Data Mining Techniques for Graph and Hypergraph Analysis

Dissertation zur Erlangung des Doktorgrades
an der Fakultät für Mathematik, Informatik und Statistik
der Ludwig-Maximilians-Universität München

vorgelegt von

Peiyan Li

Munich, den 15.04.2025

Tag der Einreichung: 15.04.2025

Erstgutachter: Prof. Dr. Christian Böhm
Zweitgutachter: Prof. Lutz Oettershagen, Ph.D.
Drittgutachter: Prof. Yllka Velaj, Ph.D.

Tag der Disputation: 26.06.2025

Eidesstattliche Versicherung

(siehe Promotionsordnung vom 12.07.2011, § 8, Abs. 2 Pkt. 5)

Hiermit erkläre ich an Eidesstatt, dass die Dissertation von mir selbstständig, ohne unerlaubte Beihilfe angefertigt ist.

Munich, den 15.04.2025

Peiyan Li

Abstract

Graph structures are essential for modeling pairwise relationships in systems ranging from social networks to biological interactions and transportation infrastructure. However, in many real-world scenarios, relationships are often beyond pairwise. For example, social networks generally feature group structures where individuals belong to multiple groups simultaneously. The hypergraph structure has been extensively considered to model such higher-order relationships, wherein hyperedges can connect an arbitrary number of nodes. This thesis focuses on developing scalable and interpretable data mining algorithms for graph and hypergraph analysis, advancing techniques to handle complex relational patterns in these networks.

We explore information diffusion in hypergraphs and study the information coverage maximization problem in this scenario. Traditional information diffusion models are designed primarily for ordinary graphs. To address this limitation, we propose HIC, the Hypergraph Independent Cascade model, which extends the conventional independent cascade model to accommodate hypergraphs. Building on HIC, we propose a novel influence maximization problem: the information coverage maximization problem in hypergraphs. Unlike traditional influence maximization, which focuses on identifying influential nodes, we target to identify key groups. We establish the NP-hardness of this problem and demonstrate the submodular monotonicity of the information spread function. To solve the problem efficiently, we developed a heuristic approach called InfDis, inspired by the Degree Discount algorithm. Extensive experiments validate the effectiveness and efficiency of this approach.

The second task addressed in the thesis is hyperlink prediction, which involves predicting interactions among multiple entities. While existing solutions generally operate on the entire hypergraph, we propose the first subgraph-based hyperlink prediction approach that captures localized characteristics of central hyperedges while mitigating scalability concerns. The proposed method, SSF, focuses on localized subgraph patterns and extracts interpretable features using structural heuristics such as walks and loops. Additionally, its edge-weakening scheme adapts to varying hypergraph densities, enabling fine-grained feature learning. We conduct extensive experiments to validate SSF's adaptive capacity, evaluate the effectiveness of its feature components, and assess its robustness across various parameter configurations.

Next, we focus on a fundamental problem—graph classification. For this problem, we propose RWF, a graph fingerprinting technique that combines structural role-based vertex partitioning with local connection strength measurement. By creating soft alignments of node subsets across graphs of varying sizes, RWF generates topology-aware fingerprints that capture intra- and inter-subset connectivity. Further, RWF supports the integration of node attributes to enhance classification performance. Empirical assessment encompassing a wide range of graph datasets demonstrates that RWF achieves high computational efficiency while maintaining robust classification accuracy.

Collectively, this thesis introduces a novel problem formulation and presents three scalable, interpretable techniques designed to address key challenges in graph and hypergraph analysis.

Zusammenfassung

Graphstrukturen sind entscheidend für die Modellierung paarweiser Beziehungen in Systemen, die von sozialen Netzwerken über biologische Interaktionen bis hin zu Verkehrsinfrastrukturen reichen. In vielen realen Szenarien gehen Beziehungen jedoch oft über Paare hinaus. So weisen soziale Netzwerke beispielsweise Gruppenstrukturen auf, in denen Individuen gleichzeitig mehreren Gruppen angehören. Die Hypergraph-Struktur wird häufig genutzt, um solche höherstufigen Beziehungen abzubilden, wobei Hyperkanten eine beliebige Anzahl von Knoten verbinden können. Diese Arbeit konzentriert sich auf die Entwicklung skalierbarer und interpretierbarer Data-Mining-Algorithmen zur Analyse von Graphen und Hypergraphen, um Techniken für den Umgang mit komplexen relationalen Mustern in diesen Netzwerken voranzutreiben.

Wir untersuchen die Informationsverbreitung in Hypergraphen und adressieren das Problem der Maximierung der Informationsabdeckung. Herkömmliche Diffusionsmodelle sind primär für Standardgraphen konzipiert. Um diese Einschränkung zu überwinden, schlagen wir HIC (Hypergraph Independent Cascade) vor, das das klassische unabhängige Kaskadenmodell auf Hypergraphen erweitert. Basierend auf HIC formulieren wir ein neuartiges Einflussmaximierungsproblem: die Maximierung der Informationsabdeckung in Hypergraphen. Im Gegensatz zur traditionellen Einflussmaximierung, die einflussreiche Knoten identifiziert, zielen wir auf Schlüsselgruppen ab. Wir beweisen die NP-Härte dieses Problems und zeigen die submodulare Monotonie der Informationsausbreitungsfunktion. Zur effizienten Lösung entwickelten wir einen heuristischen Ansatz namens InfDis, inspiriert vom Degree-Discount-Algorithmus. Umfangreiche Experimente bestätigen die Wirksamkeit und Effizienz dieses Ansatzes.

Die zweite untersuchte Aufgabe ist die Hyperlink-Vorhersage, die die Prognose von Interaktionen zwischen mehreren Entitäten betrifft. Während bestehende Lösungen üblicherweise den gesamten Hypergraphen analysieren, schlagen wir den ersten teilgraphbasierten Ansatz vor, der lokalisierte Merkmale zentraler Hyperkanten erfasst und gleichzeitig Skalierbarkeitsprobleme mindert. Die Methode SSF konzentriert sich auf lokale Teilgraphmuster und extrahiert interpretierbare Merkmale mittels struktureller Heuristiken wie Pfaden und Schleifen. Ein integriertes Kantenabschwächungsschema passt sich variierenden Hypergraphdichten an und ermöglicht feingranulare Merkmal-

slernprozesse. Experimente validieren SSFs Adaptionsfähigkeit, die Effektivität seiner Merkmalskomponenten und seine Robustheit unter verschiedenen Parameterkonfigurationen.

Im dritten Schwerpunkt behandeln wir die Grundaufgabe der Graphklassifizierung. Hierfür entwickeln wir RWF, eine Graph-Fingerabdrucktechnik, die strukturell rollenbasierte Knotenpartitionierung mit lokalen Verbindungsstärkemessungen kombiniert. Durch weiche Ausrichtungen von Knotenteilmengen über Graphen variierender Größe erzeugt RWF topologiebewusste Fingerabdrücke, die intra- und intersubset-Konnektivität erfassen. Zudem unterstützt RWF die Integration von Knotenattributen zur Steigerung der Klassifizierungsleistung. Empirische Auswertungen über diverse Graphdatensätze zeigen, dass RWF hohe Recheneffizienz bei robusten Klassifizierungsgenauigkeiten erreicht.

Zusammenfassend führt diese Arbeit eine neuartige Problemformulierung ein und präsentiert drei skalierbare, interpretierbare Techniken zur Lösung zentraler Herausforderungen in der Graph- und Hypergraphenanalyse.

Acknowledgements

Completing this doctoral journey has been one of the most challenging yet rewarding endeavors of my life, and I am deeply grateful to the many individuals who supported, inspired, and guided me along the way:

- My supervisor, Prof. Dr. Christian Böhm, for giving me the opportunity to pursue a PhD, and thank you always for the freedom in research that empowered me to explore my ideas.

- Prof. Dr. Claudia Plant for valuable discussions and for bridging connections with colleagues at Wien University.

- My reviewers, Prof. Lutz Oettershagen and Prof. Yllka Velaj, for generously dedicating their time and expertise to evaluate this thesis.

- Prof. Liming Pan, who gave me valuable support in my research.

- Dr. Kai Li, my best friend, who is always there for me.

- My colleagues at LMU Munich and Wien University, Li Qian, Collin Leiber, Walid Durani, Lukas Miklautz, Ylli Sadikaj, Anna Beer, and Pascal Weber. I am happy for all the amazing colleagues I have met.

- My parents, my brother, and above all, my love, my wife, Honglian Wang, who accompanied me to go through many tough times.

Bibliographic Note

The research discussed in this cumulative thesis builds upon work originally disseminated through peer-reviewed conferences, where preliminary findings were published in their proceedings. The content is based on the following publications:

1. Peiyan Li, Honglian Wang, Kai Li, Christian Böhm
   Influence without Authority: Maximizing Information Coverage in Hypergraphs
   Proceedings of the 2023 SIAM International Conference on Data Mining (SDM). Society for Industrial and Applied Mathematics, 2023; [LWLB23]

2. Peiyan Li, Liming Pan, Kai Li, Claudia Plant, Christian Böhm
   Interpretable Subgraph Feature Extraction for Hyperlink Prediction
   Proceedings of the IEEE International Conference on Data Mining, ICDM, 2023; [LPL+23]

3. Peiyan Li, Honglian Wang, Christian Böhm
   Scalable Graph Classification via Random Walk Fingerprints
   Proceedings of the IEEE International Conference on Data Mining, ICDM, 2024;
   ICDM 2024 Best Paper Award [LWB24]

The author also contributed to the following publications:

i. Liming Pan, Hui-Juan Shang, Peiyan Li, Haixing Dai, Wei Wang, Lixin Tian
   Predicting Hyperlinks via Hypernetwork Loop Structure
   Europhysics Letters, 2021; [PSL+21]

ii. Honglian Wang, Peiyan Li, Wujun Tao, Bailin Feng, Junming Shao
   Learning Dynamic User Behavior Based on Error-driven Event Representation
   Proceedings of the Web Conference, 2021; [WLT+21]

iii. Honglian Wang, Peiyan Li, Yang Liu, Junming Shao
   Towards Real-time Demand-aware Sequential POI Recommendation
   Information Sciences, 2021; [WLLS21]

vi. Peiyan Li, Honglian Wang, Christian Böhm, Junming Shao
Online Semi-supervised Multi-label Classification with Label Compression and Local Smooth Regression
Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, 2020; [LWBS20]

vii. Peiyan Li, Honglian Wang, Jianyun Lu, Qinli Yang, Junming Shao
Community Detection with Local Metric Learning
Proceedings of the IEEE International Conference on Data Mining, 2020; [LWL+20]

viii. Peiyan Li, Chen Huang, Han Wang, Zhili Qin, Honglian Wang, Junming Shao
Exploiting Inconsistency Problem in Multi-label Classification via Metric Learning
Proceedings of the IEEE International Conference on Data Mining, 2020; [LQW+20]

x. Chen Huang, Peiyan Li, Chongming Gao, Junming Shao
Online Budgeted Least Squares with Unlabeled Data
Proceedings of the IEEE International Conference on Data Mining, 2019; [HLG+19]

xi. Zhong Zhang, Zhili Qin, Peiyan Li, Qinli Yang, Junming Shao
Multi-view Discriminative Learning via Joint Non-negative Matrix Factorization
Proceedings of the International Conference on Database Systems for Advanced Applications, 2018; [ZQL+18]

xii. Junming Shao, Zhongjing Yu, Peiyan Li, Wei Han, Christian Sorg, Qinli Yang
Exploring Common and Distinct Structural Connectivity Patterns Between Schizophrenia and Major Depression via Cluster-driven Nonnegative Matrix Factorization
Proceedings of the IEEE International Conference on Data Mining, 2017; [SYL+17]

Further, the following publication was invited to the KAIS journal. It is an extension of Publication 3 with at least 30% new content. At the time of writing, it is under review.

iv. Peiyan Li, Honglian Wang, Christian Böhm
Graph Classification at Scale: Leveraging Random Walk Topological Fingerprints
Submitted to Knowledge and Information Systems by invitation.

Contents

1. Introduction

In an increasingly interconnected world, data rarely exists in isolation. Instead, it is often linked by a network of dependencies, interactions, and other kinds of relationships. From social media and communication systems to transportation networks and e-commerce platforms, complex systems are naturally modeled as graphs, where entities are represented as nodes and their interactions as edges. For instance, online social platforms like $\mathbb{X}$ (formerly Twitter) and LinkedIn leverages graph structures to map connections between users. The Twitter graph was used to trace misinformation spread[CBHG12, MSGL14, BO22], and identify influential accounts that amplify content virally [MZY17], by analyzing retweet cascades and follower networks. On LinkedIn, nodes represent users or companies, while edges capture professional connections or employment histories. These graphs power features like “People You May Know” [HTK14] and job recommendations [CLZK18]. On the other hand, communication systems, such as wired and wireless network infrastructures, use graph modes to develop load balancing strategies [FB15, TSO21, AE21] and optimize resource allocation [COZ$^+$17]. For example, data centers model server clusters as graphs, where edge weights represent bandwidth or latency. Algorithms like max-flow min-cut optimize data routing to prevent bottlenecks [AE21], while graph partitioning techniques distribute workloads across servers to minimize energy consumption [FB15]. Similarly, transportation networks like air traffic and urban transit systems leverage graphs to optimize connectivity and resilience [ZMHD15, AMB19], and e-commerce platforms like Amazon and Taobao utilize graph-based recommendation systems to suggest products based on user purchasing patterns and preferences [SRJ17, LZW$^+$21].

While graphs capture pairwise relationships between entities, many real-world phenomena involve interactions among multiple entities simultaneously. Hypergraphs extend traditional graphs by allowing hyperedges to connect any number of nodes[1], making them suitable for modeling group interactions. For instance, many online social networks, e.g., Facebook, WhatsApp, and WeChat, can be viewed as hypergraphs. A WhatsApp group chat forms a hyperedge linking all participants, enabling simultaneous com-

[1] An edge is a special type of hyperedge that connects only two nodes.

munication [LWLB23]. Similarly, the server structure of Discord [Rob23], where users join channels dedicated to specific topics, is fundamentally hypergraph-based. Meetup, which connects users through shared events or hobbies, further illustrates how hyperedges represent collective engagement [LHT$^+$12]. Beyond social networks, hypergraphs are also valuable in biological research, where they model multi-way interactions such as metabolic reactions and protein interactions, with hyperedges representing the resulting products [PCC14, VP19]. Other examples include collaborative research teams[TCR10], email threads [BAS$^+$18], and any other multi-agent systems [YHS$^+$24].

As the examples show, graph and hypergraph structures are essential in data science, preserving interconnected systems' rich relational information. By modeling entities as nodes and interactions as edges or hyperedges, these structures capture pairwise or higher-order relationships that traditional tabular or vector-based representations often overlook. Therefore, analyzing graphs and hypergraphs has emerged as a critical target of data mining, a core step of the Knowledge Discovery in Database (KDD) process. According to the classical definition in [FPSS96], KDD encompasses a five-step pipeline for transforming raw data into actionable knowledge, including data cleaning, integration, transformation, pattern extraction (via data mining), and interpretation. In this context, data mining specifically focuses on the automated discovery of non-trivial patterns, correlations, or anomalies within large datasets. When applied to graph or hypergraph data, this process is termed graph mining or hypergraph mining, respectively.

Different mining tasks lead to distinct types of actionable insights. For instance, community detection algorithms identify clusters of nodes characterized by high internal density and sparse interconnections [LF09, YAT16, JYJ$^+$21]; centrality measures reveal key influencers in networks [LFH10, GSC16, OMK22]; and link prediction algorithms infer missing or future relationships [LZ11, SZA$^+$20]. Over the past four decades, the graph mining domain has been shaped by diverse methodological paradigms and a wide range of mining tasks [CH06, CF06, XSY$^+$21]. Given the breadth and complexity of graph mining, we only get an overview of key technical trends. Technically, early works relied on probabilistic graphical models and statistical analysis [KS80, SV93, Mos87]. Subsequent progress shifted toward matrix-based approaches, e.g., graph spectral analysis [PSL90, FBCM04, CRS97], PageRanks [Hav99, BGS05, XG04], label propagation [WZ06, ZFNP15] and matrix factorization [KDP12, ZSBS12, QDM$^+$19, ME11], which dominated the landscape until the rise of deep learning for graphs [KW17, VCC$^+$18, HYL17, Jin21]—the current state-of-the-art for many graph-related tasks. In contrast to graph mining, hypergraph researches [FYZ$^+$19, LYS22, ACP$^+$23] have gained increasing attention only in the past decade, representing a relatively new area for exploration [2].

[2] According to a Web of Science search in January 2025, there are 14,296 papers in the graph mining domain (excluding hypergraph mining) and 366 papers focused on hypergraph mining.

