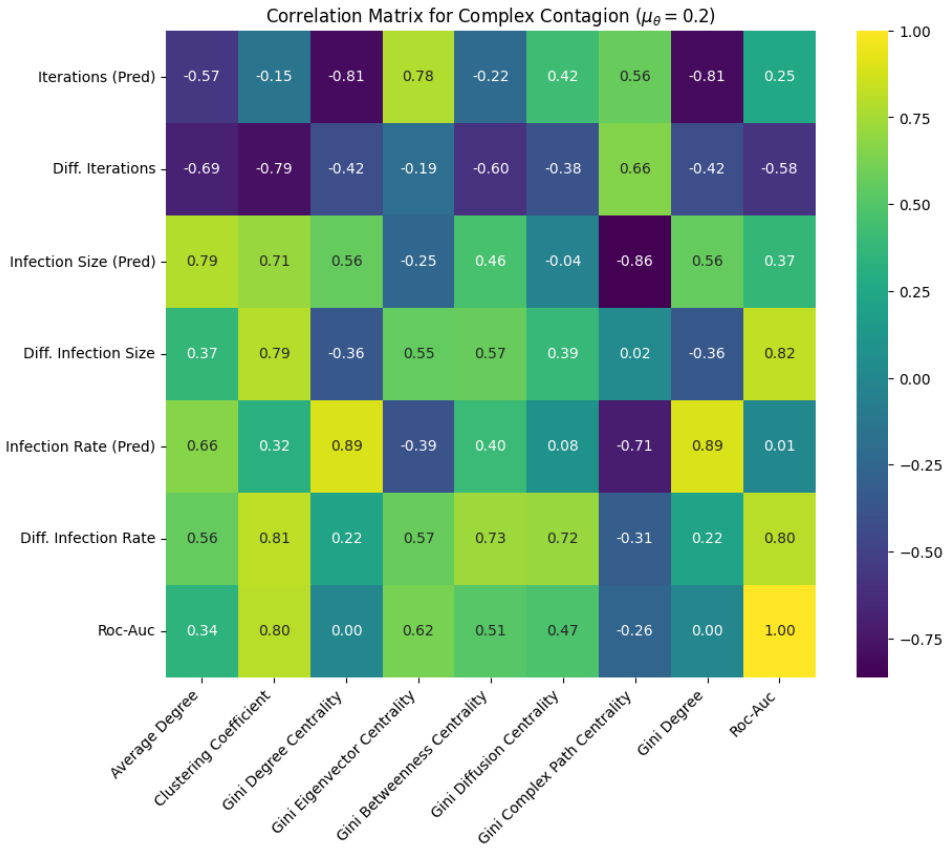

Figure 7.14: Correlation matrix between graph-level topological measures and contagion metrics for complex contagion processes ($\mu_\theta = 0.2$). The matrices are computed using simulations from all models. The matrix illustrates relationships between graph-level properties (average degree, clustering coefficient, and Gini indices) and key contagion dynamics metrics (iterations, infection size, and infection rate), including the ROC-AUC score.

Figure 7.15 illustrates how, in the complex contagion process, the correlation between contagion metrics (*Iterations(Pred)*, *Infection Size(Pred)*, *Infection Rate(Pred)*) and network connectivity features (*Average Degree*, *Clustering Coefficient*) strengthens as the contagion threshold mean (μ_θ) increases. These correlations rise sharply, approaching near-perfect positive values (*i.e.*, close to 1) by $\mu_\theta = 0.3$, which is relatively low.

This pattern contrasts with that observed in the simple contagion process (see Figure 7.9).The difference lies in the opposing roles of contagion probability (β) and contagion threshold (μ_θ) in determining the ease of propagation: while higher β values facilitate contagion, higher μ_θ values make propagation more difficult. For high μ_θ values, nodes require most of their neighbors to be infected before becoming infected, making network connectivity — specifically the number of neighbors (*Average Degree*) and clustering tendency (*Clustering Coefficient*) — crit-