1.1 Research Goals

In this cumulative thesis, we focus on data mining for graphs and hypergraphs, addressing three interconnected research tasks. While distinct in their objectives, these tasks collectively shift the analytical focus from traditional node-centric perspectives to higher-order abstractions.

1. **Information Diffusion and Influence Maximization in Hypergraphs**
   In this task, we aim to design a scalable solution for identifying highly influential hyperedges under a hypergraph-aware information diffusion model. Our problem formation requires incorporating group interactions, where influence diffusion depends on synergistic relationships among multiple nodes. Further, the target is to find a seed set of hyperedges that trigger a cascade with the maximum information coverage.

2. **Hyperlink Prediction**
   For hyperlink prediction, the goal is to derive an interpretable and scalable feather extraction technique for predicting the existence of potential hyperedges. This research goal is motivated by the observation that hyperlink prediction algorithms tend to be latent-based, which are difficult to interpret, and these solutions face high complexities in general, accounting for the combinatorial complexity of hyperedges.

3. **Graph Classification**
   Similar to the second research goal but focusing on a different task, the third research goal is to enable robust classification of graphs with heterogeneous sizes and structures through scalable, unsupervised feature extraction. This task emphasizes representation learning at the graph level.

We start with information diffusion and influence maximization in hypergraphs [LWLB23]. We contribute a new information diffusion model, Hypergraph Independent Cascade (HIC), which generalizes the classical Independent Cascade model [KKT03] by allowing group-level information diffusion. Building on this, we formalize a novel influence maximization problem for hypergraphs: identifying an optimal set of seed hyperedges to maximize information coverage. We establish the NP-hardness of the problem, prove that computing information coverage is #P-hard, and demonstrate the submodular monotonicity of the coverage function. While the classical greedy algorithm offers theoretical guarantees, its computational inefficiency limits practical utility. To overcome this, we propose Influence Discount (InfDis), a heuristic method that achieves comparable performance with reasonable interpretation while reducing runtime by four orders of magnitude compared to the naive greedy algorithm.

For hyperlink prediction, we contribute Subgraph Structural Features (SSF) [LPL$^+$23], the first subgraph-based feature extraction technique designed for this task. The rationale behind SSF is that hyperedges and non-hyperedges exhibit distinct local patterns. For this purpose, we use well-established structural heuristics like loop and walk as the basic feature and enhance the conventional subgraph extraction technique [ZC18] with an edge-weakening scheme. The contributed feature extraction technique is unsupervised and independent of trainable components. The extracted features are interpretable and are computationally efficient. The edge-weakening scheme enables SSF to adaptively encode structural information across dense and sparse hypergraphs, enabling robust performance across diverse datasets. Experimental results demonstrate that SSF is a lightweight yet powerful solution for hyperlink prediction.

We introduce Random Walk Fingerprints (RWF) [LWB24], a new unsupervised feature extraction technique for graph classification. RWF employs a structural-role-based vertex partitioning scheme to generate subset-based graph representations, enabling soft structural alignments across graphs of different sizes. Building on the node partitions, SSF computes three structural features, reflecting the connection strengths within and between node subsets. Additionally, a highlight of RWF is its flexibility in incorporating node features, making it a comprehensive tool for feature extraction. The overall complexity of RWF is competitive compared to a wide range of graph classification algorithms, and it shows robust classification results.

1.2 Thesis Structure

The thesis is structured as follows. Chapter 1 provides an overview of graph and hypergraph data mining, and outlines the three research tasks covered in this thesis. Chapter 2 introduces necessary preliminaries to help elaborate the contributions made in this thesis. Chapter 3 presents the detailed contributions. Chapter 4 includes the discussion, limitations, future directions, and final remarks. Appendix A contains the three papers and the detailed listings of author contributions.

2. Foundations

In this chapter, we first give several foundational concepts of graphs and hypergraphs. Then, we introduce two lines of graph representation learning, which includes non-deep techniques and deep-learning-based approaches. After that, we introduce how graph neural neural networks are extended to hypergraphs. These representation learning methodologies form the technical backbone of many benchmark approaches analyzed in this thesis. Lastly, we introduce the problem of influence maximization and its variants.

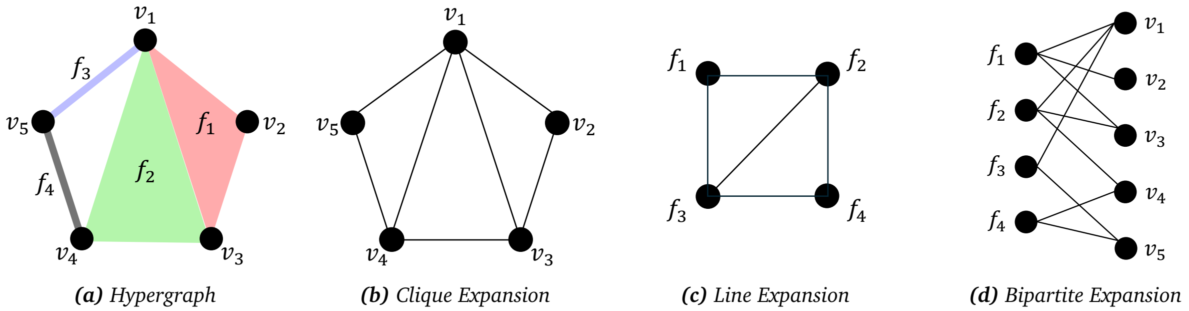
2.1 Graphs and Hypergraphs

We define a weighted graph $\mathcal{G}$ by an ordered pair $(\mathcal{V}, \mathcal{E}, \boldsymbol{w})$, where $\mathcal{V} = \{v_1, \ldots, v_n\}$ is a finite set of nodes, $\mathcal{E} = \{e_1, \ldots, e_m\}$ is a finite set of edges, and $\boldsymbol{w}$ is a vector representing edge weights. For a graph $\mathcal{G}$, its edges satisfy $|e| = 2$, and two nodes $u, v \in \mathcal{V}$ are adjacent if $\{u, v\} \in \mathcal{E}$. The weighted degree of a node v, denoted $d(v)$, is a product of the number of edges incident to v and the edge weights, i.e., $d(v) = \sum_{e \ni v} w_e$. The adjacency matrix of $\mathcal{G}$ is a $|\mathcal{V}| \times |\mathcal{V}|$ matrix $\mathbf{A}$ with:

$$\mathbf{A}_{uv} = \begin{cases} w_{uv} & \text{if } \{u, v\} \in \mathcal{E}, \\ 0 & \text{otherwise}. \end{cases}$$

When $\mathcal{G}$ is unweighted, $w_{uv} = 1$ for all $\{u, v\} \in E$. If G is weighted, $w_{uv} > 0$ for all $\{u, v\} \in E$. We do not consider edges with negative weights. In many real-world scenarios, a graph may be associated with node attributes. We use $\mathbf{X} \in \mathbb{R}^{p \times |V|}$ to denote the node feature matrix, where p is the dimension of node attributes.

Similar to the definition of a graph, we use $\mathcal{H} = (\mathcal{V}, \mathcal{F}, \boldsymbol{w}_f)$ to represent a weighted hypergraph, where $\mathcal{V} = \{v_1, \ldots, v_n\}$ is the set of nodes, $\mathcal{F} = \{f_1, \ldots, f_m\} \subset 2^{\mathcal{V}}$ is the set of hyperedges, and $\boldsymbol{w}_f$ is a vector of hyperedge weights. In this thesis, we do not consider hyperedges which contains only one node, and we simply let $|f| \geq 2$. The hypergraph



***(a)** Hypergraph* ***(b)** Clique Expansion* ***(c)** Line Expansion* ***(d)** Bipartite Expansion*

Figure 2.1: Illustration of a Hypergraph and its Graph Expansions: *The images show a hypergraph with nodes $\{v_1, v_2, v_3, v_4, v_5\}$ and hyperedges $\{f_1, f_2, f_3, f_4\}$, and its graph expansions.*

can be represented by an incidence matrix $\mathbf{S} \in \mathbb{R}^{|\mathcal{F}|\times|\mathcal{V}|}$, where:

$$\mathbf{S}_{fv} = \begin{cases} 1 & \text{if } v \in f, \\ 0 & \text{otherwise.} \end{cases}$$

For a hyperedge $f \in \mathcal{F}$, its degree is defined by the cardinality of the hyperedge, i.e., $\delta(f) = |f|$. For a node v in a hypergraph $\mathcal{H}$, its weighted degree is defined by $d(v) = \sum_{f\in\mathcal{F}} w_f \mathbf{S}_{fv}$. We use $\mathbf{D}_v \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$, $\mathbf{D}_f \in \mathbb{R}^{|\mathcal{F}|\times|\mathcal{F}|}$ and $\mathbf{W} \in \mathbb{R}^{|\mathcal{F}|\times|\mathcal{F}|}$ to denote the diagonal matrices containing the weighted node degrees, the hyperedge degrees and the hyperedge weights, respectively.

2.1.1 Hypergraph Expansion

In hypergraph analysis, a common practice involves projecting hypergraphs onto ordinary graphs. Here, we outline three widely adopted hypergraph expansion techniques:

- **Clique Expansion.** This method replaces each hyperedge with a clique, forming a weighted graph.

- **Line Expansion**[3]**.** This technique maps hyperedges to nodes in a new graph, connecting two hyperedges if they share at least one node.

- **Bipartite Expansion (also called Star Expansion).** In this approach, hyperedges are treated as independent nodes, forming a bipartite graph where original nodes connect only through hyperedges. Each hyperedge links to its constituent nodes but not directly to other hyperedges.

[3] There are other line expansion techniques, e.g., [LFA+22, KMK24]. In this thesis, when we use the notion of line expansion, if not explicitly stated, it specifically refers to the case illustrated in Fig. 2.1c

Among these expansion techniques, clique expansion is the most widely used technique due to its simplicity. With clique expansion, a hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{F})$ is projected to a weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \boldsymbol{w})$. In the matrix form, $\mathcal{G}$ can be represented by a weighted adjacency matrix $\mathbf{A} = \mathbf{S}^{\mathbf{T}}\mathbf{S}$. If removing self-loops in $\mathcal{G}$, $\mathbf{A} = \mathbf{S}^{\mathbf{T}}\mathbf{S} - \mathtt{diag}(\mathbf{S}^{\mathbf{T}}\mathbf{S})$, where $\mathtt{diag}(\mathbf{S}^{\mathbf{T}}\mathbf{S})$ is a diagonal matrix whose diagonal entries are identical to those of $\mathbf{S}^{\mathbf{T}}\mathbf{S}$. Similarly, for a weighted hypergraph, the adjacency matrix of its clique expansion is $\mathbf{A} = \mathbf{S}^{\mathbf{T}}\mathbf{W}\mathbf{S}$, where $\mathbf{W}$ is a diagonal matrix with entries corresponding to the hyperedge weights. If removing self-loops, $\mathbf{A} = \mathbf{S}^{\mathbf{T}}\mathbf{W}\mathbf{S} - \mathbf{D}_v$.

There are a few studies [CR19, KMK24, KLG+24] discussing the information loss caused by hypergraph expansion techniques. It is a critical challenge in applications where preserving the original hypergraph's structural semantics is essential. We borrow the concepts in [KMK24] to depict the limitations of the three hypergraph expansion techniques. Clique expansion and line expansion are *non-recoverable*, as the expanded graph cannot recover its corresponding hypergraph without additional knowledge; and bipartite expansion suffers from *tie-weakening*, a phenomenon where indirect connections between nodes, i.e., nodes are connected through hyperedge nodes, weaken the relationships.

To cope with the potential information loss caused by hypergraph expansion, hybrid strategies that combine multiple expansion techniques are considered. For instance, [YWYA22] uses a combination of bipartite expansion and clique expansion; [YCW+24] proposes cross expansion, which is a combination of the three hypergraph expansion techniques. These approaches aim to compensate for individual methods' shortcomings. For example, clique expansion loses global hyperedge context, and bipartite expansion may cause edge sparsity by combining complementary perspectives. Nevertheless, selecting optimal expansion strategies remains a nontrivial problem, as the choice hinges on both the properties of datasets (e.g., hyperedge cardinality and sparsity) and the objectives of the downstream task.

2.1.2 Random Walk

A random walk on a graph is a stochastic process that starts at a given node, and then iteratively transitions to neighboring nodes. In an unweighted graph, random walk progresses by selecting the next node from the current node's neighbors uniformly at random. For a weighted graph, the process adapts such that the probability of transitioning to each neighboring node is proportional to the weight of the connecting edge.

In a naive random walk, the transition matrix for an unweighted undirected graph $\mathcal{G}$ is $\mathbf{P} = \mathbf{D}^{-1}\mathbf{A}$, where $\mathbf{A}$ is the corresponding adjacency matrix. Its entry, e.g., $\mathbf{P}_{uv} = \frac{\mathbf{A}_{uv}}{d(v)}$, represents the probability of moving from node u to v.

There are many variations of random walks, such as biased random walks [Alt80, HH97, FF09], lazy random walks [SDWL14], random walk with restart [TFP06] and Lévy Walks [ZDK15]. These adaptations are widely employed to simulate complex phenomena in physics, biology, and network science. For a systematic review of random walk theory and applications, please refer to the following book [Woe17] and surveys [XLN$^+$19, MPL17].

When it comes to hypergraphs, there are two main lines of adaption. The first line is similar to simple random walks. For a walk that starts at an arbitrary node u:

1. Select a hyperedge $f \ni u$ with probability proportional to the hyperedge weight $w_f/d(u)$.
2. Select a node $v \in e$ uniformly at random, and set the current node to v.

Let $\mathbf{P}$ be the transition matrix for the above hypergraph random walk. Then each entry of $\mathbf{P}$ is:

$$\mathbf{P}_{uv} = \sum_{f \ni u} \frac{w_f}{d(u)} \frac{\mathbf{S}_{fv}}{\delta(f)}$$

where $d(u)$ and $d(v)$ denote the weighted degree of node u and v, respectively. In a matrix form:

$$\mathbf{P} = \mathbf{D}_v^{-1}\mathbf{S}^T\mathbf{W}\mathbf{D}_e^{-1}\mathbf{S}$$

It has been shown in [CR19] that the above hypergraph random walk is the same as a simple random walk on the graph clique expansion of the hypergraph. A corollary is proposed in the same paper: edge-dependent vertex weights might be a key to utilizing higher-order information. In that way, a hypergraph is represented by a weighted incidence matrix $\mathbf{R} \in \mathbb{R}^{|\mathcal{F}| \times |\mathcal{V}|}$ such that:

$$\mathbf{R}_{fv} = \begin{cases} \gamma_f(v) & \text{if } v \in f, \\ 0 & \text{otherwise}. \end{cases}$$

Correspondingly, the hypergraph random walk has the following steps:

1. Select a hyperedge $f \ni u$ with probability proportional to the hyperedge weight $w_f/d(u)$.
2. Select a node $v \in e$ with probability $\gamma_f(v)/\delta(f)$, and set the current node to v.

Thus, the entry of the transition matrix becomes:

$$\mathbf{P}_{uv} = \sum_{f \in \mathcal{F}(u)} \frac{w_f}{d(u)} \frac{\gamma_f(v)}{\delta(f)}.$$

In a matrix form:

$$\mathbf{P} = \mathbf{D}_v^{-1}\mathbf{S}^T\mathbf{W}\mathbf{D}_e^{-1}\mathbf{R}$$

The second line of random walks, incorporating edge-dependent vertex weights, has shown interesting results in hypergraph community detection [HAPP20] and node ranking [CR19]. A potential limitation is a lack of benchmark datasets since specifying edge-dependent vertex weights requires additional knowledge of the data. Current implementations typically rely on ad hoc heuristics or random distributions to assign these weights.

2.2 Non-Deep Graph Representation Learning

From a broader view, graph representation learning aims to answer the following question: How can we encode the structural richness of graphs and the semantic information associated with nodes and edges into numerical representations while preserving their inherent properties? The solution is graph embedding, a family of techniques that focuses on preserving these complex structures and associated attribute information into compact low-dimensional vector spaces. These embeddings thus enable downstream tasks like node classification/clustering, link prediction, edge classification, and graph classification/clustering. We review the main technical trend of these methodologies in this section and section 2.3. This section only introduces non-deep graph embeddings, including three categories of methodologies: graph statistics, matrix factorization, and random walk-based embeddings.

Graph Statistics Early graph analysis uses handcrafted topological metrics to characterize node/graph properties. At the node level, degree centrality provides basic connectivity information, while clustering coefficients [WS98] quantify local triangle density to identify community structures. Higher-order centrality measures, such as betweenness centrality [Fre77] for identifying bridge nodes and eigenvector centrality [Ruh00] for measuring influence through recursive neighbor importance, are used to capture the structural roles of nodes. At the graph level, global statistics like diameter [Bol81], density [Law01], modularity or assortativity [New02] summarize macroscopic properties. These graph statistics provide valuable insights for characterizing nodes and graphs. Currently, the study of graph statistics does not have a leading role. However, these statistics still provide guidance or act as complementary tools for downstream tasks [LLC$^+$20, ALH22, LFZ$^+$24].

Matrix Factorization Classical spectral methods like Laplacian Eigenmaps [BN03] and Spectral Clustering [NJW01] rely on graph Laplacian matrices to embed nodes by optimizing the spectral properties of graphs. These techniques operate under the principle that the eigenvectors of the graph Laplacian capture meaningful low-dimensional structure. For instance, Laplacian Eigenmaps explicitly minimize a cost function involving the graph's Laplacian to preserve locality, while Spectral Clustering partitions graphs using the eigenvectors of the normalized Laplacian to group nodes with strong connectivity. By focusing on the low-frequency eigenvectors of the Laplacian matrices (associated with the smallest eigenvalues), these methods emphasize smoothness over the graph structure, preserving global connectivity patterns such as community hierarchies or manifold geometry. The spectral methods generally contain eigenvalue decomposition and operate on the Laplacian matrices. Other kinds of matrix factorization techniques operate on the adjacency or feature matrices, employing different kinds of data assumptions like low-rankness [HCD12, TLSZ14, NVM$^+$18, EASDP20] and sparsity [SXZF07, KR12, CSX12]. The former assumes that data lies on a low-dimensional space, often relevant to dimensionality reduction and graph clustering, while the latter assumes that data can be represented as a linear combination of a small number of basis elements and is typically favored in scenarios requiring feature selection and noise reduction.

Random-Walk Based Embeddings DeepWalk [PARS14] is a pioneer study that first adapts word embedding techniques to graphs. In DeepWalk, nodes are treated as words, and sentences are node sequences generated via truncated random walk. DeepWalk processes them by applying Skip-Gram [MCCD13], which maximizes the likelihood of context nodes (within a window) given a target node. LINE [TQW$^+$15] explores a similar idea but eliminates random walks. It directly models first-order proximity (direct edges) and second-order proximity (shared neighbors) through explicit objective functions. Further, it utilizes edge sampling and negative sampling to train embeddings directly, making it suitable for large-scale networks. Node2Vec [GL16] extends DeepWalk by introducing a biased random walk strategy controlled by parameters, which balance breadth-first (BFS) and depth-first (DFS) sampling, enabling embeddings to interpolate between homophily and structural equivalence. While more flexible than DeepWalk, its performance remains sensitive to hyperparameter tuning and walk design. Notably, the three random-walk-based methods have been theoretically unified under a matrix factorization framework [QDM$^+$18], highlighting their underlying conceptual coherence.

2.3 Deep Graph Representation Learning

Deep learning for graphs revolutionized graph representation learning by introducing graph neural networks (GNNs). Following the classical taxonomy, we broadly categorize GNNs into spectral-based and spatial-based approaches. Additionally, we introduce graph transformers and unsupervised graph representation learning techniques.

2.3.1 Spectral-based and Spatial-based Graph Neural Networks

Spectral-based Graph Neural Networks The foundation of spectral-based GNNs [DBV16, KW17, XSC+18, BGLA21] is the Laplacian matrix $\mathbf{L} = \mathbf{D} - \mathbf{A}$. Its normalized version is $\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$. For undirected graphs, both Laplacian matrices are symmetric, enabling eigen-decomposition of the form: $\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$, where $\mathbf{U}$ is a matrix of eigenvectors, and $\mathbf{\Lambda}$ contains eigenvalues. Borrowing the concept of Fourier Transform from the field of Digital Signal Processing (DSP), a signal $\boldsymbol{x} \in \mathbb{R}^{|\mathcal{V}| \times 1}$ on a graph is transformed to the spectral domain via $\hat{\boldsymbol{x}} = \mathbf{U}^T\boldsymbol{x}$, and the signal reconstructs back to the spatial domain via $\boldsymbol{x} = \mathbf{U}\hat{\boldsymbol{x}}$. Here, $\mathbf{U}$ servers as the Fourier basis. A learnable filter $g(\mathbf{\Lambda})$ operates on the eigenvalues to perform filtering in the spectral domain. The filtered signal becomes:

$$\boldsymbol{x}_{out} = \mathbf{U}g(\mathbf{\Lambda})\mathbf{U}^T\boldsymbol{x}$$

Different spectral-based GNNs propose distinct parameterizations of $g(\mathbf{\Lambda})$ to balance expressiveness and computational efficiency. Here, we list several representative methods:

- **ChebyNet** [DBV16] uses Chebyshev polynomials $\mathbf{T}_k$ to approximate the polynomial filter: $g(\mathbf{\Lambda}) = \sum_{k=0}^{K-1} \theta_k \mathbf{\Lambda}^k$, where $\theta \in \mathbb{R}^K$ is a vector of polynomial coefficients. This approximation avoids eigen-decomposition by leveraging the recurrence relation of Chebyshev polynomials:

$$g(\Lambda) \approx \sum_{k=0}^{K} \alpha_k \mathbf{T}_k(\tilde{\mathbf{\Lambda}}), \quad \tilde{\mathbf{\Lambda}} = \frac{2\mathbf{\Lambda}}{\lambda_{\max}} - I$$

  where $\alpha \in \mathbb{R}^K$ is a vector of Chebyshev coefficients, and $\mathbf{T}_k(\tilde{\mathbf{\Lambda}})$ is the Chebyshev polynomial. The localized filter enables efficient computation while capturing multi-hop neighborhood information.

- **GCN** [KW17] simplify ChebyNet with $K = 1$, reducing the polynomial to a first-order approximation. It introduces a renormalization trick and adds self-loops to stabilize training. GCN acts as a low-pass filter that smooths node features across immediate neighbors.

- **GWNN** [XSC+18] replaces the Fourier basis with spectral graph wavelets for localized filtering. The filter becomes: $g(\Lambda) = \mathbf{U}e^{-s\mathbf{\Lambda}}\mathbf{U}^\top$, where s is a scaling parameter that enables multi-resolution analysis. GWNN avoids global Fourier transforms and reduces computational complexity via fast wavelet transforms.

- **ARMA** [BGLA21] approximates the filter using an autoregressive-moving-average

(ARMA) process:

$$g(\mathbf{\Lambda}) = \left(\mathbf{I} + \sum_{k=1}^{K} q_k \mathbf{\Lambda}^k\right)^{-1} \sum_{k=0}^{K-1} p_k \mathbf{\Lambda}^m,$$

where q_k, p_k are learnable parameters. ARMA filters model long-range dependencies without learning high-degree polynomials, which improves efficiency and stability.

Spatial-based Graph Neural Networks In contrast to spectral-based GNNs, which leverage the graph spectrum, spatial-based GNNs operate directly on the graph topology by aggregating features from local neighborhoods. These architectures can be unified by a framework called Message Passing Neural Networks [GSR+17], where each node iteratively updates its representation by aggregating features from its neighbors. Formally, the update rule for a node v at the ℓ-th layer is defined as:

$$\begin{aligned} h_{N(v)}^{(\ell)} &= \mathrm{AGGREGATION}\left(\left\{h_v^{(\ell-1)}, h_u^{(\ell-1)} : u \in N(v)\right\}\right) \\ h_v^{(\ell)} &= \mathrm{COMBINATION}\left(h_{N(v)}^{(\ell)}, h_v^{(\ell-1)}\right) \end{aligned}$$

where $\mathrm{AGGREGATION}$ and $\mathrm{COMBINATION}$ are parameterized functions, $h_v^{(\ell-1)}$ is the representation of node v at the $(\ell-1)$-th layer, encoding the $(l-1)$-hop structure information. $h_{N(v)}^{(l)}$ denotes the aggregated features from the node and its neighborhood $N(v)$. Below, we show how key spatial-based GNNs instantiate this framework through distinct aggregation and combination strategies.

- **GraphSAGE** [HYL17] enables inductive learning on large graphs by sampling neighbors and aggregating their features. Its feature aggregation operates on a fixed-size sample of $N(v)$. Common aggregation functions include Mean, Max, and LSTM.

  - **Mean**:

$$h_{N(v)}^{(\ell)} = \frac{1}{|N(v)|} \sum_{u \in N(v)} h_u^{(\ell-1)}$$

  - **Max**:

$$h_{N(v)}^{(\ell)} = \max_{u \in \mathcal{N}(v)} \sigma\left(\mathbf{W}_{\mathrm{pool}} h_u^{(\ell-1)} + \mathbf{b}_{\mathrm{pool}}\right)$$

  - **LSTM**: Although LSTM [HS97] is not permutation invariant, GraphSAGE uses a LSTM on random permutation of neighbors as a more expressive aggregation function.

After feature aggregation, GraphSAGE combines the node's own features with the aggregated neighborhood features, followed by a learnable transformation:

$$h_v^{(\ell)} = \sigma\left(\mathbf{W}^{(\ell)} \cdot \left[h_v^{(\ell-1)} \parallel h_{N(v)}^{(\ell)}\right]\right),$$

where $\parallel$ denotes concatenation.

- **GAT** [VCC$^+$18] uses attention mechanisms to weigh the importance of neighboring nodes dynamically. It computes attention coefficients α_{vu} between v and each neighbor u via:

  $$\alpha_{vu} = \text{softmax}_u\left(\text{LeakyReLU}\left(\mathbf{a}^T\left[\mathbf{W}h_v^{(\ell-1)} \parallel \mathbf{W}h_u^{(\ell-1)}\right]\right)\right),$$

  where a and $\mathbf{W}$ are learnable parameters. The aggregated message is a weighted sum:

  $$h_{N(v)}^{(\ell)} = \sum_{u \in \mathcal{N}(v)} \alpha_{vu}\mathbf{W}h_u^{(\ell-1)}.$$

  To stabilize the learning process of self-attention, GAT uses multi-head attention, where K independent attention mechanisms are executed. After that, it applies a nonlinear activation to the aggregated result, and concatenate the features:

  $$h_v^{(\ell)} = \parallel_{k=1}^{K} \sigma\left(h_{N(v),k}^{(\ell)}\right)$$

  The aggregation process can also be performed by employ averaging:

  $$h_v^{(\ell)} = \sigma\left(\frac{1}{K}\sum_{k=1}^{K} h_{N(v),k}^{(\ell)}\right)$$

- **GIN** [XHLJ19] improves the discriminative power of GNNs by designing injective aggregation functions, aligning with the Weisfeiler-Lehman (WL) graph isomorphism test. Its aggregation function is a sum of neighbor features:

  $$h_{N(v)}^{(\ell)} = \sum_{u \in N(v)} h_u^{(\ell-1)}.$$

  For the combination part, it injects the node's own features via a learnable scaling parameter $\epsilon^{(\ell)}$, followed by an MLP:

  $$h_v^{(\ell)} = \text{MLP}^{(\ell)}\left(\left(1 + \epsilon^{(\ell)}\right) \cdot h_v^{(\ell-1)} + h_{N(v)}^{(\ell)}\right).$$

After feature aggregation and combination, these hidden node representations can be read out into a graph-level representation for graph classification or regression. Alternatively, they may be fed into task-specific components for node-level or edge-level predictions.

Spectral vs. Spatial It should be noted that the distinction between spectral and spatial GNNs originates from their mathematical foundations. However, there is not a clear boundary between them. Spectral GNNs such as GCN and ChebyNet can also be described in the MPNN framework [GSR$^+$17, BRH$^+$21, BHG$^+$21, GMP21]. For instance, the spectral convolution of GCN can be reinterpreted as a mean aggregation step in spatial message passing [BRH$^+$21]. This convergence highlights that spectral operations often implicitly define spatial aggregation rules, enabling theoretical analyses of expressive power, e.g., the equivalence between GCN and the Weisfeiler-Lehman test [GMP21].

2.3.2 Graph Transformers

At the time of writing the thesis, Graph Transformers [DB21, YCL$^+$21, KBH$^+$21, RGD$^+$22, CHYW23, XWL$^+$24] have emerged as a new paradigm of graph neural networks. A graph transformer is an adaptation of the transformer architecture [VSP$^+$17], initially designed for sequential data. It modifies the self-attention mechanism to incorporate the graph structure [DB21].

Graph transformers address critical limitations of message-passing GNNs, such as over-squashing, limited long-range dependency modeling, and insufficient utilization of edge features. The key technical innovations of graph transformers include:

- **Positional/Structural Encoding** aims to incorporate the absolute or relative location of nodes into the embedding. Common strategies include Laplacian eigenvectors [DB21], random walk probabilities [DLL$^+$22], shortest path distances [YCL$^+$21] and node centrality matrices [YCL$^+$21]. In practice, these strategies are usually combined.

- **Global Attention** aims to capture long-range dependencies among nodes. Unlike GAT, which performs the attention mechanism in the local neighborhood, graph transformers use global attention and depict interactions between all node pairs. In awareness of the high computational costs of global dense attention, some studies also consider sparse attention, which typically relies on sampling [SVV$^+$23, SLV$^+$25].

- **Handling Edge Features** [DB21, RBX$^+$20] is a byproduct of positional encoding and the attention mechanism since positional encodings indirectly encode edge information into node representations, and the attention mechanism dynamically adapts interactions between nodes based on edge features. This enables the utilization of both node features and edge semantics, which are particularly valuable in multi-modal graphs.

2.3.3 Unsupervised Graph Neural Networks

Unsupervised graph representation learning aims to encode graph-structured data into low-dimensional embeddings without labeled supervision. Key paradigms include graph autoencoders and contrastive learning, each leveraging different mechanisms to capture structural or semantic patterns.

Graph Autoencoder Graph autoencoders [WCZ16, KW16, PHL+18] are unsupervised models that learn node embeddings by reconstructing graph structural information, such as adjacency matrices, node features, or substructures. They typically consist of an encoder that maps nodes to latent representations and a decoder that reconstructs graph data. A simple example is the GAE model [KW16], which calculates the node embeddings as $\mathbf{Z} = \texttt{GCN}(\mathbf{A}, \mathbf{X})$, and reconstructs the adjacency matrix $\hat{\mathbf{A}} = \sigma(\mathbf{Z}\mathbf{Z}^T)$ by minimizing the mean squared error or the cross entropy loss between $\hat{\mathbf{A}}$ and $\mathbf{A}$. SDNE [WCZ16] is an early graph autoencoder that employs shallow stacked autoencoders to reconstruct adjacency matrices. It preserves first-order proximity (direct edges) via a Laplacian-eigenmaps-inspired loss and second-order proximity (neighborhood similarity) through reconstruction loss. Unlike SDNE, GAE and VGAE [KW16] use graph convolutional networks (GCNs) [KW17] as encoders. GAE reconstructs adjacency matrices using a simple inner-product decoder, and VGAE extends this framework with variational inference, learning probabilistic embeddings that enhance robustness and generalization. Building upon GAE and VGAE, ARGA and ARVGA [PHL+18] further add adversarial regularization by training a discriminator to align latent embeddings with a prior distribution. This strategy improves embedding quality by enforcing distributional constraints. Once learned, embeddings can be directly applied to downstream tasks such as community detection and link prediction. Several studies [WPL+17, ZLZL22, GWY+22, MBTK22] also explore integrating additional task-specific decoders into the autoencoder architecture, enabling end-to-end training for improved performance.

Graph Contrastive Learning Unlike graph autoencoders, contrastive methods learn embeddings by maximizing agreement between augmented views of the same graph [JWQ+24]. These augmentations are broadly categorized into two types: (1) feature or topology modifications, such as feature masking [TTA+21, YCS+20], feature shuffling [JPT21], edge permutation [ZXY+20], and node dropping [XCL+21]; and (2) subgraph sampling strategies [HK20, WZM$^+$22, ZHSS24]. It is important to note that the augmentations should preserve the underlying semantic structure to avoid losing task-relevant information during training. There are distinct training strategies for contrastive learning. DGI [VFH$^+$19] introduces a global-local contrast by contrasting node embeddings against a graph-wide embedding, enforcing nodes to align with the global context. GRACE [ZXY$^+$20] applies edge removal and feature masking to create two views, then maximizes similarity between the same node across views while repelling negative pairs. GraphCL [YCS$^+$20] systematically evaluates the impact of augmentations, showing that edge perturbation and feature masking benefit tasks like molecular property prediction. Instead of conducting augmentation on the whole graph, MVGRL [HK20] employs graph diffusion to sample subgraphs as augmented views and contrasts node features with subgraph representations.

Graph Pretraining In unsupervised graph embedding, another important paradigm is graph pretraining [HLG$^+$20, LJFS21], which derives from pretraining strategies in natural language processing [DYW$^+$19]. Graph pretraining involves training models on auxiliary unsupervised or self-supervised tasks, such as graph reconstruction or contrastive learning, to learn transferable neural network structures. The objective is to capture generalized knowledge that can enhance performance on downstream tasks. After pretraining, the embeddings or model weights are typically used to initialize or fine-tune task-specific models, enabling adaptation to target applications.