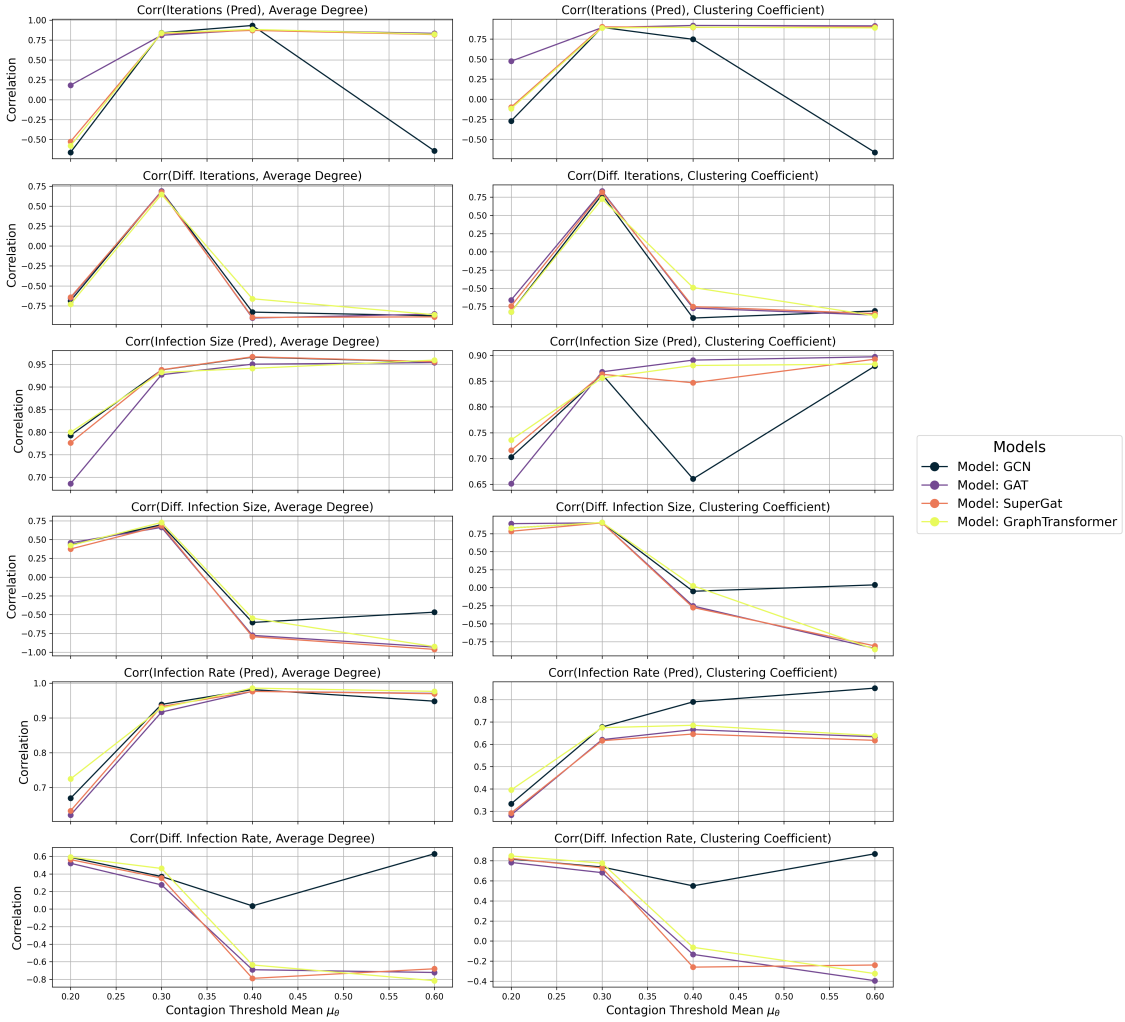

Figure 7.15: Evolution of correlations between diffusion metrics and topological features (*Average Degree*, *Clustering Coefficient*) across threshold means $\mu_\theta \in \{0.2, 0.3, 0.4, 0.6\}$ on complex contagion simulations for different models. Lines are color-coded to indicate model types.

ical for social diffusion. Conversely, at low μ_θ values, contagion spreads more easily, reducing the importance of neighborhood structure, local clustering, or the number of neighbors in determining infection likelihood.

In analyzing the differences between models, a striking observation emerges for high μ_θ: the correlations associated with the GCN model begin to deviate significantly from those of the other models. Specifically, for the correlation between *Iterations(Pred)* and network connectivity metrics, the GCN network exhibits a strong negative correlation at $\mu_\theta = 0.6$, whereas the other models demonstrate strong positive correlations. This means that higher connectivity in

GCN-predicted networks might accelerate stabilization, and the opposite in the rest of models. This divergence suggests that the structural changes induced by the GCN model under high contagion thresholds might hinder the diffusion process, potentially due to an over-reliance on highly connected nodes or the formation of bottlenecks that limit propagation efficiency. This could occur because densely connected clusters quickly reach a state where all susceptible nodes are infected, or the remaining uninfected nodes cannot be reached due to the high contagion thresholds. The GCN's emphasis on local smoothing may create highly cohesive substructures, which either facilitate complete saturation within clusters or isolate nodes effectively, reducing the number of iterations required for stabilization.

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 contagion to spread more widely before stabilization. These differences suggest that the GCN's predictions may emphasize localized spread within tightly-knit clusters, whereas other models generate networks that support broader, slower propagation.

A similar anomaly is observed in the correlation involving *Diff. Infection Rate*. While the GCN-predicted network maintains a strong positive correlation across all contagion thresholds, the other models experience a sharp decline in correlation strength starting at $\mu_\theta = 0.3$. This consistent strong positive correlation in GCN across all thresholds could reflect its structural bias in generating denser networks or preserving high-degree nodes. These features likely boost the predicted infection rate compared to the real network. Conversely, the declining correlations in other models starting at $\mu_\theta = 0.3$ might suggest that their predictions diverge from the real network's topology in ways that lower infection rates under stringent threshold conditions. They could be incorporating global topological adjustments that inadvertently reduce the prominence of hubs or create shortcuts that bypass high-degree nodes, thereby diminishing infection rates relative to the real network.

For small contagion thresholds, the GAT model also stands out, as it reduces the strength of correlations involving *Iterations(Pred)* and *Infection Size(Pred)*. This attenuation might reflect the GAT model's emphasis on selective attention to specific node features, which could alter the balance of local versus global connectivity in ways that suppress the impact of network structure on contagion processes. In contrast, the SuperGAT and GraphTransformer models display highly similar correlation trends across all features.

These observations point to potential trade-offs in model design: GCN may prioritize denser or more clustered predictions that accelerate contagion stabilization and enhance infection

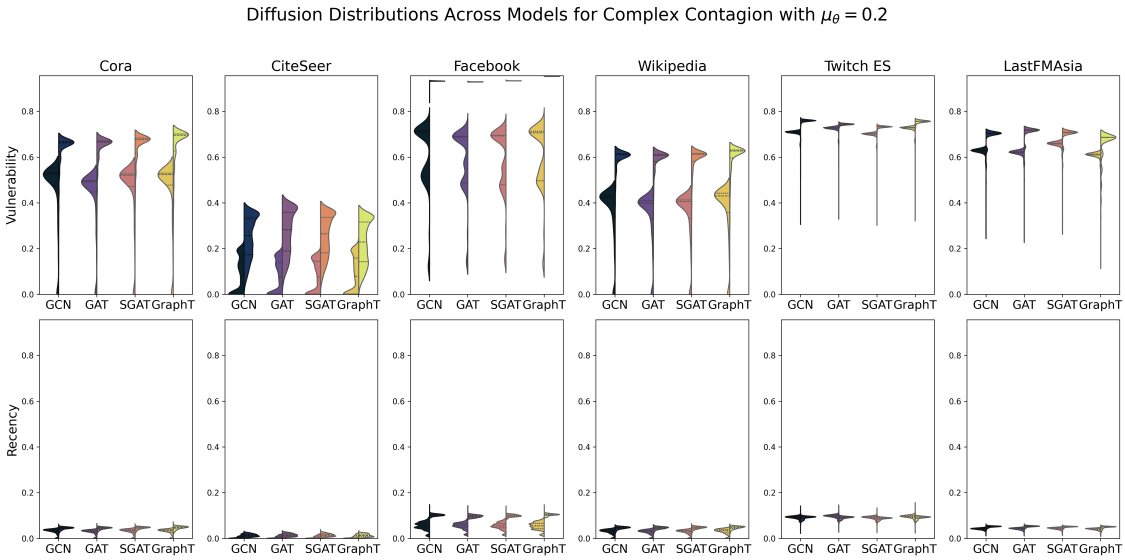

Figure 7.16: Distribution analysis of *Vulnerability* and *Recency* metrics in complex contagion processes ($\mu_\theta = 0.2$) across different datasets and LP models. Each violin plot is split to show the distribution in true networks (left side) versus predicted networks (right side).

rates, while other models might better approximate real-world dynamics by producing networks less prone to exaggerated diffusion effects.

Node-Level Analysis

Finally, we delve into analyze the relationship between node properties and complex contagion progression.

Figure 7.16 shows the participation of nodes in complex contagion across diverse datasets, revealing how LP-generated edges reshape node vulnerabilities. At $\mu_\theta = 0.2$, most networks display a consistent pattern of high *Vulnerability* and low *Recency*, with *CiteSeer* emerging as a distinct outlier. This distinctive behavior, consistently observed across both simple and complex contagion models (Figure 7.10), reflects its unique network structure characterized by low *Complex Path Centrality* and *Degree*.

Furthermore, the predicted networks consistently demonstrate elevated node *Vulnerability* compared to their real counterparts. In *Facebook*, for instance, LP-affected networks achieve near-total node reach, contrasting with the real network's more limited spread of $60 - 80\%$. These subtle yet significant LP-induced modifications underscore the algorithms' profound impact on network diffusion dynamics, extending our previous observations from simple to complex contagion models.

Figure 7.17: Correlation matrices between node-level centrality measures and contagion metrics for complex contagion processes on a range of threshold means $\mu_\theta \in \{0.2, 0.3, 0.4, 0.6\}$. The matrices are computed using simulations from all models. They also include the correlations with the VCMPR@k score.

Figure 7.17 presents the evolution of correlations between node characteristics and contagion metrics across varying contagion thresholds. In contrast to the graph-level correlations shown in Figure 7.14, these node-level correlations are significantly weaker, reflecting more localized influences. At lower contagion thresholds, *Complex Path Centrality* and *Degree* emerge as the most influential factors in determining a node's susceptibility to contagion. At higher contagion thresholds, *Degree*, *Degree Centrality*, and *Diffusion Centrality* become the dominant predictors, underscoring the shifting dynamics of node influence as the contagion process intensifies.

Figure 7.18 illustrates how different LP models influence these correlation dependencies. The correlation between a node's *Vulnerability* and *Recency* with most centrality measures