2.4 Hypergraph Neural Networks

Hypergraph Neural Networks (HGNNs) extend traditional GNNs to operate on hypergraphs, trying to preserve higher-order relationships depicted by hyperedges. The core design principles of HGNNs align with GNNs [HY21, GFJJ22]. The key differences are: (1) GNNs model node-to-node message passing, while HGNNs need to consider the impact of hyperedges; (2) HGNNs should be able to handle variable-sized hyperedges, necessitating novel aggregation/pooling strategies. Below, we introduce HGNNs by investigating how hyperedges are treated. In general, we divide these methods into implicit and explicit ways.

Implicit Hyperedge Representation Implicit methods encode hyperedges through hypergraph expansions, which reduce hypergraphs to ordinary graphs. HGNN [FYZ$^+$18] leverages hypergraph Laplacian operators to propagate features, treating hyperedges as weighted cliques. HyperGCN [YNY$^+$19] approximates hyperedges as graph edges via a “mediator” node strategy, enabling GCN-like operations. HyperGT [LTY$^+$24] uses the bipartite expansion graph to regularize the attention matrix. HJRL [YCW$^+$24] combines clique expansion and bipartite expansion in order to learn a shared representation of nodes and hyperedges. LEGCN [YWYA22] uses a kind of line expansion to transform node-hyperedge pair as nodes in a standard graph and then apply GCN on it.

Explicit Hyperedge Representation Hyperedges can be represented as explicit learnable entities. Some explicit methods treat hyperedges as dynamic components with their own embeddings, updated during message passing. Typically, these methods can be described by a dual message-passing framework, which consists of node-to-hyperedge aggregation and hyperedge-to-node aggregation. HNHN [DSB20] is a simple dual message passing neural network, which applies separate normalization for each message passing from either node to hyperedge or hyperedge to node, ensuring numerical stability. UniGCNII [HY21] uses a two-stage hyperedge-to-node message passing. Before aggregation, the hyperedge representation is normalized by their degrees. HNHN and UniGCNII assume equal importance of nodes within each hyperedge. Hyper-SAGNN [ZZM20] uses self-attention to model node interactions within hyperedges, learning context-aware contributions of nodes. To cope with the limitation that fixed aggregation functions (e.g., sum, max, mean, etc.) lose structural and feature information, AllSet [CPPM22] replaces fixed aggregation with learnable, permutation-invariant multiset functions. This study also shows that AllSet generalizes existing HGNN models such as HyperGCN [YNY$^+$19], HGNN [FYZ$^+$18], and HyperSAGE [AGRW20] by framing them as special cases with specific aggregation choices. Motivated by the fact that a node in different hyperedges may have different roles, WHATsNet [CKYS23] introduces role-specific node representations by ranking nodes via centrality scores and encoding positional roles within hyperedges, capturing context-dependent heterogeneity. This progression reflects a shift from static aggregation to adaptive, context-aware frameworks for hypergraph representation learning.

2.5 Influence Maximization and Beyond

In this section, we review the classical influence maximization problem with representative solutions and discuss relevant problem settings, including its extension to hypergraphs.

2.5.1 Classical Influence Maximization

The classical influence maximization (IM) problem, a cornerstone of social network analysis, was formalized by Kempe et al. in 2003 [KKT03]. This problem aims to identify a small set of seed nodes in a social network such that, when activated, they trigger the largest possible cascade of influence under a predefined diffusion model. The objective is to maximize the expected number of nodes activated through the spread process. In their study, two kinds of diffusion models are studied:

- **Independent Cascade (IC).** In a discrete-time stochastic process, each newly activated node u has one chance to activate its inactive neighbor v with a probability p_{uv}. Activation attempts are independent across edges and steps. The process terminates when no new activations occur.

- **Linear Threshold (LT).** In LT, each node v has a threshold θ_v, which satisfies $\theta_v \in [0, 1]$, and each edge has a non-negative weight w_{uv} such that $\sum_{u \in N(v)} w_{uv} \leq 1$. Influence weights on edges determine the strength of peer effects. A node becomes active if the cumulative influence from its active neighbors exceeds the threshold.

As shown in [KKT03], IM is NP-hard under IC and LT. Considering the submodularity and monotonicity of the influence spread function, the greedy algorithm, which selects seeds iteratively by choosing the node with the highest marginal gain in influence spread, can be used to solve this problem. This greedy algorithm achieves a $(1 - 1/e)$ approximation ratio relative to the optimal. This pioneering study uses Markov Chain Monte Carlo (MCMC) simulations to calculate the influence spread, as the influence spread can be approximated arbitrarily close by increasing the number of MCMC simulations.

Considering the high computation costs of a large number of MCMC simulations, a line of algorithms tries to reduce the number of MCMC simulations. For instance, CELF [LKG+07] exploits submodularity to reduce redundant calculations. It skips MCMC simulations of vertices that are known to be suboptimal. Another line of algorithms avoids costly MCMC simulations by *trading space for speed*, such as TIM [TSX15], IMM [TXS14], and RIS [BBCL14]. These algorithms sample reverse cascades (denoted as RR sets) to estimate influence efficiently. After sampling sufficient RR sets, finding the seed set becomes a maximum cover problem.

On the other hand, numerous heuristic algorithms are proposed to address the computational challenges of influence maximization (IM) in large social networks, prioritizing practical efficiency over theoretical approximation guarantees. For instance, some studies [CSH+14, LXY+17, ZZ17] consider IM as a node ranking problem, using topology-driven scores to identify high-impact nodes. Another line of research exploits graph structural patterns, such as communities, to reduce computational overhead. By assuming networks exhibit modularity (e.g., communities that are dense inside and with sparse interconnections), these approaches constrain influence propagation to localized regions [CZP+14, GW20] or limit diffusion to a fixed number of steps [GBLV13].

2.5.2 Variations of Influence Maximization

In the last twenty years, influence maximization has evolved to address diverse real-world scenarios and constraints. Below are key variations:

- **Budgeted IM** [NZ13, PHWV20] targets to maximize influence under node-specific activation costs and a total budget.

- **Competitive IM** [BKS07, ZLJW$^+$22] models influence propagation in the presence of competing campaigns, where nodes may adopt one of multiple cascading ideas.

- **Topic-aware IM** [CFL$^+$15, BBM13] targets to maximize influence for content-specific campaigns, where nodes have topic-dependent influence probabilities.

- **Influence Minimization** [YLW$^+$19, YLG20] aims to minimize the spread of misinformation by seeding "protector" nodes.

- **Profit Maximization** [LL12, ZLY$^+$17] maximizes revenue from influenced users, accounting for campaign costs. This combines IM with profit-driven optimization.

- **Diversity-aware IM** [TLZ$^+$14] ensures that seed nodes represent diverse demographics (e.g., age, gender, interests).

Among these variations, Budgeted, Competitive, and Topic-aware IM adapt the core problem to constraints like limited resources, rival campaigns, and content-specific dynamics. Meanwhile, Influence Minimization, Profit Maximization, and Diversity-aware IM reflect broader societal and economic priorities, such as fighting misinformation, making profits, and ensuring fair representation.

It is worth noting that these variations are generally based on ordinary graphs. When accounting for social groups, this necessitates frameworks that consider group-level effects. Below are representative adaptions:

- **SIMPH** [ZZG$^+$18] incorporates crowd psychology, wherein a social group, a tail node, will be activated with some probability only after each node in the head node set is activated. This mechanism captures directed information propagation through group structures based on the IC model.

- **GIM** [ZGW19] simulates election-like dynamics, where a group is activated once a critical fraction of its members are activated. The objective shifts to maximizing the number of activated groups rather than individual users. GIM is built on the IC model.

- **SubTSSH** [ACSS21] extend the LT model to hypergraphs. Nodes influence hyperedges when a sufficient number of their constituent nodes are activated. Hyperedges, in turn, influence nodes connected to them when enough hyperedges are activated.

These advancements extend IM to hypergraphs. They highlight the shift from individual-centric models to systems that account for collective behavior. By bridging classic diffusion models (e.g., IC, LT) with group-aware mechanisms, modern IM research continues to enhance its realism and practical utility. In Section 3.2, we will introduce our problem definition and corresponding solution.

3. Contributions

In this chapter, we introduce the main publications included in this cumulative thesis. The essential contributions will be highlighted. Further details can be found in the corresponding papers in the Appendix.

3.1 Common Properties of the Research Tasks and Methodologies

The three publications address three distinct yet inter-connected tasks: (1) *information diffusion and influence maximization in hypergraphs*, (2) *hyperlink prediction*, and (3) *graph classification*. These tasks shift the analytical focus from traditional node-centric perspectives to higher-order relational abstractions. Specifically, the first two tasks aim to identify highly influential or potentially existing hyperedges, respectively, while the third task classifies entire graph objects. Among these, the first task introduces a novel research problem, which generalizes traditional influence maximization problems to hypergraphs; the latter two have been explored in previous studies. For the latter two tasks, hyperlink prediction has gained increasing attention in recent years (e.g., [PSL$^+$21, ZCJC18, SZK21], to name a few, more related studies can be found in this survey [CL23]), and graph classification benefits from numerous well-established methodologies (e.g., spectral algorithms [SZA21, TMK$^+$18, VZ17], graph kernels [VSKB10, YV15, KJM20] and graph neural networks [ZCNC18, LRK18]). The three tasks present unique computational and theoretical challenges due to the following reasons:

- **Combinatorial Complexity**: The number of potential connections in graphs or hypergraphs grows quadratically or exponentially with network size. This complexity renders tasks such as (a) graph similarity learning [GXTL10], i.e., comparing the similarity between pairs of graphs, which serves as the foundation for graph classification; and (b) hyperlink prediction, i.e., predicting missing hyperedges, which is computationally demanding since it has a vast searching space[4].

- **Higher-order Relational Modeling**: Hypergraphs capture more generalized relationships than graphs, necessitating specialized techniques. For instance, (a) information diffusion in hypergraphs [SGS18] should account for group dynamics, and (b) centrality analysis [Ben19] faces limitations as many graph-theoretic principles do not directly generalize to hypergraphs.

- **Interpretability Constraints**: Despite the success of graph neural networks, their complex architectures often make their decision-making processes less transparent. This lack of transparency raises concerns in high-stakes domains like healthcare and risk management.

Motivated by these challenges, we aims to develop **scalable** and **interpretable** solutions for the three tasks, which forms the general research goal of this thesis. In the first task, we introduce a novel influence maximization problem in hypergraphs: selecting a seed set of hyperedges to maximize the number of informed nodes under a hypergraph-specific information diffusion model. To address this problem, we formalize a new diffusion model (HIC) for hypergraphs considering group-level information propagation. Based on HIC, we can identify high-influential hyperedges from an information diffusion perspective. We prove that the defined problem is NP-hard. To solve it, we propose InfDis, a heuristic algorithm that balances efficiency and effectiveness in solution discovery. For the remaining tasks, we propose two unsupervised structural feature extraction approaches: SSF and RWF. SSF is the first subgraph-based feature extraction technique for hyperlink prediction. It offers fine-grained characterization of hyperedges and their neighborhoods through localized topological patterns. RWF employs structural-role-based vertex partitioning and quantifies intra- and inter-partition connection strengths as predictive features. Notably, the extracted features of both algorithms are rooted in well-established structural heuristics such as loops and walks, ensuring interpretability. Both methods avoid reliance on labeled data, making them broadly applicable. We show the scalability of the proposed methods by comparing them to a wide range of state-of-the-arts. Next, we present the technical contributions and advantages of each approach in detail.

[4] To make this problem feasible, current mainstream algorithms [SZK21, YNN$^+$20, XRK13, ZCJC18, PSL$^+$21] reformulate prediction as a binary classification problem, by introducing a candidate set. Our paper [LPL$^+$23] follows this way. We also note that this fashion has been challenged in [YLHS24].

3.2 Information Diffusion and Influence Maximization in Hypergraphs

As introduced in Section 2.5, influence maximization (IM) is a well-established research topic with many different problem settings. Traditional IM aims to identify a seed set of nodes that maximize information spread under predefined diffusion dynamics. While existing works predominantly focus on node-centric propagation in graphs, we extend this paradigm to hypergraphs (i.e., social networks with group structure), where hyperedges (groups) enable simultaneous information sharing among multiple nodes.

In the first paper [LWLB23], we propose a novel influence maximization problem: *Information Coverage Maximization in Hypergraphs*. Specifically, the target is to maximize information coverage in hypergraphs, by selecting seed hyperedges (groups) and following a predefined hypergraph information diffusion model (i.e., the Hypergraph Independent Cascade model (HIC)). We present the Influence Discount (InfDis) algorithm to address this problem. Our contributions are threefold:

- **A Novel Problem**. We formulate a new influence maximization problem. The HIC model is central to the problem definition, which generalizes the classical IC model to hypergraphs. In HIC, hyperedges (not nodes) act as diffusion units: a hyperedge activates if selected as a seed or activated by an overlapping hyperedge, with propagation probability based on shared nodes. HIC mirrors real-world behaviors in online communities, where every social group member has the same information exposure. Activating a hyperedge means to inform all its nodes, and overlapping groups enable cascading effects. This group-centric mechanism shifts the optimization target from selecting seed nodes with high authority to selecting a set of seed hyperedges, which can strongly affect information propagation.

- **Theoretical Foundation**. We establish the theoretical foundation of the problem. We prove NP-hardness via reduction to the maximum coverage problem and show that computing the exact information coverage is #P-hard. Additionally, we demonstrate the submodularity and monotonicity of the coverage function, enabling greedy approaches with approximation guarantees. However, traditional greedy methods suffer from high computational costs.

- **A Practical Solution**. We design InfDis, a heuristic inspired by Degree Discount, to efficiently approximate information coverage. InfDis iteratively selects hyperedges with the highest estimated marginal gain in coverage while discounting the influence of hyperedges that overlap with the selected seeds. By restricting coverage estimation to immediate neighbors (under small propagation probabilities), InfDis avoids costly simulations and achieves near-linear time complexity. Specifically, InfDis includes the following two steps.

- Direct/Indirect Coverage Estimation: Quantifies new nodes covered by a hyperedge and its activated neighbors.
- Influence Discounting: Adjusts the coverage of neighboring hyperedges after seed selection to avoid redundancy.

Experiments on nine real-world datasets demonstrate InfDis's superiority. It outperforms baseline heuristics (Degree, Betweenness) and matches the accuracy of greedy methods while being 10,000× faster. It processes the Stackoverflow graph with 700k nodes and 69k hyperedges in 0.139 seconds. InfDis also adapts well to varying propagation probabilities and hypergraph densities, showcasing robustness.

3.3 Subgraph-based Hyperlink Prediction

The second paper [LPL+23] addresses hyperlink prediction, which aims to predict missing hyperedges in hypergraphs. Existing methods (e.g., FamilySet [SZK21], NHP [YNN+20], Loop [PSL+21], HPLSF [XRK13], CMM [ZCJC18]), often operate on the entire hypergraph, leading to scalability challenges. To overcome this, we propose SSF (Subgraph Structural Features), a novel method that extracts interpretable subgraph features for hyperlink prediction. The core of SSF is that we measure the connection strength among subgraph nodes when the edges expanded by the focal hyperedge are gradually weakened.

SSF has four steps: (1) subgraph extraction, (2) edge weakening for generating subgraph variations, (3) feature extraction, and (4) classification. In the subgraph extraction step, SSF extracts localized subgraphs around candidate hyperedges, which is similar to the subgraph extraction process of SEAL [ZC18], with the only difference being that one is hypergraph and one is graph. After the subgraph for a hyperedge f is extracted, we use clique expansion to get the matrix representation of the subgraph. The edge weight $\boldsymbol{w}_{uv}$ signifies the number of hyperedges that u and v belong to simultaneously, and we mark the edges expanded by the candidate hyperedge as focal edges. In the edge weakening process, we gradually weaken the edge weights of focal edges, using a parameter α to control the degree of edge weakening. Consequently, we can select a finite number of discrete states of the edge weakening process, creating subgraph variations. For example:

1. Decrease the weights of the focal edges by 1, indicating the removal of the candidate hyperedge from the subgraph.

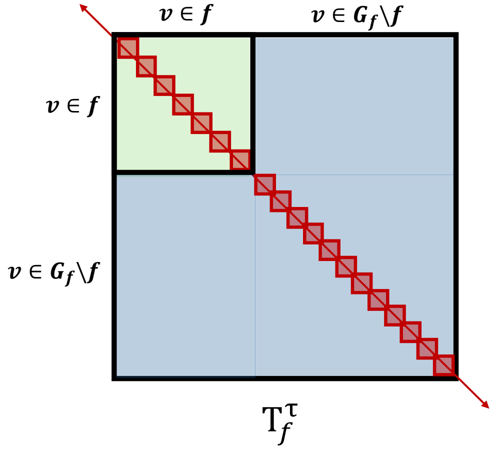


***Figure 3.1:** **Illustration of the Features Extracted by SSF:** The loop feature $\texttt{loop}_{f,\tau}$ is the average of diagonal entries inside the green square. The walk feature $\texttt{walk}_{f,\tau}$ is the average of all entries inside the green square. The subgraph-level loop perturbation $\Delta\texttt{total\_loop}_{f,\tau}$ represents the change in the average of the diagonal entries across the entire matrix compared to the initial relational matrix $\mathbf{T}_{f,initial}^{\tau}$.*

2. Decrease the weights of the focal edges to 0, indicating the complete removal of the edges expanded by the candidate hyperedge from the subgraph, i.e., these edges are blocked.