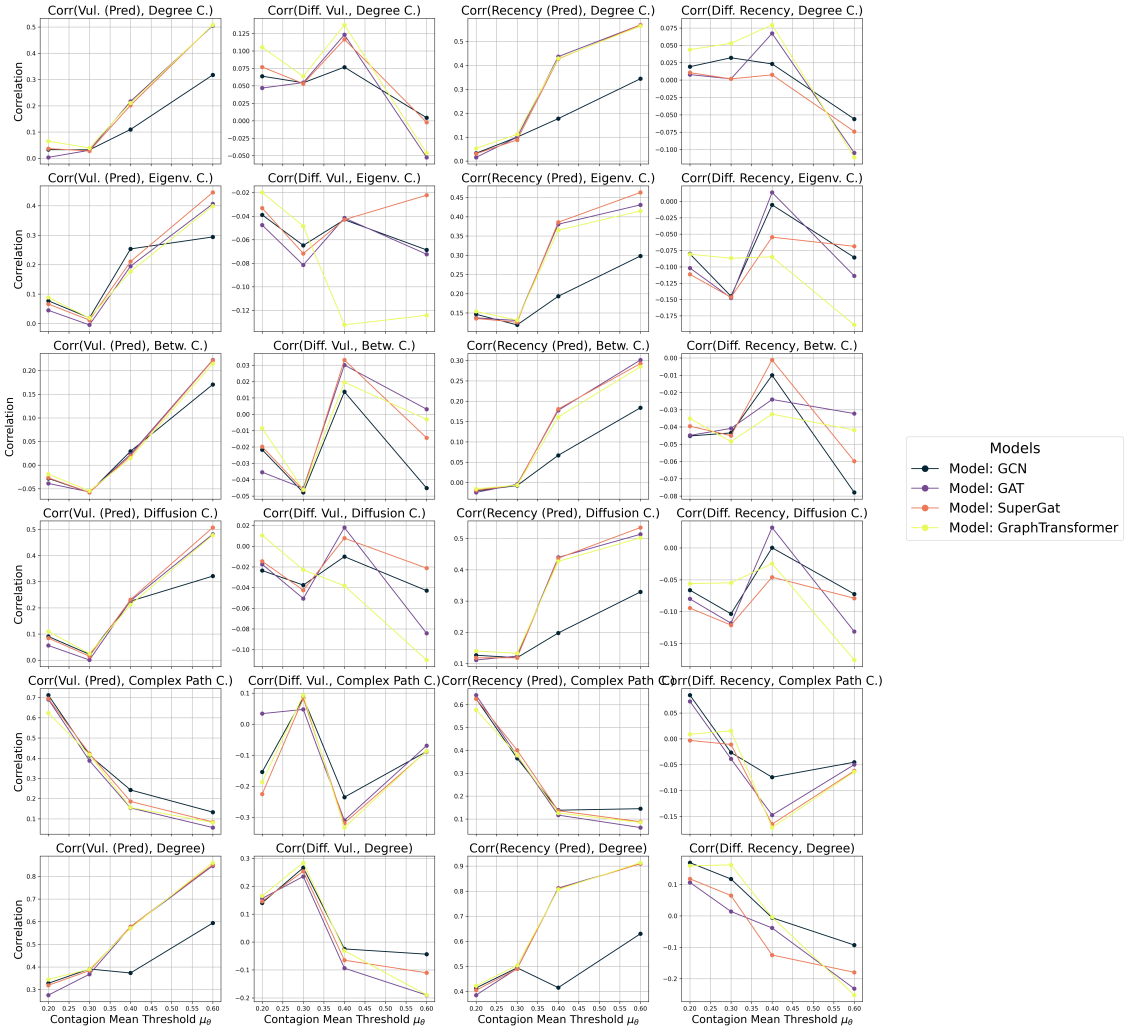

Figure 7.18: Evolution of correlations between node diffusion metrics and centrality features on a range of threshold means $\mu_\theta \in \{0.2, 0.3, 0.4, 0.6\}$ on complex contagion simulations for different models. Lines are color-coded to indicate model types.

weakens as the complex contagion threshold μ_θ decreases. This trend is consistent with the dynamics observed in the simple contagion scenario, where higher probabilities of contagion lead to a greater susceptibility of nodes to infection. Consequently, the impact of neighborhood characteristics, connection quality, or the number of neighbors on the likelihood of infection diminishes.

Interestingly, as with simple contagion, the *Complex Path Centrality* emerges as an exception to this pattern. It maintains a stronger correlation for small μ_θ, suggesting it captures unique structural features that remain relevant at high-contagion regimes. This highlights its potential

as a robust centrality measure in diverse contagion processes.

We further focus on how different model architectures uniquely affect the relationships between predicted and actual node properties, as well as their correlations with centrality measures. The GCN model stands out as the most distinct, particularly at high contagion thresholds, a trend already highlighted in the graph-level analysis. At the node level, the GCN-predicted network demonstrates consistently lower correlations between *Vulnerability* and all centrality measures, as well as between *Recency* and all centrality measures, with the exception of *Complex Path Centrality*. This behavior supports the hypothesis proposed earlier: the structural changes induced by the GCN model under high contagion thresholds may disrupt the social contagion process. This could stem from an over-reliance on highly connected nodes or the creation of structural bottlenecks that limit propagation efficiency and suppress the influence of broader network characteristics.

In contrast, other models exhibit remarkably similar correlation trends, indicating that their attention mechanisms likely capture network features in comparable ways. This uniformity suggests a convergence in their modeling approaches, which may be effective in low-contagion scenarios but less capable of differentiating structural dynamics at higher thresholds.

8 Conclusion

In the rapidly evolving fields of network science and computational sociology, understanding the dynamics of social contagion is critical. This research introduced a novel approach to studying how link-prediction (LP) models, particularly those utilizing Graph Neural Networks (GNNs), reshape network topologies. By suggesting or removing connections, these models fundamentally alter propagation pathways, influencing both simple and complex contagion processes. 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.

The graph-level analysis provided insights into global structural properties and their implications for diffusion metrics, while the node-level study revealed the roles of individual nodes in propagation. This dual perspective, combined with our focus on both simple and complex contagion scenarios, offers a comprehensive understanding of information and behavior spread. Moreover, our detailed characterization of networks at these levels underscores the intricate interplay between topology, centrality and diffusion.

The research employed state-of-the-art Graph Neural Network architectures — including Graph Convolutional Networks (GCN), Graph Attention Networks (GAT), SuperGAT, and Graph Transformers — to predict network structures. By rigorously comparing these models' performance and their impact on social diffusion parameters, we contributed to the emerging field of network reconstruction and predictive modeling. Our comprehensive approach meticulously evaluated the technical capabilities of GNN-based LP models through a multi-

faceted assessment strategy. We employed both classical performance metrics like AUC-ROC and innovative evaluation techniques, such as the Vertex-Centric Max Precision Recall at k (VCMPR@k), to provide a nuanced analysis of model performance. Beyond technical analysis, we explored the broader implications of LP on the dynamics of social contagion.

From over 100,000 SI simulations, we derived several critical findings:

1. Simple Contagion Dynamics

    - LP models consistently introduced structural shortcuts, reducing average path lengths and enhancing diffusion efficiency. This phenomenon confirmed prior findings by Centola et al. [25].
    - Contagion metrics exhibited robustness across models and network depths, highlighting consistent diffusion patterns.
    - 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 and heightened LP impact on *Infection Rate*.
    - The contagion probability β modulated infection dynamics: higher probabilities diminished the influence of network topology by making nodes uniformly susceptible, reducing the importance of neighborhood characteristics.
    - Graph Transformers, leveraging global attention, provided smoother, more stable diffusion patterns. GCN, GAT and SuperGAT present similar behaviors.

2. Complex Contagion Dynamics

    - 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.

Our findings emphasize the importance of considering both graph- and node-level characteristics. Measures like *Complex Path Centrality* and node degree emerged as pivotal in determining contagion susceptibility, with *Complex Path Centrality* distinguishing itself as uniquely insightful for capturing the relationship between network topology and contagion behavior.

This work advances our understanding of how LP models influence contagion dynamics, demonstrating the potential of machine learning techniques to reveal the mechanisms underlying social behavior and information diffusion. However, this study has some limitations. First, our experiments are conducted on static networks, which may not fully capture the temporal dynamics of real-world social networks. Additionally, while we analyzed diverse datasets, further research could include larger and more heterogeneous networks, such as those from communication, transportation, or biological systems, to evaluate the generalizability of the findings. Finally, although GNN-based LP models were the focus, incorporating comparisons with simpler or hybrid algorithms could provide a more holistic understanding of LP's impact on diffusion processes.

Future research could explore real-world applications, such as optimizing network interventions or mitigating misinformation spread, by tailoring LP models to enhance beneficial contagion while suppressing harmful diffusion. Investigating the role of temporal and dynamic networks could yield further insights into the evolving interplay between LP algorithms and network behavior. Expanding to multiplex networks, where nodes have multiple types of connections, would offer a richer framework to study layered contagion dynamics. Furthermore, developing interpretability techniques for GNN-based LP models could also throw light on the causal mechanisms driving observed effects, paving the way for more transparent and actionable insights.

References

[1] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, p. 4–24, Jan. 2021. [Online]. Available: http://dx.doi.org/10.1109/TNNLS.2020.2978386

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

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

[4] Statia Research Department. Time spent monthly on leading website categories in italy 2024. [Online]. Available: https://www.statista.com/statistics/594259/italy-monthly-time-spent-on-top-15-website-categories/

[5] 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

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

[7] G. D. S. Martino, S. Cresci, A. Barron-Cedeno, S. Yu, R. D. Pietro, and P. Nakov, "A survey on computational propaganda detection," 2020. [Online]. Available: https://arxiv.org/abs/2007.08024

[8] 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.

[9] 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

[10] 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

[11] M. Jackson, "A typology of social capital and associated network measures," *Social Choice and Welfare*, vol. 54, pp. 311–336, 2020. [Online]. Available: https://doi.org/10.1007/s00355-019-01189-3

[12] H. Allcott, M. Gentzkow, W. Mason, A. Wilkins, P. Barberá, T. Brown, J. C. Cisneros, A. Crespo-Tenorio, D. Dimmery, D. Freelon, S. González-Bailón, A. M. Guess, Y. M. Kim, D. Lazer, N. Malhotra, D. Moehler, S. Nair-Desai, H. N. E. Barj, B. Nyhan, A. C. P. de Queiroz, J. Pan, J. Settle, E. Thorson, R. Tromble, C. V. Rivera, B. Wittenbrink, M. Wojcieszak, S. Zahedian, A. Franco, C. K. de Jonge, N. J. Stroud, and J. A. Tucker, "The effects of facebook and instagram on the 2020 election: A deactivation experiment," *Proceedings of the National Academy of Sciences*, vol. 121, no. 21, p. e2321584121, 2024. [Online]. Available: https://www.pnas.org/doi/abs/10.1073/pnas.2321584121

[13] B. Gesuele, C. Metallo, M.-D. Guillamón, and A.-M. Ríos, *The Use of Social Media for Electoral Purposes. The Case of the Italian Election in 2018*, 10 2021, pp. 249–263.

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

[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] D. Liben-nowell and J. Kleinberg, "The link prediction problem for social networks," *Journal of the American Society for Information Science and Technology*, vol. 58, 11 2003.

[17] M. Nickel, X. Jiang, and V. Tresp, "Reducing the rank in relational factorization models by including observable patterns," in *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, Eds., vol. 27. Curran Associates, Inc., 2014. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2014/file/556f391937dfd4398cbac35e050a2177-Paper.pdf

[18] H. Zhao, L. Du, and W. Buntine, "Leveraging node attributes for incomplete relational data," 2017. [Online]. Available: https://arxiv.org/abs/1706.04289

[19] V. Martínez, C. Cano, and A. Blanco, "Prophnet: A generic prioritization method through propagation of information," *BMC bioinformatics*, vol. 15 Suppl 1, p. S5, 01 2014.

[20] N. Shibata, Y. Kajikawa, and I. Sakata, "Link prediction in citation networks," *Journal of the American Society for Information Science and Technology*, vol. 63, no. 1, pp. 78–85, 2012. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/asi.21664

[21] J. Sassine and H. Rahmandad, "How does network structure impact socially reinforced diffusion?" *Organization Science*, vol. 35, 02 2023.

[22] G. Cencetti, D. A. Contreras, M. Mancastroppa, and A. Barrat, "Distinguishing simple and complex contagion processes on networks," *Phys. Rev. Lett.*, vol. 130, p. 247401, Jun 2023. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.130.247401

[23] A. Banerjee, A. G. Chandrasekhar, E. Duflo, and M. O. Jackson, "The diffusion of microfinance," *Science*, vol. 341, no. 6144, p. 1236498, 2013. [Online]. Available: https://www.science.org/doi/abs/10.1126/science.1236498

[24] N. Hodas and K. Lerman, "The simple rules of social contagion," *Scientific reports*, vol. 4, p. 4343, 03 2014.