We then get a set of transition/relational matrices corresponding to the subgraphs: $\{\mathbf{T}_{f,initial}, \mathbf{T}_{f,\alpha_1}, \mathbf{T}_{f,\alpha_2}, \cdots, \mathbf{T}_{f,\alpha_{t-1}}\}$[5]. The next step is feature extraction. We extract three kinds of features for each matrix:

$$\texttt{loop}_{f,\tau} = \frac{1}{|f|}\sum_{v\in f}[\mathbf{T}_f^\tau]_{vv}$$

$$\texttt{walk}_{f,\tau} = \frac{1}{|f|^2}\sum_{u,v\in f}[\mathbf{T}_f^\tau]_{uv}$$

$$\Delta\texttt{total\_loop}_{f,\tau} = \frac{1}{|\mathcal{V}_f^h|}\left(\text{tr}([\mathbf{T}_{f,initial}^\tau]) - \text{tr}([\mathbf{T}_f^\tau])\right)$$

where $\mathbf{T}_f^\tau$ captures τ-hop connection strengths. After feature extraction, these features are combined into a vector and classified using an MLP, enabling inductive prediction even for unseen nodes.

In summary, the main technical contributions of SSF are:

[5] The transition matrix is calculated by $\mathbf{D}^{-1}\mathbf{A}$, while the relational matrix is calculated by $\mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$. Both matrices can reflect the connection strengths between pairs of nodes. In our published work, we use the relational matrices because symmetric matrices have computational advantages, such as faster eigenvalue decomposition and improved numerical stability during iterative processes.

- **Interpretable Feature Extraction:** Loop and walk features capture local connectivity patterns. Subgraph-level loop perturbations measure structural changes caused by weakening edges. The three features can also be explained from the perspective of eigen-decomposition.
- **Adaptive Edge Weakening:** The edge weakening mechanism dynamically adjusts edge weights of the focal edges of the subgraph, allowing SSF to generalize across dense and sparse hypergraphs. Compared to subgraph-based link prediction techniques [ZC18, CJ20, WZ21], which relies on labeling tricks [ZLX$^+$21] to distinguish focal nodes and other subgraph nodes, SSF uses a different way, by gradually weaken the weights of focal edges. In this way, SSF encodes fine-grained topological fingerprints that relate to the existence of candidate hyperedge. From another perspective, the proposed edge weakening mechanism provides a new way of keeping higher-order information when using hypergraph expansion techniques.

Experiments on 10 real-world datasets demonstrate SSF's effectiveness over state-of-the-art baselines in AUC and Precision. Ablation studies confirm the importance of each feature component, while parameter analysis highlights robustness. The edge weakening scheme is shown to enhance adaptability, particularly in sparse hypergraphs where traditional methods fail. The scalability of SSF is shown by its ability to process subgraphs, not whole graphs. Excluding subgraph extraction, the complexity for computing features for a hyperedge is $O(t \cdot \tau \cdot |\mathcal{V}_f^h|^3)$, which is at the same level of various subgraph GNNs. The difference is that we do not enforce end-to-end training but rely on single-pass feature extraction, and the extracted features are ready to feed into an MLP. Thus, SSF requires less time and computational resources. In summary, by focusing on local subgraphs and interpretable structural features, SSF achieves state-of-the-art in multiple hyperlink prediction tasks while maintaining scalability and transparency.

3.4 Feature Extraction for Graph Classification

The third paper [LWB24] addresses graph classification. Graph classification faces scalability challenges due to the increasing size of real-world graphs and the large volume of graph datasets. In the literature, there are two main lines of approaches: graph kernels [BK05, ZWX$^+$18] and graph neural networks [HYL17, HY21]. While graph kernels offer interpretability, they struggle with large datasets due to quadratic complexity in the number of graphs. GNNs, though powerful, require heavy computational resources and lack interpretability if considering complex architectures. To address these limitations, we propose Random Walk Fingerprints (RWF), a scalable, unsupervised, and interpretable structural feature extraction method for graph classification. The main technical contributions are twofold:

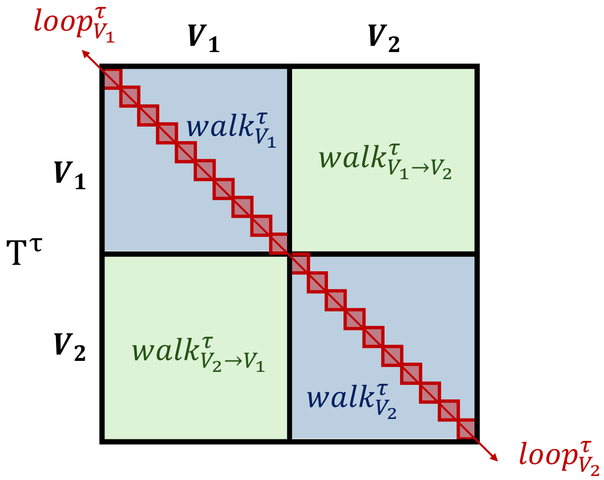
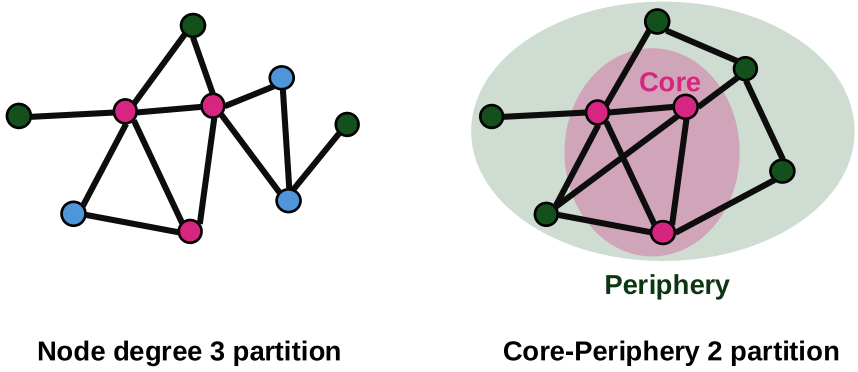


Figure 3.2: Illustration of the Features Extracted by RWF: *The loop feature of a node subset V_1 is $\texttt{loop}^{\tau}_{V_1}$, denotes the average of diagonal entries inside the blue square of V_1. The walk feature of a node subset V_1 is $\texttt{walk}^{\tau}_{V_1}$, which denotes the average of all entries inside the blue square corresponding to V_1. The walk feature between two node subsets, i.e., $\texttt{walk}^{\tau}_{V_1 \to V_2}$ and $\texttt{walk}^{\tau}_{V_2 \to V_1}$, denotes the average of all entries inside the green square correspondingly.*

- **Novel Graph Fingerprinting Technique:** We introduce scalable and interpretable structural features based on loop and walk. The basis of these structural features is the same as the second paper [LPL$^+$23]. The difference of RWF is that we measure loop and walk between node subsets, as shown in Figure 3.2. In other words, these features quantify connection strengths within and between node subsets, capturing local connectivity patterns. By leveraging sparse matrix operations and incremental walk-step calculations, RWF achieves a time complexity of $O(N\tau_c mn)$, making it suitable for large graph datasets and sparse graphs. Besides, RWF is capable of assimilating node features.

- **Vertex Partitioning Heuristics:** We propose two vertex partitioning heuristics to align node subsets across graphs of varying sizes. Unlike community detection methods that focus on identifying dense subgraphs characterized by high internal density and sparse interconnections, our degree-based and core-periphery heuristics partition nodes by structural roles. As shown in Figure 3.3, the degree heuristic sorts nodes by degree and divides them into k subsets, while the core-periphery heuristic identifies central and peripheral nodes using the KM-config algorithm [KM18]. These partitions enable consistent comparisons of local connectivity patterns, enhancing discriminative power. Experiments show that the degree heuristic outperforms alternatives, balancing simplicity and effectiveness.

We conduct comprehensive empirical validation across 12 real-world datasets, includ-



Node degree 3 partition

Core-Periphery 2 partition

Figure 3.3: ***Illustration of Two Vertex Partitioning Heuristics:*** *The left figure characterizes degree-based partitioning, where the nodes are sorted by their degree and then divided into three partitions. The right figure depicts the core-periphery heuristic, where the KM-config algorithm identifies core nodes and periphery nodes [KM18].*

ing social networks and molecular graphs. RWF achieves state-of-the-art accuracy on $8/12$ datasets and demonstrates competitive runtime performance. RWF's design avoids the need for expensive kernel matrix computations or end-to-end training, thus simplifying deployment. While parameters like the number of partitions and maximum walk steps require tuning, our experiments show that RWF is stable with a small number of partitions, such as 2 or 3, and RWF is not sensitive to the walk length. The method's scalability is validated on datasets with up to 127,094 graphs, highlighting its practicality for real-world applications.

4. Conclusion

In this thesis, we presented multiple advances in the area of graph mining and hypergraph mining, in particular in the tasks of information coverage maximization in hypergraphs, hyperlink prediction and graph classification. In the following, we discuss relevant concepts, and show the limitations of proposed methodologies. Then, we introduce potential research opportunities arising from these researches, and provide final remarks.

4.1 Discussion

Model Simplicity vs. Real-world Complexity. The tension between model simplicity and real-world complexity is a pervasive challenge in applied research [MS24]. While simplified models, such as the proposed HIC model, enable tractable analysis and algorithmic guarantees, they inevitably lose nuances inherent to real-world social dynamics. Complex models, on the other hand, risk becoming analytically intractable or computationally prohibitive. Model simplicity and real-world complexity are like a coin of two sides. Such limitations highlight a recurring theme in applied science: balancing simplicity to achieve actionable insights while acknowledging the messiness of reality.

Deep Approaches vs. Non-deep Approaches The rise of deep learning over the past decade has reshaped computational paradigms across disciplines, achieving the best performances in tasks ranging from image classification [KSH17] to protein structure prediction [JEP$^+$21]. Its ability to autonomously learn hierarchical representations from raw data is a cornerstone of modern artificial intelligence. However, this success raises a critical question:

Is deep learning a universal solution?

While powerful, deep learning techniques typically require high computational resources, and they introduce trade-offs, particularly between performance and interpretability [CSS23]. Though hardware and distributed computing advances have enabled the training of increasingly larger models, the computational overhead remains a barrier for resource-constrained institutions. Furthermore, the black-box nature of these systems limits trust in high-stakes domains. This thesis argues that algorithm selection must align with the concrete goal. We contributed three non-deep approaches tailored to the three tasks. Indeed, these tasks could also employ deep learning techniques. There are GNNs for estimating the influence spread in ordinary graphs [KMKP22, PTVM23, LJW+23, PTV+24]. Hyperlink prediction and graph classification are the domains that deep learning has been widely explored, such as various HGNNs [SZK21, RZSW24, YNN+20] and GNNs [ZCNC18, LRK18, BZR21, HJLH22]. These solutions often impose prohibitive computational complexity and resource demands. We benchmark against various neural architectures in the latter two tasks. We show that through proper algorithmic design, our non-deep methods achieve competitive or superior performances in many cases. These results demonstrate that effectiveness need not correlate with architectural complexity, and scalability, interpretability, and effectiveness can be achieved simultaneously.

4.2 Limitations

Simplified Problem Setting In the above discussion, we mentioned that the simplicity of the HIC model introduces limitations. SSF has a similar issue. It assumes a predefined candidate set of potential hyperedges, aligning with the mainstream solutions. However, generating this set remains a non-trivial problem. Current approaches rely on negative sampling or domain knowledge, which may introduce bias or fail to cover meaningful candidates, potentially impacting prediction reliability.

Feature Design Constraints SSF and RWF rely on handcrafted structural features (e.g., walks, loops), which prioritize interpretability and scalability. While effective in many cases, their performance could degrade if the chosen heuristics misalign with the underlying network formation mechanisms. This family of algorithms may inherently constrain the representational power of extracted features, as performance is bounded by the suitability of manually engineered heuristics. Currently, there is no theoretical framework for analyzing the representational power of these handcrafted features.

Unaddressed Theoretical Aspects In the study of maximizing information coverage in hypergraphs, the NP-hardness proof relies on a reduction from the maximum coverage problem under a restricted scenario where the propagation probability $p = 0$. While this reduction validly establishes NP-hardness, it does not explicitly address the problem's complexity in general cases where $p > 0$. Specifically, the theoretical relationship between the maximum coverage problem and the proposed problem remains unexplored when propagation dynamics enable seed hyperedges to activate additional hyperedges beyond the initial selection. This gap leaves open whether existing maximum coverage algorithms could serve as heuristics for practical scenarios where $p > 0$. Furthermore, the proposed InfDis heuristic prioritizes empirical efficiency over theoretical guarantees. Though it performs well in experiments, the lack of an approximation guarantee limits its theoretical reliability in worst-case scenarios.

4.3 Future Work

Learning Propagation Probabilities from Data While influence maximization algorithms excel at identifying optimal seed nodes under assumed propagation probabilities, they typically treat these probabilities as abstract parameters rather than empirically grounded values. This is the critical limitation of current influence maximization research, which prioritizes algorithmic efficiency over modeling actual diffusion processes. By deriving these probabilities from observed cascades [6], bridging this gap opens rich research opportunities. By analyzing real-world diffusion patterns (e.g., retweets, citation cascades, or viral content trajectories), we might be able to quantify node-specific influence in diverse contexts, identify universal or platform-specific diffusion mechanisms, and develop predictive models grounded in empirical behavior. Such efforts would advance theoretical frameworks and enhance practical strategies for managing misinformation, optimizing marketing campaigns, or modeling social contagion.

Temporal Effects in Hyperlink Prediction Current hyperlink prediction algorithms mainly focus on identifying missing hyperedges in static hypergraphs. The problem of forecasting future hyperlinks remains under-explored. The key challenges of temporal hyperlink prediction lie in adapting temporal dynamics (e.g., recency, periodicity, drift, burstiness) to multi-node interactions [IKB24]. While hyperedges inherently encode higher-order dependencies, their temporal constraints frequently arise from overlapping pairwise interactions. This observation suggests potential alignment with frameworks like neural relational inference (NRI) [KFW⁺18, XLN⁺22], which infers latent relational structures between entities. By extending NRI's relational reasoning to group-level temporal coordination, we may be able to bridge temporal modeling and hyperedge forecasting, enabling joint prediction of when and why multi-node interactions emerge.

Sub-structural Learning for Graphs In RWF [LWB24], we explored role-based structural partitioning methods beyond Degree and Core-Periphery, like structural clustering

[6] Like quantifying epidemiology's reproduction number [BIH⁺24], propagation probabilities could be estimated through an event-driven lens.

techniques. A key limitation of these approaches lies in their strong dependence on parameter tuning, which becomes impractical for a large number of graphs with different sub-structural compositions. To mitigate this problem, we may explore an unsupervised parameter-free decomposition strategy that breaks graphs into universal and fundamental blocks, enabling shared substructures to generalize across diverse graph types. These substrcutures can then be explicitly used in graph-related downstream tasks.

4.4 Final Remarks

For this thesis, we proposed three data mining techniques for graph and hypergraph analysis. From a technical perspective, this thesis prioritizes practicality over algorithmic novelty. The results support that scalable and interpretable solutions can achieve substantial performance. Overall, the contributed techniques represent a small step within the rapidly expanding field of scientific research. We hope these techniques could help domain experts to uncover novel patterns in their data while inspiring new directions for the research community.

References

[ACP^{+}23] Alessia Antelmi, Gennaro Cordasco, Mirko Polato, Vittorio Scarano, Carmine Spagnuolo, and Dingqi Yang. A survey on hypergraph representation learning. *ACM Computing Surveys*, 56(1):1–38, 2023.

[ACSS21] Alessia Antelmi, Gennaro Cordasco, Carmine Spagnuolo, and Przemysław Szufel. Social influence maximization in hypergraphs. *Entropy*, 23(7):796, 2021.

[AE21] Nada Alzaben and Daniel W Engels. End-to-end routing in sdn controllers using max-flow min-cut route selection algorithm. In *2021 23rd International Conference on Advanced Communication Technology (ICACT)*, pages 461–467. IEEE, 2021.

[AGRW20] Devanshu Arya, Deepak K Gupta, Stevan Rudinac, and Marcel Worring. Hypersage: Generalizing inductive representation learning on hypergraphs. *arXiv preprint arXiv:2010.04558*, 2020.

[ALH22] Jibouni Ayoub, Dounia Lotfi, and Ahmed Hammouch. Link prediction using betweenness centrality and graph neural networks. *Social Network Analysis and Mining*, 13(1):5, 2022.

[Alt80] Wolgang Alt. Biased random walk models for chemotaxis and related diffusion approximations. *Journal of mathematical biology*, 9:147–177, 1980.

[AMB19] Fernando Ascensão, Frederico Mestre, and A Márcia Barbosa. Prioritizing road defragmentation using graph-based tools. *Landscape and Urban Planning*, 192:103653, 2019.

[BAS^{+}18] Austin R Benson, Rediet Abebe, Michael T Schaub, Ali Jadbabaie, and Jon Kleinberg. Simplicial closure and higher-order link prediction. *Proceedings of the National Academy of Sciences*, 115(48):E11221–E11230, 2018.

[BBCL14] Christian Borgs, Michael Brautbar, Jennifer Chayes, and Brendan Lucier. Maximizing social influence in nearly optimal time. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*, pages 946–957. SIAM, 2014.

[BBM13] Nicola Barbieri, Francesco Bonchi, and Giuseppe Manco. Topic-aware social influence propagation models. *Knowledge and information systems*, 37:555–584, 2013.

[Ben19] Austin R Benson. Three hypergraph eigenvector centralities. *SIAM Journal on Mathematics of Data Science*, 1(2):293–312, 2019.

[BGLA21] Filippo Maria Bianchi, Daniele Grattarola, Lorenzo Livi, and Cesare Alippi. Graph neural networks with convolutional arma filters. *IEEE transactions on pattern analysis and machine intelligence*, 44(7):3496–3507, 2021.

[BGS05] Monica Bianchini, Marco Gori, and Franco Scarselli. Inside pagerank. *ACM Transactions on Internet Technology (TOIT)*, 5(1):92–128, 2005.

[BHG$^+$21] Muhammet Balcilar, Pierre Héroux, Benoit Gauzere, Pascal Vasseur, Sébastien Adam, and Paul Honeine. Breaking the limits of message passing graph neural networks. In *International Conference on Machine Learning*, pages 599–608. PMLR, 2021.

[BIH$^+$24] Nicola Luigi Bragazzi, Sarafa Adewale Iyaniwura, Qing Han, Woldegebriel Assefa Woldegerima, and Jude Dzevela Kong. Quantifying the basic reproduction number and underestimated fraction of mpox cases worldwide at the onset of the outbreak. *Journal of the Royal Society Interface*, 21(216):20230637, 2024.

[BK05] Karsten M Borgwardt and Hans-Peter Kriegel. Shortest-path kernels on graphs. In *Fifth IEEE international conference on data mining (ICDM'05)*, pages 8–pp. IEEE, 2005.

[BKS07] Shishir Bharathi, David Kempe, and Mahyar Salek. Competitive influence maximization in social networks. In *Internet and Network Economics: Third International Workshop, WINE 2007, San Diego, CA, USA, December 12-14, 2007. Proceedings 3*, pages 306–311. Springer, 2007.

[BN03] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396, 2003.

[BO22] Amirhosein Bodaghi and Jonice Oliveira. The theater of fake news spreading, who plays which role? a study on real graphs of spreading on twitter. *Expert Systems with Applications*, 189:116110, 2022.

[Bol81] Béla Bollobás. The diameter of random graphs. *Transactions of the American Mathematical Society*, 267(1):41–52, 1981.

[BRH$^+$21] Muhammet Balcilar, Guillaume Renton, Pierre Héroux, Benoit Gaüzère, Sébastien Adam, and Paul Honeine. Analyzing the expressive power of graph neural networks in a spectral perspective. In *International Conference on Learning Representations*, 2021.

[BZR21] Beatrice Bevilacqua, Yangze Zhou, and Bruno Ribeiro. Size-invariant graph representations for graph classification extrapolations. In *International Conference on Machine Learning*, pages 837–851. PMLR, 2021.

[CBHG12] Meeyoung Cha, Fabrício Benevenuto, Hamed Haddadi, and Krishna Gummadi. The world of connections and information flow in twitter. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 42(4):991–998, 2012.

[CF06] Deepayan Chakrabarti and Christos Faloutsos. Graph mining: Laws, generators, and algorithms. *ACM computing surveys (CSUR)*, 38(1):2–es, 2006.

[CFL$^+$15] Shuo Chen, Ju Fan, Guoliang Li, Jianhua Feng, Kian-lee Tan, and Jinhui Tang. Online topic-aware influence maximization. *Proceedings of the VLDB Endowment*, 8(6):666–677, 2015.

[CH06] Diane J Cook and Lawrence B Holder. *Mining graph data*. John Wiley & Sons, 2006.

[CHYW23] Chen Cai, Truong Son Hy, Rose Yu, and Yusu Wang. On the connection between mpnn and graph transformer. In *International conference on machine learning*, pages 3408–3430. PMLR, 2023.

[CJ20] Lei Cai and Shuiwang Ji. A multi-scale approach for graph link prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 3308–3315, 2020.

[CKYS23] Minyoung Choe, Sunwoo Kim, Jaemin Yoo, and Kijung Shin. Classification of edge-dependent labels of nodes in hypergraphs. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 298–309, 2023.

[CL23] Can Chen and Yang-Yu Liu. A survey on hyperlink prediction. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[CLZK18] Xi Chen, Yiqun Liu, Liang Zhang, and Krishnaram Kenthapadi. How linkedin economic graph bonds information and product: applications in linkedin salary. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 120–129, 2018.

[COZ$^+$17] Themistoklis Charalambous, Ayca Ozcelikkale, Mario Zanon, Paolo Falcone, and Henk Wymeersch. On the resource allocation problem in wireless networked control systems. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 4147–4154. IEEE, 2017.

[CPPM22] Eli Chien, Chao Pan, Jianhao Peng, and Olgica Milenkovic. You are allset: A multiset function framework for hypergraph neural networks. In *International Conference on Learning Representations*, 2022.

[CR19] Uthsav Chitra and Benjamin Raphael. Random walks on hypergraphs with edge-dependent vertex weights. In *International conference on machine learning*, pages 1172–1181. PMLR, 2019.

[CRS97] Dragoš M Cvetković, Peter Rowlinson, and Slobodan Simic. *Eigenspaces of graphs*. Number 66. Cambridge University Press, 1997.

[CSH$^+$14] Suqi Cheng, Huawei Shen, Junming Huang, Wei Chen, and Xueqi Cheng. Imrank: influence maximization via finding self-consistent ranking. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 475–484, 2014.

[CSS23] Barnaby Crook, Maximilian Schlüter, and Timo Speith. Revisiting the performance-explainability trade-off in explainable artificial intelligence (xai). In *2023 IEEE 31st International Requirements Engineering Conference Workshops (REW)*, pages 316–324. IEEE, 2023.

[CSX12] Yudong Chen, Sujay Sanghavi, and Huan Xu. Clustering sparse graphs. *Advances in neural information processing systems*, 25, 2012.

[CZP$^+$14] Yi-Cheng Chen, Wen-Yuan Zhu, Wen-Chih Peng, Wang-Chien Lee, and Suh-Yin Lee. Cim: Community-based influence maximization in social networks. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 5(2):1–31, 2014.

[DB21] Vijay Prakash Dwivedi and Xavier Bresson. A generalization of transformer networks to graphs. *AAAI Workshop on Deep Learning on Graphs: Methods and Applications*, 2021.

[DBV16] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems*, 29, 2016.

[DLL$^+$22] Vijay Prakash Dwivedi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Graph neural networks with learnable structural and positional representations. In *International Conference on Learning Representations*, 2022.

[DSB20] Yihe Dong, Will Sawin, and Yoshua Bengio. Hnhn: Hypergraph networks with hyperedge neurons. *ICML Graph Representation Learning and Beyond Workshop*, 2020.

[DYW$^+$19] Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. Unified language model pretraining for natural language understanding and generation. *Advances in neural information processing systems*, 32, 2019.

[EASDP20] Negin Entezari, Saba A Al-Sayouri, Amirali Darvishzadeh, and Evangelos E Papalexakis. All you need is low (rank) defending against adversarial attacks on graphs. In *Proceedings of the 13th international conference on web search and data mining*, pages 169–177, 2020.

[FB15] Lorenz Fischer and Abraham Bernstein. Workload scheduling in distributed stream processors using graph partitioning. In *2015 IEEE International Conference on Big Data (Big Data)*, pages 124–133. IEEE, 2015.

[FBCM04] Charless Fowlkes, Serge Belongie, Fan Chung, and Jitendra Malik. Spectral grouping using the nystrom method. *IEEE transactions on pattern analysis and machine intelligence*, 26(2):214–225, 2004.

[FF09] Agata Fronczak and Piotr Fronczak. Biased random walks in complex networks: The role of local navigation rules. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, 80(1):016107, 2009.

[FPSS96] Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. From data mining to knowledge discovery in databases. *AI magazine*, 17(3):37–37, 1996.

[Fre77] Linton C Freeman. A set of measures of centrality based on betweenness. *Sociometry*, pages 35–41, 1977.

[FYZ$^+$18] Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. Hypergraph neural networks. *AAAI 2019*, 2018.

[FYZ$^+$19] Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. Hypergraph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 3558–3565, 2019.

[GBLV13] Amit Goyal, Francesco Bonchi, Laks VS Lakshmanan, and Suresh Venkatasubramanian. On minimizing budget and time in influence propagation over social networks. *Social network analysis and mining*, 3:179–192, 2013.

[GFJJ22] Yue Gao, Yifan Feng, Shuyi Ji, and Rongrong Ji. Hgnn+: General hypergraph neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):3181–3199, 2022.

[GL16] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864, 2016.

[GMP21] Floris Geerts, Filip Mazowiecki, and Guillermo Perez. Let's agree to degree: Comparing graph convolutional networks in the message-passing framework. In *International Conference on Machine Learning*, pages 3640–3649. PMLR, 2021.

[GSC16] Naveen Gupta, Anurag Singh, and Hocine Cherifi. Centrality measures for networks with community structure. *Physica A: Statistical Mechanics and its Applications*, 452:46–59, 2016.

[GSR$^+$17] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pages 1263–1272. PMLR, 2017.

[GW20] Jianxiong Guo and Weili Wu. Influence maximization: Seeding based on community structure. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 14(6):1–22, 2020.

[GWY$^+$22] Zhihao Guo, Feng Wang, Kaixuan Yao, Jiye Liang, and Zhiqiang Wang. Multi-scale variational graph autoencoder for link prediction. In *Proceedings of the Fifteenth ACM international conference on web search and data mining*, pages 334–342, 2022.

[GXTL10] Xinbo Gao, Bing Xiao, Dacheng Tao, and Xuelong Li. A survey of graph edit distance. *Pattern Analysis and applications*, 13:113–129, 2010.

[HAPP20] Koby Hayashi, Sinan G Aksoy, Cheong Hee Park, and Haesun Park. Hypergraph random walks, laplacians, and clustering. In *Proceedings of the 29th acm international conference on information & knowledge management*, pages 495–504, 2020.

[Hav99] Taher Haveliwala. Efficient computation of pagerank. Technical report, Stanford, 1999.

[HCD12] Cho-Jui Hsieh, Kai-Yang Chiang, and Inderjit S Dhillon. Low rank modeling of signed networks. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 507–515, 2012.

[HH97] NA Hill and D-P Häder. A biased random walk model for the trajectories of swimming micro-organisms. *Journal of theoretical biology*, 186(4):503–526, 1997.

[HJLH22] Xiaotian Han, Zhimeng Jiang, Ninghao Liu, and Xia Hu. G-mixup: Graph data augmentation for graph classification. In *International Conference on Machine Learning*, pages 8230–8248. PMLR, 2022.

[HK20] Kaveh Hassani and Amir Hosein Khasahmadi. Contrastive multi-view representation learning on graphs. In *Proceedings of International Conference on Machine Learning*, pages 3451–3461, 2020.

[HLG$^+$19] Chen Huang, Peiyan Li, Chongming Gao, Qinli Yang, and Junming Shao. Online budgeted least squares with unlabeled data. In *2019 IEEE International Conference on Data Mining (ICDM)*, pages 309–318. IEEE, 2019.

[HLG$^+$20] Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. Strategies for pre-training graph neural networks. In *International Conference on Learning Representations*, 2020.

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[HTK14] Shih-Wen Huang, Daniel Tunkelang, and Karrie Karahalios. The role of network distance in linkedin people search. In *Proceedings of the 37th international acm sigir conference on research & development in information retrieval*, pages 867–870, 2014.

[HY21] Jing Huang and Jie Yang. Unignn: a unified framework for graph and hypergraph neural networks. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, 2021.

[HYL17] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

[IKB24] Iacopo Iacopini, Márton Karsai, and Alain Barrat. The temporal dynamics of group interactions in higher-order social networks. *Nature Communications*, 15(1):7391, 2024.

[JEP+21] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *nature*, 596(7873):583–589, 2021.

[Jin21] Wei Jin. Graph mining with graph neural networks. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 1119–1120, 2021.

[JPT21] Baoyu Jing, Chanyoung Park, and Hanghang Tong. Hdmi: High-order deep multiplex infomax. In *Proceedings of the web conference 2021*, pages 2414–2424, 2021.

[JWQ+24] WEI JU, YIFAN WANG, YIFANG QIN, ZHENGYANG MAO, ZHIPING XIAO, JUNYU LUO, JUNWEI YANG, YIYANG GU, DONGJIE WANG, QINGQING LONG, et al. Towards graph contrastive learning: A survey and beyond. *J. ACM*, 1(1), 2024.

[JYJ+21] Di Jin, Zhizhi Yu, Pengfei Jiao, Shirui Pan, Dongxiao He, Jia Wu, S Yu Philip, and Weixiong Zhang. A survey of community detection approaches: From statistical modeling to deep learning. *IEEE Transactions on Knowledge and Data Engineering*, 35(2):1149–1170, 2021.

[KBH+21] Devin Kreuzer, Dominique Beaini, Will Hamilton, Vincent Létourneau, and Prudencio Tossou. Rethinking graph transformers with spectral attention. *Advances in Neural Information Processing Systems*, 34:21618–21629, 2021.

[KDP12] Da Kuang, Chris Ding, and Haesun Park. Symmetric nonnegative matrix factorization for graph clustering. In *Proceedings of the 2012 SIAM international conference on data mining*, pages 106–117. SIAM, 2012.

[KFW$^+$18] Thomas Kipf, Ethan Fetaya, Kuan-Chieh Wang, Max Welling, and Richard Zemel. Neural relational inference for interacting systems. In *International conference on machine learning*, pages 2688–2697. Pmlr, 2018.

[KJM20] Nils M Kriege, Fredrik D Johansson, and Christopher Morris. A survey on graph kernels. *Applied Network Science*, 5:1–42, 2020.

[KKT03] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146, 2003.

[KLG$^+$24] Sunwoo Kim, Soo Yong Lee, Yue Gao, Alessia Antelmi, Mirko Polato, and Kijung Shin. A survey on hypergraph neural networks: An in-depth and step-by-step guide. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 6534–6544, 2024.

[KM18] Sadamori Kojaku and Naoki Masuda. Core-periphery structure requires something else in the network. *New Journal of physics*, 20(4):043012, 2018.

[KMK24] David Y Kang, Qiaozhu Mei, and Sang-Wook Kim. Low mileage, high fidelity: Evaluating hypergraph expansion methods by quantifying the information loss. In *Proceedings of the ACM on Web Conference 2024*, pages 959–968, 2024.

[KMKP22] Sanjay Kumar, Abhishek Mallik, Anavi Khetarpal, and Bhawani Sankar Panda. Influence maximization in social networks using graph embedding and graph neural network. *Information Sciences*, 607:1617–1636, 2022.

[KR12] Kshitij Khare and Bala Rajaratnam. Sparse matrix decompositions and graph characterizations. *Linear algebra and its applications*, 437(3):932–947, 2012.

[KS80] Ross P Kindermann and J Laurie Snell. On the relation between markov random fields and social networks. *Journal of Mathematical Sociology*, 7(1):1–13, 1980.

[KSH17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[KW16] Thomas N Kipf and Max Welling. Variational graph auto-encoders. *NIPS Workshop on Bayesian Deep Learning*, 2016.

[KW17] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

[Law01] Eugene L Lawler. *Combinatorial optimization: networks and matroids*. Courier Corporation, 2001.

[LF09] Andrea Lancichinetti and Santo Fortunato. Community detection algorithms: a comparative analysis. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, 80(5):056117, 2009.