[25] D. Centola and M. Macy, "Complex contagions and the weakness of long ties," *American Journal of Sociology*, vol. 113, no. 3, pp. 702–734, 2007. [Online]. Available: http://www.jstor.org/stable/10.1086/521848

[26] T. N. Kipf and M. Welling, "Variational graph auto-encoders," 2016. [Online]. Available: https://arxiv.org/abs/1611.07308

[27] I. Chami, R. Ying, C. Ré, and J. Leskovec, "Hyperbolic graph convolutional neural networks," *CoRR*, vol. abs/1910.12933, 2019. [Online]. Available: http://arxiv.org/abs/1910.12933

[28] M. Zhang and Y. Chen, "Link prediction based on graph neural networks," 2018. [Online]. Available: https://arxiv.org/abs/1802.09691

[29] J. You, R. Ying, and J. Leskovec, "Position-aware graph neural networks," 2019. [Online]. Available: https://arxiv.org/abs/1906.04817

[30] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 61–80, 2009.

[31] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: online learning of social representations," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, ser. KDD '14. ACM, Aug. 2014, p. 701–710. [Online]. Available: http://dx.doi.org/10.1145/2623330.2623732

[32] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: Large-scale information network embedding," in *Proceedings of the 24th International Conference on World Wide Web*, ser. WWW '15. International World Wide Web Conferences Steering Committee, May 2015, p. 1067–1077. [Online]. Available: http://dx.doi.org/10.1145/2736277.2741093

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

[34] 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

[35] 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

[36] 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

[37] 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

[38]

[39] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: Going beyond euclidean data," *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 18–42, 2017.

[40] P. W. Battaglia, J. B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner, C. Gulcehre, F. Song, A. Ballard, J. Gilmer, G. Dahl, A. Vaswani, K. Allen, C. Nash, V. Langston, C. Dyer, N. Heess, D. Wierstra, P. Kohli, M. Botvinick, O. Vinyals, Y. Li, and R. Pascanu, "Relational inductive biases, deep learning, and graph networks," 2018. [Online]. Available: https://arxiv.org/abs/1806.01261

[41] F. Scarselli, M. Gori, A. Tsoi, M. Hagenbuchner, and G. Monfardini, "Computational capabilities of graph neural networks," *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council*, vol. 20, pp. 81–102, 02 2009.

[42] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," 2017. [Online]. Available: https://arxiv.org/abs/1704.01212

[43] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," 2018. [Online]. Available: https://arxiv.org/abs/1711.07971

[44] J. B. Lee, R. A. Rossi, S. Kim, N. K. Ahmed, and E. Koh, "Attention models in graphs: A survey," 2018. [Online]. Available: https://arxiv.org/abs/1807.07984

[45] U. Alon and E. Yahav, "On the bottleneck of graph neural networks and its practical implications," 2021. [Online]. Available: https://arxiv.org/abs/2006.05205

[46] O. Shchur, M. Mumme, A. Bojchevski, and S. Günnemann, "Pitfalls of graph neural network evaluation," 2019. [Online]. Available: https://arxiv.org/abs/1811.05868

[47] Y. Dong, S. Wang, Y. Wang, T. Derr, and J. Li, "On structural explanation of bias in graph neural networks," 2022. [Online]. Available: https://arxiv.org/abs/2206.12104

[48] Y. Li, X. Wang, Y. Ning, and H. Wang, "Fairlp: Towards fair link prediction on social network graphs," *Proceedings of the International AAAI Conference on Web*

and Social Media, vol. 16, no. 1, pp. 628–639, May 2022. [Online]. Available: https://ojs.aaai.org/index.php/ICWSM/article/view/19321

[49] M. McPherson, L. Smith-Lovin, and J. M. Cook, "Birds of a feather: Homophily in social networks," *Annual Review of Sociology*, vol. 27, pp. 415–444, 2001. [Online]. Available: http://www.jstor.org/stable/2678628

[50] F. Masrour, T. Wilson, H. Yan, P.-N. Tan, and A. Esfahanian, "Bursting the filter bubble: Fairness-aware network link prediction," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 01, pp. 841–848, Apr. 2020. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/5429

[51] 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

[52] A. Subramonian, L. Sagun, and Y. Sun, "Networked inequality: Preferential attachment bias in graph neural network link prediction," 2024. [Online]. Available: https://arxiv.org/abs/2309.17417

[53] F. Santos, Y. Lelkes, and S. Levin, "Link recommendation algorithms and dynamics of polarization in online social networks," *Proceedings of the National Academy of Sciences*, vol. 118, p. e2102141118, 12 2021.

[54] T. Debono and F. Santos, *The Effect of Link Recommendation Algorithms on Network Centrality Disparities*, 03 2023, pp. 74–85.

[55] L. Lü and T. Zhou, "Link prediction in complex networks: A survey," *Physica A: Statistical Mechanics and its Applications*, vol. 390, no. 6, pp. 1150–1170, 2011. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S037843711000991X

[56] L. Weng, J. Ratkiewicz, N. Perra, B. Gonçalves, C. Castillo, F. Bonchi, R. Schifanella, F. Menczer, and A. Flammini, "The role of information diffusion in the evolution of social networks," 2013. [Online]. Available: https://arxiv.org/abs/1302.6276

[57] D. Li, Y. Zhang, Z. Xu, D. Chu, and S. Li, "Exploiting information diffusion feature for link prediction in sina weibo," *Scientific Reports*, vol. 6, p. 20058, 01 2016.

[58] D. Vega-Oliveros, L. Zhao, and L. Berton, "Evaluating link prediction by diffusion processes in dynamic networks," *Scientific Reports*, vol. 9, 07 2019.

[59] G. Jeh and J. Widom, "Simrank: a measure of structural-context similarity," in *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '02. New York, NY, USA: Association for Computing Machinery, 2002, p. 538–543. [Online]. Available: https://doi.org/10.1145/775047.775126

[60] S. Brin and L. Page, "Reprint of: The anatomy of a large-scale hypertextual web search engine," *Computer Networks*, vol. 56, no. 18, pp. 3825–3833, 2012, the WEB we live in. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128612003611

[61] L. Adamic and E. Adar, "Friends and neighbors on the web," *Social Networks*, vol. 25, pp. 211–230, 07 2003.