[LFA$^+$22] Xu T Liu, Jesun Firoz, Sinan Aksoy, Ilya Amburg, Andrew Lumsdaine, Cliff Joslyn, Brenda Praggastis, and Assefaw H Gebremedhin. High-order line graphs of non-uniform hypergraphs: Algorithms, applications, and experimental analysis. In *2022 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, pages 784–794. IEEE, 2022.

[LFH10] Andrea Landherr, Bettina Friedl, and Julia Heidemann. A critical review of centrality measures in social networks. *Business & Information Systems Engineering*, 2:371–385, 2010.

[LFZ$^+$24] Yangding Li, Shaobin Fu, Yangyang Zeng, Hao Feng, Ruoyao Peng, Jinghao Wang, and Shichao Zhang. Centrality-based relation aware heterogeneous graph neural network. *Knowledge-Based Systems*, 283:111174, 2024.

[LHT$^+$12] Xingjie Liu, Qi He, Yuanyuan Tian, Wang-Chien Lee, John McPherson, and Jiawei Han. Event-based social networks: linking the online and offline social worlds. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1032–1040, 2012.

[LJFS21] Yuanfu Lu, Xunqiang Jiang, Yuan Fang, and Chuan Shi. Learning to pre-train graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 4276–4284, 2021.

[LJW$^+$23] Chen Ling, Junji Jiang, Junxiang Wang, My T Thai, Renhao Xue, James Song, Meikang Qiu, and Liang Zhao. Deep graph representation learning and optimization for influence maximization. In *International Conference on Machine Learning*, pages 21350–21361. PMLR, 2023.

[LKG$^+$07] Jure Leskovec, Andreas Krause, Carlos Guestrin, Christos Faloutsos, Jeanne VanBriesen, and Natalie Glance. Cost-effective outbreak detection in networks. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 420–429, 2007.

[LL12] Wei Lu and Laks VS Lakshmanan. Profit maximization over social networks. In *2012 IEEE 12th international conference on data mining*, pages 479–488. IEEE, 2012.

[LLC+20] Suichan Li, Bin Liu, Dongdong Chen, Qi Chu, Lu Yuan, and Nenghai Yu. Density-aware graph for deep semi-supervised visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13400–13409, 2020.

[LPL+23] Peiyan Li, Liming Pan, Kai Li, Claudia Plant, and Christian Böhm. Interpretable subgraph feature extraction for hyperlink prediction. In *2023 IEEE International Conference on Data Mining (ICDM)*, pages 279–288. IEEE, 2023.

[LQW+20] Peiyan Li, Zhili Qin, Honglian Wang, Qinli Yang, and Junming Shao. Exploiting inconsistency problem in multi-label classification via metric learning. In *2020 IEEE International Conference on Data Mining (ICDM)*, pages 1100–1105. IEEE, 2020.

[LRK18] John Boaz Lee, Ryan Rossi, and Xiangnan Kong. Graph classification using structural attention. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1666–1674, 2018.

[LTY+24] Zexi Liu, Bohan Tang, Ziyuan Ye, Xiaowen Dong, Siheng Chen, and Yanfeng Wang. Hypergraph transformer for semi-supervised classification. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7515–7519. IEEE, 2024.

[LWB24] Peiyan Li, Honglian Wang, and Christian Böhm. Scalable graph classification via random walk fingerprints. In *2024 IEEE International Conference on Data Mining (ICDM)*, pages 231–240. IEEE, 2024.

[LWBS20] Peiyan Li, Honglian Wang, Christian Böhm, and Junming Shao. Online semi-supervised multi-label classification with label compression and local smooth regression. In Christian Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 1359–1365. International Joint Conferences on Artificial Intelligence Organization, 7 2020. Main track.

[LWL+20] Peiyan Li, Honglian Wang, Jianyun Lu, Qinli Yang, and Junming Shao. Community detection with local metric learning. In *2020 IEEE International Conference on Data Mining (ICDM)*, pages 312–321. IEEE, 2020.

[LWLB23] Peiyan Li, Honglian Wang, Kai Li, and Christian Bohm. Influence without authority: Maximizing information coverage in hypergraphs. In *Proceedings of the 2023 SIAM International Conference on Data Mining (SDM)*, pages 10–18. SIAM, 2023.

[LXY+17] Qi Liu, Biao Xiang, Nicholas Jing Yuan, Enhong Chen, Hui Xiong, Yi Zheng, and Yu Yang. An influence propagation view of pagerank. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 11(3):1–30, 2017.

[LYS22] Geon Lee, Jaemin Yoo, and Kijung Shin. Mining of real-world hypergraphs: Patterns, tools, and generators. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 5144–5147, 2022.

[LZ11] Linyuan Lü and Tao Zhou. Link prediction in complex networks: A survey. *Physica A: statistical mechanics and its applications*, 390(6):1150–1170, 2011.

[LZW+21] Weiwen Liu, Yin Zhang, Jianling Wang, Yun He, James Caverlee, Patrick PK Chan, Daniel S Yeung, and Pheng-Ann Heng. Item relationship graph neural networks for e-commerce. *IEEE Transactions on Neural Networks and Learning Systems*, 33(9):4785–4799, 2021.

[MBTK22] Nairouz Mrabah, Mohamed Bouguessa, Mohamed Fawzi Touati, and Riadh Ksantini. Rethinking graph auto-encoder models for attributed graph clustering. *IEEE Transactions on Knowledge and Data Engineering*, 35(9):9037–9053, 2022.

[MCCD13] Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In Yoshua Bengio and Yann LeCun, editors, *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, 2013.

[ME11] Aditya Krishna Menon and Charles Elkan. Link prediction via matrix factorization. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2011, Athens, Greece, September 5-9, 2011, Proceedings, Part II 22*, pages 437–452. Springer, 2011.

[Mos87] Lincoln E Moses. Graphical methods in statistical analysis. *Annual Review of Public Health*, 8(1):309–353, 1987.

[MPL17] Naoki Masuda, Mason A Porter, and Renaud Lambiotte. Random walks and diffusion on networks. *Physics reports*, 716:1–58, 2017.

[MS24] Marco Morucci and Arthur Spirling. Model complexity for supervised learning: Why simple models almost always work best, and why it matters for applied research. Technical report, Working Paper, 2024.

[MSGL14] Seth A Myers, Aneesh Sharma, Pankaj Gupta, and Jimmy Lin. Information network or social network? the structure of the twitter follow graph. In *Proceedings of the 23rd international conference on world wide web*, pages 493–498, 2014.

[MZY17] Yan Mei, Weiliang Zhao, and Jian Yang. Influence maximization on twitter: A mechanism for effective marketing campaign. In *2017 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE, 2017.

[New02] Mark EJ Newman. Assortative mixing in networks. *Physical review letters*, 89(20):208701, 2002.

[NJW01] Andrew Ng, Michael Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 14, 2001.

[NVM$^+$18] Huda Nassar, Nate Veldt, Shahin Mohammadi, Ananth Grama, and David F Gleich. Low rank spectral network alignment. In *Proceedings of the 2018 World Wide Web Conference*, pages 619–628, 2018.

[NZ13] Huy Nguyen and Rong Zheng. On budgeted influence maximization in social networks. *IEEE Journal on Selected Areas in Communications*, 31(6):1084–1094, 2013.

[OMK22] Lutz Oettershagen, Petra Mutzel, and Nils M Kriege. Temporal walk centrality: ranking nodes in evolving networks. In *Proceedings of the ACM Web Conference 2022*, pages 1640–1650, 2022.

[PARS14] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710, 2014.

[PCC14] Nicole Pearcy, Jonathan J Crofts, and Nadia Chuzhanova. Hypergraph models of metabolism. *International Journal of Biological, Veterinary, Agricultural and Food Engineering*, 8(8):752–756, 2014.

[PHL$^+$18] Shirui Pan, Ruiqi Hu, Guodong Long, Jing Jiang, Lina Yao, and Chengqi Zhang. Adversarially regularized graph autoencoder for graph embedding. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 2609–2615. International Joint Conferences on Artificial Intelligence Organization, 7 2018.

[PHWV20] Pierre Perrault, Jennifer Healey, Zheng Wen, and Michal Valko. Budgeted online influence maximization. In *International Conference on Machine Learning*, pages 7620–7631. PMLR, 2020.

[PSL90] Alex Pothen, Horst D Simon, and Kang-Pu Liou. Partitioning sparse matrices with eigenvectors of graphs. *SIAM journal on matrix analysis and applications*, 11(3):430–452, 1990.

[PSL$^+$21] Liming Pan, Hui-Juan Shang, Peiyan Li, Haixing Dai, Wei Wang, and Lixin Tian. Predicting hyperlinks via hypernetwork loop structure. *Europhysics Letters*, 135(4):48005, 2021.

[PTV$^+$24] George Panagopoulos, Nikolaos Tziortziotis, Michalis Vazirgiannis, Jun Pang, and Fragkiskos D Malliaros. Learning graph representations for influence maximization. *Social Network Analysis and Mining*, 14(1):203, 2024.

[PTVM23] George Panagopoulos, Nikolaos Tziortziotis, Michalis Vazirgiannis, and Fragkiskos Malliaros. Maximizing influence with graph neural networks. In *Proceedings of the International Conference on Advances in Social Networks Analysis and Mining*, pages 237–244, 2023.

[QDM$^+$18] Jiezhong Qiu, Yuxiao Dong, Hao Ma, Jian Li, Kuansan Wang, and Jie Tang. Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec. In *Proceedings of the eleventh ACM international conference on web search and data mining*, pages 459–467, 2018.

[QDM$^+$19] Jiezhong Qiu, Yuxiao Dong, Hao Ma, Jian Li, Chi Wang, Kuansan Wang, and Jie Tang. Netsmf: Large-scale network embedding as sparse matrix factorization. In *The World Wide Web Conference*, pages 1509–1520, 2019.

[RBX$^+$20] Yu Rong, Yatao Bian, Tingyang Xu, Weiyang Xie, Ying Wei, Wenbing Huang, and Junzhou Huang. Self-supervised graph transformer on large-scale molecular data. *Advances in Neural Information Processing Systems*, 33, 2020.

[RGD+22] Ladislav Rampášek, Michael Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Dominique Beaini. Recipe for a general, powerful, scalable graph transformer. *Advances in Neural Information Processing Systems*, 35:14501–14515, 2022.

[Rob23] Bradley Robinson. Governance on, with, behind, and beyond the discord platform: A study of platform practices in an informal learning context. *Learning, Media and Technology*, 48(1):81–94, 2023.

[Ruh00] Britta Ruhnau. Eigenvector-centrality—a node-centrality? *Social networks*, 22(4):357–365, 2000.

[RZSW24] Xiaobin Rui, Jiaxin Zhuang, Chengcheng Sun, and Zhixiao Wang. Higher-order link prediction via light hypergraph neural network and hybrid aggregator. *International Journal of Machine Learning and Cybernetics*, pages 1–15, 2024.

[SDWL14] Jianbing Shen, Yunfan Du, Wenguan Wang, and Xuelong Li. Lazy random walks for superpixel segmentation. *IEEE Transactions on Image Processing*, 23(4):1451–1462, 2014.

[SGS18] Qi Suo, Jin-Li Guo, and Ai-Zhong Shen. Information spreading dynamics in hypernetworks. *Physica A: Statistical Mechanics and its Applications*, 495:475–487, 2018.

[SLV+25] Hamed Shirzad, Honghao Lin, Balaji Venkatachalam, Ameya Velingker, David Woodruff, and Danica J Sutherland. Even sparser graph transformers. *Advances in Neural Information Processing Systems*, 37:71277–71305, 2025.

[SRJ17] Shakila Shaikh, Sheetal Rathi, and Prachi Janrao. Recommendation system in e-commerce websites: a graph based approached. In *2017 IEEE 7th International Advance Computing Conference (IACC)*, pages 931–934. IEEE, 2017.

[SV93] Moninder Singh and Marco Valtorta. An algorithm for the construction of bayesian network structures from data. In *Uncertainty in artificial intelligence*, pages 259–265. Elsevier, 1993.

[SVV+23] Hamed Shirzad, Ameya Velingker, Balaji Venkatachalam, Danica J Sutherland, and Ali Kemal Sinop. Exphormer: Sparse transformers for graphs. In *International Conference on Machine Learning*, pages 31613–31632. PMLR, 2023.

[SXZF07] Jimeng Sun, Yinglian Xie, Hui Zhang, and Christos Faloutsos. Less is more: Compact matrix decomposition for large sparse graphs. In *Proceedings of the 2007 SIAM International Conference on Data Mining*, pages 366–377. SIAM, 2007.

[SYL+17] Junming Shao, Zhongjing Yu, Peiyan Li, Wei Han, Christian Sorg, and Qinli Yang. Exploring common and distinct structural connectivity patterns between schizophrenia and major depression via cluster-driven nonnegative matrix factorization. In *2017 IEEE International Conference on Data Mining (ICDM)*, pages 1081–1086. IEEE, 2017.

[SZA+20] Zhan Su, Xiliang Zheng, Jun Ai, Yuming Shen, and Xuanxiong Zhang. Link prediction in recommender systems based on vector similarity. *Physica A: Statistical Mechanics and its Applications*, 560:125154, 2020.

[SZA21] Saurabh Sawlani, Lingxiao Zhao, and Leman Akoglu. Fast attributed graph embedding via density of states. In *IEEE ICDM*, pages 559–568, 2021.

[SZK21] Balasubramaniam Srinivasan, Da Zheng, and George Karypis. Learning over families of sets-hypergraph representation learning for higher order tasks. In *Proceedings of the 2021 SIAM International Conference on Data Mining (SDM)*, pages 756–764. SIAM, 2021.

[TCR10] Carla Taramasco, Jean-Philippe Cointet, and Camille Roth. Academic team formation as evolving hypergraphs. *Scientometrics*, 85(3):721–740, 2010.

[TFP06] Hanghang Tong, Christos Faloutsos, and Jia-Yu Pan. Fast random walk with restart and its applications. In *Sixth international conference on data mining (ICDM'06)*, pages 613–622. IEEE, 2006.

[TLSZ14] Kewei Tang, Risheng Liu, Zhixun Su, and Jie Zhang. Structure-constrained low-rank representation. *IEEE transactions on neural networks and learning systems*, 25(12):2167–2179, 2014.

[TLZ+14] Fangshuang Tang, Qi Liu, Hengshu Zhu, Enhong Chen, and Feida Zhu. Diversified social influence maximization. In *2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014)*, pages 455–459. IEEE, 2014.

[TMK+18] Anton Tsitsulin, Davide Mottin, Panagiotis Karras, Alexander Bronstein, and Emmanuel Müller. Netlsd: hearing the shape of a graph. In *ACM KDD*, pages 2347–2356, 2018.

[TQW$^+$15] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In *Proceedings of the 24th international conference on world wide web*, pages 1067–1077, 2015.

[TSO21] Samaneh Torkzadeh, Hadi Soltanizadeh, and Ali A Orouji. Energy-aware routing considering load balancing for sdn: a minimum graph-based ant colony optimization. *Cluster Computing*, 24(3):2293–2312, 2021.

[TSX15] Youze Tang, Yanchen Shi, and Xiaokui Xiao. Influence maximization in near-linear time: A martingale approach. In *Proceedings of the 2015 ACM SIGMOD international conference on management of data*, pages 1539–1554, 2015.

[TTA$^+$21] Shantanu Thakoor, Corentin Tallec, Mohammad Gheshlaghi Azar, Remi Munos, Petar Veličković, and Michal Valko. Bootstrapped representation learning on graphs. In *ICLR 2021 Workshop on Geometrical and Topological Representation Learning*, 2021.

[TXS14] Youze Tang, Xiaokui Xiao, and Yanchen Shi. Influence maximization: Near-optimal time complexity meets practical efficiency. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, pages 75–86, 2014.

[VCC$^+$18] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

[VFH$^+$19] Petar Velickovic, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. Deep graph infomax. *ICLR (poster)*, 2(3):4, 2019.

[VP19] Maria Vaida and Kevin Purcell. Hypergraph link prediction: learning drug interaction networks embeddings. In *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, pages 1860–1865. IEEE, 2019.

[VSKB10] S Vichy N Vishwanathan, Nicol N Schraudolph, Risi Kondor, and Karsten M Borgwardt. Graph kernels. *The Journal of Machine Learning Research*, 11:1201–1242, 2010.

[VSP$^+$17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[VZ17] Saurabh Verma and Zhi-Li Zhang. Hunt for the unique, stable, sparse and fast feature learning on graphs. *NeurIPS*, 2017.

[WCZ16] Daixin Wang, Peng Cui, and Wenwu Zhu. Structural deep network embedding. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1225–1234, 2016.

[WLLS21] Honglian Wang, Peiyan Li, Yang Liu, and Junming Shao. Towards real-time demand-aware sequential poi recommendation. *Information Sciences*, 547:482–497, 2021.

[WLT+21] Honglian Wang, Peiyan Li, Wujun Tao, Bailin Feng, and Junming Shao. Learning dynamic user behavior based on error-driven event representation. In *Proceedings of the Web Conference 2021*, pages 2457–2465, 2021.

[Woe17] Wolfgang Woess. *Groups, graphs and random walks*, volume 436. Cambridge University Press, 2017.

[WPL+17] Chun Wang, Shirui Pan, Guodong Long, Xingquan Zhu, and Jing Jiang. Mgae: Marginalized graph autoencoder for graph clustering. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 889–898, 2017.

[WS98] Duncan J Watts and Steven H Strogatz. Collective dynamics of 'small-world'networks. *nature*, 393(6684):440–442, 1998.

[WZ06] Fei Wang and Changshui Zhang. Label propagation through linear neighborhoods. In *Proceedings of the 23rd international conference on Machine learning*, pages 985–992, 2006.

[WZ21] Xiyuan Wang and Muhan Zhang. Glass: Gnn with labeling tricks for subgraph representation learning. In *International conference on learning representations*, 2021.

[WZM+22] Yili Wang, Kaixiong Zhou, Rui Miao, Ninghao Liu, and Xin Wang. Adagcl: Adaptive subgraph contrastive learning to generalize large-scale graph training. In *Proceedings of the 31st ACM international conference on information & knowledge management*, pages 2046–2055, 2022.

[XCL+21] Dongkuan Xu, Wei Cheng, Dongsheng Luo, Haifeng Chen, and Xiang Zhang. Infogcl: Information-aware graph contrastive learning. *Advances in Neural Information Processing Systems*, 34:30414–30425, 2021.

[XG04] Wenpu Xing and Ali Ghorbani. Weighted pagerank algorithm. In *Proceedings. Second Annual Conference on Communication Networks and Services Research, 2004.*, pages 305–314. IEEE, 2004.

[XHLJ19] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019.

[XLN$^+$19] Feng Xia, Jiaying Liu, Hansong Nie, Yonghao Fu, Liangtian Wan, and Xiangjie Kong. Random walks: A review of algorithms and applications. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 4(2):95–107, 2019.

[XLN$^+$22] Chenxin Xu, Maosen Li, Zhenyang Ni, Ya Zhang, and Siheng Chen. Groupnet: Multiscale hypergraph neural networks for trajectory prediction with relational reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6498–6507, 2022.

[XRK13] Ye Xu, Dan Rockmore, and Adam M Kleinbaum. Hyperlink prediction in hypernetworks using latent social features. In *International Conference on Discovery Science*, pages 324–339. Springer, 2013.

[XSC$^+$18] Bingbing Xu, Huawei Shen, Qi Cao, Yunqi Qiu, and Xueqi Cheng. Graph wavelet neural network. In *International Conference on Learning Representations*, 2018.

[XSY$^+$21] Feng Xia, Ke Sun, Shuo Yu, Abdul Aziz, Liangtian Wan, Shirui Pan, and Huan Liu. Graph learning: A survey. *IEEE Transactions on Artificial Intelligence*, 2(2):109–127, 2021.

[XWL$^+$24] Yujie Xing, Xiao Wang, Yibo Li, Hai Huang, and Chuan Shi. Less is more: on the over-globalizing problem in graph transformers. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

[YAT16] Zhao Yang, René Algesheimer, and Claudio J Tessone. A comparative analysis of community detection algorithms on artificial networks. *Scientific reports*, 6(1):30750, 2016.

[YCL$^+$21] Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? *Advances in neural information processing systems*, 34:28877–28888, 2021.

[YCS$^+$20] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. *Advances in neural information processing systems*, 33:5812–5823, 2020.

[YCW$^+$24] Yuguang Yan, Yuanlin Chen, Shibo Wang, Hanrui Wu, and Ruichu Cai. Hypergraph joint representation learning for hypervertices and hyperedges via cross expansion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 9232–9240, 2024.

[YHS$^+$24] Shuo Yu, Huafei Huang, Yanming Shen, Pengfei Wang, Qiang Zhang, Ke Sun, and Honglong Chen. Formulating and representing multiagent systems with hypergraphs. *IEEE Transactions on Neural Networks and Learning Systems*, 2024.

[YLG20] Lan Yang, Zhiwu Li, and Alessandro Giua. Containment of rumor spread in complex social networks. *Information Sciences*, 506:113–130, 2020.

[YLHS24] Taehyung Yu, Soo Yong Lee, Hyunjin Hwang, and Kijung Shin. Prediction is not classification: On formulation and evaluation of hyperedge prediction. In *2024 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 349–356. IEEE, 2024.

[YLW$^+$19] Ruidong Yan, Deying Li, Weili Wu, Ding-Zhu Du, and Yongcai Wang. Minimizing influence of rumors by blockers on social networks: algorithms and analysis. *IEEE transactions on network science and engineering*, 7(3):1067–1078, 2019.

[YNN$^+$20] Naganand Yadati, Vikram Nitin, Madhav Nimishakavi, Prateek Yadav, Anand Louis, and Partha Talukdar. Nhp: Neural hypergraph link prediction. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 1705–1714, 2020.

[YNY$^+$19] Naganand Yadati, Madhav Nimishakavi, Prateek Yadav, Vikram Nitin, Anand Louis, and Partha Talukdar. Hypergcn: A new method for training graph convolutional networks on hypergraphs. In *Advances in Neural Information Processing Systems (NeurIPS) 32*, pages 1509–1520, 2019.

[YV15] Pinar Yanardag and SVN Vishwanathan. Deep graph kernels. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1365–1374, 2015.

[YWYA22] Chaoqi Yang, Ruijie Wang, Shuochao Yao, and Tarek Abdelzaher. Semi-supervised hypergraph node classification on hypergraph line expansion.

In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 2352–2361, 2022.

[ZC18] Muhan Zhang and Yixin Chen. Link prediction based on graph neural networks. *Advances in neural information processing systems*, 31, 2018.

[ZCJC18] Muhan Zhang, Zhicheng Cui, Shali Jiang, and Yixin Chen. Beyond link prediction: Predicting hyperlinks in adjacency space. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[ZCNC18] Muhan Zhang, Zhicheng Cui, Marion Neumann, and Yixin Chen. An end-to-end deep learning architecture for graph classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[ZDK15] Vasily Zaburdaev, Sergey Denisov, and Joseph Klafter. Lévy walks. *Reviews of Modern Physics*, 87(2):483–530, 2015.

[ZFNP15] Olga Zoidi, Eftychia Fotiadou, Nikos Nikolaidis, and Ioannis Pitas. Graph-based label propagation in digital media: A review. *ACM Computing Surveys (CSUR)*, 47(3):1–35, 2015.

[ZGW19] Jianming Zhu, Smita Ghosh, and Weili Wu. Group influence maximization problem in social networks. *IEEE Transactions on Computational Social Systems*, 6(6):1156–1164, 2019.

[ZHSS24] Shichang Zhang, Ziniu Hu, Arjun Subramonian, and Yizhou Sun. Motif-driven contrastive learning of graph representations. *IEEE Transactions on Knowledge and Data Engineering*, 36(8):4063–4075, 2024.

[ZLJW$^+$22] Jinhang Zuo, Xutong Liu, Carlee Joe-Wong, John CS Lui, and Wei Chen. Online competitive influence maximization. In *International Conference on Artificial Intelligence and Statistics*, pages 11472–11502. PMLR, 2022.

[ZLX$^+$21] Muhan Zhang, Pan Li, Yinglong Xia, Kai Wang, and Long Jin. Labeling trick: A theory of using graph neural networks for multi-node representation learning. *Advances in Neural Information Processing Systems*, 34:9061–9073, 2021.

[ZLY$^+$17] Yuqing Zhu, Deying Li, Ruidong Yan, Weili Wu, and Yuanjun Bi. Maximizing the influence and profit in social networks. *IEEE Transactions on Computational Social Systems*, 4(3):54–64, 2017.

[ZLZL22] Hongyuan Zhang, Pei Li, Rui Zhang, and Xuelong Li. Embedding graph auto-encoder for graph clustering. *IEEE Transactions on Neural Networks and Learning Systems*, 34(11):9352–9362, 2022.

[ZMHD15] Xiaodong Zhang, Elise Miller-Hooks, and Kevin Denny. Assessing the role of network topology in transportation network resilience. *Journal of transport geography*, 46:35–45, 2015.

[ZQL$^+$18] Zhong Zhang, Zhili Qin, Peiyan Li, Qinli Yang, and Junming Shao. Multi-view discriminative learning via joint non-negative matrix factorization. In *Database Systems for Advanced Applications: 23rd International Conference, DASFAA 2018, Gold Coast, QLD, Australia, May 21-24, 2018, Proceedings, Part II 23*, pages 542–557. Springer, 2018.

[ZSBS12] Tinghui Zhou, Hanhuai Shan, Arindam Banerjee, and Guillermo Sapiro. Kernelized probabilistic matrix factorization: Exploiting graphs and side information. In *Proceedings of the 2012 SIAM international Conference on Data mining*, pages 403–414. SIAM, 2012.

[ZWX$^+$18] Zhen Zhang, Mianzhi Wang, Yijian Xiang, Yan Huang, and Arye Nehorai. Retgk: Graph kernels based on return probabilities of random walks. *Advances in Neural Information Processing Systems*, 31, 2018.

[ZXY$^+$20] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. Deep graph contrastive representation learning. In *ICML Workshop on Graph Representation Learning and Beyond*, 2020.

[ZZ17] Yu Zhang and Yan Zhang. Top-k influential nodes in social networks: A game perspective. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1029–1032, 2017.

[ZZG$^+$18] Jianming Zhu, Junlei Zhu, Smita Ghosh, Weili Wu, and Jing Yuan. Social influence maximization in hypergraph in social networks. *IEEE Transactions on Network Science and Engineering*, 6(4):801–811, 2018.

[ZZM20] Ruochi Zhang, Yuesong Zou, and Jian Ma. Hyper-SAGNN: a self-attention based graph neural network for hypergraphs. In *International Conference on Learning Representations (ICLR)*, 2020.

List of Figures

A. Appended Papers

The remainder of this dissertation contains the contributed publications with their corresponding supplementary pages. Further, we provide essential information about the publication, such as the title, abstract, the conference it was published, all authors, the division of work among authors, and the assigned Digital Object Identifier (DOI).

A.1 Influence without Authority: Maximizing Information Coverage in Hypergraphs

Title	Influence without Authority: Maximizing Information Coverage in Hypergraphs
Authors	Peiyan Li, Honglian Wang, Kai Li, Christian Böhm
Publication Outlet	Proceedings of the 2023 SIAM International Conference on Data Mining (SDM). Society for Industrial and Applied Mathematics, 2023. *CORE ranking: A, Accepted as an oral presentation; acceptance rate for oral presentations: 27.4%*
DOI-URL	`https://doi.org/10.1137/1.9781611977653.ch2`
Code	`https://github.com/KXDY233/InfDis`

Division of Work

Peiyan Li
Formulation of the Problem: information coverage maximization in hypergraphs. Problem Analysis: the proposed problem is NP-hard; the computation of the information coverage of a given seed is #P-hard; and the information coverage function is both monotone and submodular. Formulation of the designed heuristic solution. Implementing the algorithm. Designing the experiments. Executing the experiments. Writing the majority of the paper.

Honglian Wang
Creating the visualizations. Discussions on problem analysis. Refinement of the draft.

Kai Li
Discussions on problem analysis. Refinement of the draft.

Christian Böhm
Discussions on problem analysis: how to prove the NP-hardness by reduction. Frequent suggestions for improving the writing. Refinement of the final draft. Mentoring and supervision.

Abstract

In many social networks, besides peer-to-peer communication, people share information via groups. An interesting problem arises in this scenario: for such networks, which are the best groups to start information diffusion so that the number of eventually informed nodes can be maximized? In this study, we formulate a novel information coverage maximization problem in the context of hypergraphs, wherein nodes are connected by arbitrary-size hyperedges (i.e., groups). In contrast to the existing literature on influence maximization, which aims to find authority nodes with high influence, we are interested in identifying the key groups. To address this problem, we present a new information diffusion model for hypergraphs, namely Hypergraph-Independent-Cascade (HIC). HIC generalizes the popular independent cascade model to hypergraphs to allow capturing group-level information diffusion. We prove the NP-hardness of the proposed problem under HIC, and the submodular monotone property of the information coverage function. Further, inspired by the Degree Discount algorithm, we derive a new heuristic method named Influence Discount (InfDis). Extensive experiments provide empirical evidence for the effectiveness and efficiency of our approach.

Thesis-Reference [LWLB23]

The paper was removed for the publication version of this thesis.

A.2 Interpretable Subgraph Feature Extraction for Hyperlink Prediction

Title	Interpretable Subgraph Feature Extraction for Hyperlink Prediction
Authors	Peiyan Li, Liming Pan, Kai Li, Claudia Plant, Christian Böhm
Publication Outlet	Proceedings of the 2023 IEEE International Conference on Data Mining (ICDM) *CORE ranking: A*. Regular paper acceptance rate: 9.37%.*
DOI	`https://doi.org/10.1109/ICDM58522.2023.00037`
Code	`https://github.com/KXDY233/SSF`
Note	This paper was recognized as one of the top-ranking papers at IEEE ICDM 2023 and was invited for extension and submission to Knowledge and Information Systems.

Division of Work

Peiyan Li
Developing the idea of using subgraph feature extraction for hyperlinnk prediction. Implementing the majority of the algorithm. Designing the experiments. Executing the experiments. Creating the visualizations. Writing the majority of the paper.

Liming Pan
Implementing a part of the algorithm. Discussions on the novelty of using subgraph feature extraction.

Kai Li
Refinement of the draft.

Claudia Plant
Discussions on the introduction: how to introduce the motivation and how to explain the novelty. Frequent suggestions for improving the writing. Refinement of the final draft. Mentoring and supervision.

Christian Böhm
Identification of relevant related works. Frequent suggestions for improving the writing. Refinement of the final draft. Mentoring and supervision.

Abstract

Hyperlink prediction aims to predict interactions among multiple entries, constituting a practical yet challenging problem in the literature. While a handful of solutions have been proposed, they generally operate on the entire hypergraph. A practical subgraph-based solution not only enables better identification of localized characteristics of the central hyperedge but also alleviates scalability concerns. In this study, we present SSF, an innovative hyperlink prediction methodology based on Subgraph Structural Features. The rationale behind SSF is that hyperedges and non-hyperedges exhibit distinct local patterns, which can be unveiled through the assimilation of subgraph structural features. To this end, we utilize well-established structural heuristics such as walks and loops as the fundamental building blocks. We commence by extracting a subgraph encompassing each focal hyperedge, subsequently integrating an edge weakening scheme to facilitate feature extraction from the initial subgraph and its variations. The extracted feature vector is interpretable, and the designed edge weakening scheme empowers SSF with an adaptive capability to handle hypergraphs with varying densities. Lastly, a multilayer perceptron classifier is trained for prediction. Experiment results on ten real-world hypergraph networks demonstrate the effectiveness of the proposed approach.

Thesis-Reference [LPL^{+}23]

The paper was removed for the publication version of this thesis.

A.3 Scalable Graph Classification via Random Walk Fingerprints

Title	Scalable Graph Classification via Random Walk Fingerprints
Authors	Peiyan Li, Honglian Wang, Christian Böhm
Publication Outlet	Proceedings of the 2024 IEEE International Conference on Data Mining (ICDM) *CORE ranking: A*. Regular paper acceptance rate: 10.9%.*
DOI	`https://doi.org/10.1109/ICDM59182.2024.00030`
Code	`https://github.com/KXDY233/RWF`
Note	This paper win the Best Paper Award of IEEE ICDM 2024.

Division of Work

Peiyan Li
Developing the idea of using random walk fingerprints for graph classification. Implementing the algorithm. Designing the experiments. Executing the experiments. Creating the visualizations. Writing the majority of the paper.

Honglian Wang
Creating the visualizations. Refinement of the draft.

Christian Böhm
Identification of relevant related works. Suggestions for improving the writing. Refinement of the draft. Mentoring and supervision.

Abstract

Graph classification has long been a focus of network mining, with graph kernel methods and representation learning at the forefront. Despite their success, many of these studies require heavy computation, making them impractical for large-scale datasets. In this paper, we design a novel structural feature extraction technique that leverages node subsets and connection strength reflected by random-walk-based heuristics, presenting a scalable, unsupervised, and easily interpretable alternative. Initially, we partition each graph based on the structural roles of nodes. This process creates soft alignments of node subsets across graphs of varying sizes. Then, we measure the connection strengths within and between these subsets, which form the fingerprints for graph classification. Additionally, this technique can seamlessly incorporate node features. Through empirical assessment encompassing a broad range of graph datasets, we demonstrate that our method achieves high levels of computational efficiency while maintaining robust classification accuracy.

Thesis-Reference [LWB24]

The paper was removed for the publication version of this thesis.