[62] B. Mønsted, P. Sapiezynski, E. Ferrara, and S. Lehmann, "Evidence of complex contagion of information in social media: An experiment using twitter bots," *PLOS ONE*, vol. 12, 03 2017.

[63] J. G. Sassine and H. Rahmandad, "How Does Network Structure Impact Socially Reinforced Diffusion?" *Organization Science*, vol. 35, no. 1, pp. 52–70, January 2024. [Online]. Available: https://ideas.repec.org/a/inm/ororsc/v35y2024i1p52-70.html

[64] D. Guilbeault and D. Centola, "Topological measures for identifying and predicting the spread of complex contagions," *Nature Communications*, vol. 12, 07 2021.

[65] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–44, 05 2015.

[66] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities." *Proceedings of the National Academy of Sciences*, vol. 79, no. 8, pp. 2554–2558, 1982. [Online]. Available: https://www.pnas.org/doi/abs/10.1073/pnas.79.8.2554

[67] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000. [Online]. Available: https://www.science.org/doi/abs/10.1126/science.290.5500.2323

[68] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000. [Online]. Available: https://www.science.org/doi/abs/10.1126/science.290.5500.2319

[69] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," 2016. [Online]. Available: https://arxiv.org/abs/1607.00653

[70] C. Merkwirth and T. Lengauer, "Automatic generation of complementary descriptors with molecular graph networks," *Journal of chemical information and modeling*, vol. 45, pp. 1159–68, 09 2005.

[71] E. Min, R. Chen, Y. Bian, T. Xu, K. Zhao, W. Huang, P. Zhao, J. Huang, S. Ananiadou, and Y. Rong, "Transformer for graphs: An overview from architecture perspective," 2022. [Online]. Available: https://arxiv.org/abs/2202.08455

[72] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2016. [Online]. Available: https://arxiv.org/abs/1409.0473

[73] 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

[74] T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, L. Màrquez, C. Callison-Burch, and J. Su, Eds. Lisbon, Portugal: Association for Computational Linguistics, Sep. 2015, pp. 1412–1421. [Online]. Available: https://aclanthology.org/D15-1166

[75] C. Ying, T. Cai, S. Luo, S. Zheng, G. Ke, D. He, Y. Shen, and T.-Y. Liu, "Do transformers really perform badly for graph representation?" in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34. Curran Associates, Inc., 2021, pp. 28 877–28 888. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2021/file/f1c1592588411002af340cbaedd6fc33-Paper.pdf

[76] A. Shehzad, F. Xia, S. Abid, C. Peng, S. Yu, D. Zhang, and K. Verspoor, "Graph transformers: A survey," 2024. [Online]. Available: https://arxiv.org/abs/2407.09777

[77] G. Berlusconi, F. Calderoni, N. Parolini, M. Verani, and C. Piccardi, "Link prediction in criminal networks: A tool for criminal intelligence analysis," *PLoS ONE*, vol. 11, 2016. [Online]. Available: https://api.semanticscholar.org/CorpusID:9556784

[78] M. Lim, A. Abdullah, N. Jhanjhi, and M. Supramaniam, "Hidden link prediction in criminal networks using the deep reinforcement learning technique," *Computers*, vol. 8, no. 1, 2019. [Online]. Available: https://www.mdpi.com/2073-431X/8/1/8

[79] A. M. Abdolhosseini-Qomi, N. Yazdani, and M. Asadpour, "Overlapping communities and the prediction of missing links in multiplex networks," 2020. [Online]. Available: https://arxiv.org/abs/1912.03496

[80] Z. Su, X. Zheng, J. Ai, Y. Shen, and X. Zhang, "Link prediction in recommender systems based on vector similarity," *Physica A: Statistical Mechanics and its Applications*, vol. 560, p. 125154, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378437120306038

[81] M. Vahidi Farashah, A. Etebarian, R. Azmi, and R. Ebrahimzadeh, "A hybrid recommender system based-on link prediction for movie baskets analysis," *Journal of Big Data*, vol. 8, 02 2021.

[82] T. Oyetunde, M. Zhang, Y. Chen, Y. J. Tang, and C. Lo, "Boostgapfill: improving the fidelity of metabolic network reconstructions through integrated constraint and pattern-based methods," *Bioinformatics*, vol. 33, p. 608–611, 2016. [Online]. Available: https://api.semanticscholar.org/CorpusID:206345281

[83] E. Nasiri, K. Berahmand, M. Rostami, and M. Dabiri, "A novel link prediction algorithm for protein-protein interaction networks by attributed graph embedding," *Comput. Biol. Med.*, vol. 137, no. C, Oct. 2021. [Online]. Available: https://doi.org/10.1016/j.compbiomed.2021.104772

[84] M. Zhang, *Graph Neural Networks: Link Prediction*. Singapore: Springer Nature Singapore, 2022, pp. 195–223. [Online]. Available: https://doi.org/10.1007/978-981-16-6054-2_10

[85] D. Arrar, N. Kamel, and A. Lakhfif, "A comprehensive survey of link prediction methods," *J. Supercomput.*, vol. 80, no. 3, p. 3902–3942, Sep. 2023. [Online]. Available: https://doi.org/10.1007/s11227-023-05591-8

[86] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, p. 509–512, Oct. 1999. [Online]. Available: http://dx.doi.org/10.1126/science.286.5439.509

[87] P. Jaccard, "Etude de la distribution florale dans une portion des alpes et du jura," *Bulletin de la Societe Vaudoise des Sciences Naturelles*, vol. 37, pp. 547–579, 01 1901.

[88] L. Katz, "A new status index derived from sociometric analysis," *Psychometrika*, vol. 18, pp. 39–43, 1953.

[89] H. Tong, C. Faloutsos, and J.-Y. Pan, "Fast random walk with restart and its applications," *Sixth International Conference on Data Mining (ICDM'06)*, pp. 613–622, 2006. [Online]. Available: https://api.semanticscholar.org/CorpusID:3926195

[90] W. Liu and L. Lü, "Link prediction based on local random walk," *EPL (Europhysics Letters)*, vol. 89, no. 5, p. 58007, Mar. 2010. [Online]. Available: http://dx.doi.org/10.1209/0295-5075/89/58007

[91] T. Zhou, L. Lü, and Y.-C. Zhang, "Predicting missing links via local information," *The European Physical Journal B*, vol. 71, no. 4, p. 623–630, Oct. 2009. [Online]. Available: http://dx.doi.org/10.1140/EPJB/E2009-00335-8

[92] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[93] M. E. J. Newman, "Network structure from rich but noisy data," *Nature Physics*, vol. 14, no. 6, p. 542–545, Mar. 2018. [Online]. Available: http://dx.doi.org/10.1038/s41567-018-0076-1

[94] A. Breit, S. Ott, A. Agibetov, and M. Samwald, "Openbiolink: a benchmarking framework for large-scale biomedical link prediction," *Bioinformatics*, vol. 36, no. 13, pp. 4097–4098, 2020.

[95] J. Kunegis and A. Lommatzsch, "Learning spectral graph transformations for link prediction," in *Proceedings of the 26th Annual International Conference on Machine Learning*, ser. ICML '09. New York, NY, USA: Association for Computing Machinery, 2009, p. 561–568. [Online]. Available: https://doi.org/10.1145/1553374.1553447

[96] G. Crichton, Y. Guo, S. Pyysalo, and A. Korhonen, "Neural networks for link prediction in realistic biomedical graphs: a multi-dimensional evaluation of graph embedding-based approaches," *BMC Bioinformatics*, vol. 19, pp. 1–11, 2018.

[97] X. Yue, Z. Wang, J. Huang, S. Parthasarathy, S. Moosavinasab, Y. Huang, S. M. Lin, W. Zhang, P. Zhang, and H. Sun, "Graph embedding on biomedical networks: methods, applications and evaluations," *Bioinformatics*, vol. 36, no. 4, pp. 1241–1251, 2020.

[98] M. Ali, C. T. Hoyt, D. Domingo-Fernández, J. Lehmann, and H. Jabeen, "Biokeen: a library for learning and evaluating biological knowledge graph embeddings," *Bioinformatics*, vol. 35, no. 18, pp. 3538–3540, 2019.

[99] J. Hanley and B. Mcneil, "The meaning and use of the area under a receiver operating characteristic (roc) curve," *Radiology*, vol. 143, pp. 29–36, 05 1982.

[100] M. Fey and J. E. Lenssen, "Fast graph representation learning with pytorch geometric," 2019. [Online]. Available: https://arxiv.org/abs/1903.02428

[101] A. McCallum, "Cora dataset," Ann Arbor, MI, 2017. [Online]. Available: https://doi.org/10.3886/E100859V1

[102] Z. Yang, W. W. Cohen, and R. Salakhutdinov, "Revisiting semi-supervised learning with graph embeddings," 2016. [Online]. Available: https://arxiv.org/abs/1603.08861

[103] C. L. Giles, K. D. Bollacker, and S. Lawrence, "Citeseer: an automatic citation indexing system," in *Proceedings of the Third ACM Conference on Digital Libraries*, ser. DL '98. New York, NY, USA: Association for Computing Machinery, 1998, p. 89–98. [Online]. Available: https://doi.org/10.1145/276675.276685

[104] J. Leskovec and J. Mcauley, "Learning to discover social circles in ego networks," in *Advances in Neural Information Processing Systems*, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds., vol. 25. Curran Associates, Inc., 2012.

[Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2012/file/7a614fd06c325499f1680b9896beedeb-Paper.pdf

[105] R. Yang, J. Shi, X. Xiao, Y. Yang, J. Liu, and S. S. Bhowmick, "Scaling attributed network embedding to massive graphs," *Proceedings of the VLDB Endowment*, vol. 14, no. 1, pp. 37–49, 2021.

[106] B. Rozemberczki, C. Allen, and R. Sarkar, "Multi-scale attributed node embedding," 2021. [Online]. Available: https://arxiv.org/abs/1909.13021

[107] B. Rozemberczki and R. Sarkar, "Characteristic functions on graphs: Birds of a feather, from statistical descriptors to parametric models," 2020. [Online]. Available: https://arxiv.org/abs/2005.07959

[108] M. Newman, *Networks: An Introduction*. Oxford: Oxford University Press, 2010. [Online]. Available: http://dx.doi.org/10.1093/acprof:oso/9780199206650.001.0001

[109] U. Brandes, "On variants of shortest-path betweenness centrality and their generic computation," *Social Networks*, vol. 30, no. 2, pp. 136–145, 2008. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378873307000731

[110] "Academic social networks: Modeling, analysis, mining and applications," *Journal of Network and Computer Applications*, vol. 132, pp. 86–103, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1084804519300438

[111] J. Hasell, "Measuring inequality: what is the gini coefficient?" *Our World in Data*, 2023, https://ourworldindata.org/what-is-the-gini-coefficient.

Acknowledgments

I would like to express my gratitude to my advisor, Dr. Alberto Testolin, for his guidance and help in the development of this work.

A special thanks to the IIIA-CSIC Institute, where this project first took shape. I am especially grateful to Dr. Jesús Cerquides and PhD student Björn Kommander for their mentorship and constructive feedback during this formative phase. Their expertise and dedication have significantly shaped the direction and depth of this work.