Information Fusion 133 (2026) 104304

Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

Full Length Article

Towards graph concept-cognitive learning: A survey and beyond

Fei Hao [a,*], Mengyu Yan [b,c,*], Jinhai Li [b,c], Weihua Xu [d]

[a] School of Artificial Intelligence and Computer Science, Shaanxi Normal University, Xi'an, 710119, Shaanxi, China
[b] Yunnan Key Laboratory of Complex Systems and Brain-Inspired Intelligence, Kunming University of Science and Technology, Kunming, 650500, Yunan, China
[c] Faculty of Science, Kunming University of Science and Technology, Kunming, 650500, Yunan, China
[d] School of Artificial Intelligence, Southwest University, Chongqing, 400715, China

ARTICLE INFO

Keywords:
Graph data
Concept-cognitive learning
Formal concept analysis

ABSTRACT

Graphs serve as a powerful representation for the networked structure of connected data and are widely applied in domains such as social systems, ecosystems, biological networks, knowledge graphs, and information systems. With the advancing integration of graph learning technologies and cognitive sciences, Graph Concept-cognitive Learning (GCCL) has emerged as a novel paradigm, merging Concept-cognitive Learning (CCL) with graph analysis. This paper provides a comprehensive overview of GCCL, presenting its foundational framework and detailing methods for integrating concept-cognitive learning into various graph representations. These representations enable tasks such as graph structure analysis, knowledge discovery, and graph classification. Furthermore, the paper explores key application domains, including social networking analysis, knowledge graph reasoning, and combinatorial optimization, while introducing datasets and simulation platforms tailored for GCCL research. Finally, we discuss promising research directions, positioning GCCL as a transformative approach in the broader field of graph learning.

1. Introduction

Graph data plays a pivotal role in modern data analysis and computational applications, enabling more efficient and intuitive ways to represent relationships between entities. Unlike traditional tabular data, graph data structures model complex, interconnected systems by defining nodes (i.e, objects) and edges (i.e, relationships). Graph data mining and analysis exhibits crucial practical values in various fields such as social network analysis [1], recommendation systems [2], fraud detection [3], and bioinformatics [4], where understanding the interplay between entities is essential. With the increasing volume and complexity of data in various industries, the ability to harness graph data for pattern recognition, prediction, and optimization has become a powerful tool in advancing technology and decision-making processes.

Graph learning models have gained significant attention for their ability to capture complex relationships in data represented as graphs [5,6]. The existing graph learning methods mainly include the deep learning method based on graph convolution network and the graph contrastive learning method based on self-supervised learning. Graph convolution networks (GCNs) [7,8] are one of the core methods of graph learning. Its basic idea is to aggregate and update features on graph structure through message passing mechanism. The core of GCN is to use the graph's adjacency matrix and node feature matrix to aggregate information from neighbor nodes into the target node via a convolutional operation, thereby learning node representation. Graph contrastive learning [9,10] is also an important research direction in the field of graph learning in recent years, which belongs to self-supervised learning. The core idea is to use unlabeled graph data to generate supervision signals by constructing a comparative task, so as to learn a high-quality graph representation. These models excel in various tasks like node classification [11,12], link prediction [13,14], and graph classification [15,16]. GNNs leverage message-passing mechanisms to aggregate information from neighboring nodes, allowing them to model relational data effectively [17]. However, GNNs often struggle with scalability as they require high memory and computational resources for large-scale graphs. Additionally, they suffer from over-smoothing, where increasing the number of layers leads to indistinguishable node embeddings, and its poor interpretability is caused by "black box" characteristic. Furthermore, GNNs may be sensitive to noisy or incomplete graph structures, impacting model robustness. Improving scalability, depth capability, and robustness to noise has become active research areas of research in graph learning.

Concept-Cognitive Learning (CCL) [18–21] is a new cross-research content in Artificial Intelligence, which is formed by formal concept analysis [22], rough set [23], granular computing [24], and cognitive computing [25]. Moreover, CCL refers to learning concepts from given

* Corresponding authors.
E-mail addresses: fhao@snnu.edu.cn (F. Hao), 20240061@kust.edu.cn (M. Yan).

https://doi.org/10.1016/j.inffus.2026.104304
Received 14 November 2025; Received in revised form 21 February 2026; Accepted 13 March 2026
Available online 16 March 2026

clues through specific cognitive models to reveal the systematic laws of human brain learning concepts. It should be pointed out that the specific forms of clues here can be diversified, such as some objects covered by the concept, incomplete features of the concept, incomplete concept information, and so on. Up to now, CCL has aroused the research interest of many scholars and has established different CCL models [26–30]. These models are established by the axiomatic method and analysis of cognitive mechanisms, and they can explain the mechanism and process of human brain learning concepts to a certain extent. In addition, many scholars have done relevant research to confirm that it has certain advantages to complete downstream tasks by CCL [31–37], so graph learning is no exception.

Graph Concept-cognitive Learning (GCCL) has emerged as an innovative paradigm that organically fuses CCL with graph analysis [38–41]. As a powerful tool for describing network structures of connected data, graphs are widely applied in fields such as computational social systems, biological networks, and knowledge graphs. GCCL leverages this advantage and integrates the core logic of CCL simulating human cognitive processes to learn and apply concepts through their extent and intent—into graph analysis frameworks. This integration not only provides a new theoretical system and technical support for pattern recognition [42], reasoning [43,44], cognitive memory enhancement [45], and decision-making in complex networks [46,47] but also enables more accurate simulation of human cognitive mechanisms towards graph structures [48]. Therefore, GCCL exhibits tremendous potential in knowledge discovery, intelligent reasoning, and system optimization, opening up new research directions for the in-depth integration of artificial intelligence and cognitive computing. Its specific applications range from enhancing graph classification and link prediction tasks to supporting concept structure-enhanced graph federated learning in practical scenarios like traffic flow prediction, demonstrating broad application prospects in both academic research and industrial practices.

1.1. Contributions and outline

The major contributions of this paper are summarized as follows.

- *Graph representation with CCL*: In order to adapt to CCL theory with graph data, we present the representation approaches for the undirected graph, attributed graph, and uncertain graph, respectively. That is to say, the various graph can be represented with the different graph formal contexts.
- *Graph cognition with CCL*: By combining FCA's structured hierarchy of concepts with graph cognition, we pioneer a framework enables more interpretable, scalable, and adaptive graph learning systems. Technically, two equivalence relations between concept/concept stability and graph topology are established. With these equivalence relations, we elaborate the concept cognition approaches for the undirected graph, attributed graph, and uncertain graph, for further supporting the graph concept-cognitive learning paradigm.
- *Datasets and Platforms*: To better facilitate the experimental research and validation on GCCL, we summarize the relevant benchmarking graph datasets as well as the existing simulation platforms or tool kits based on GCCL.
- *Insights into future directions in GCCL*: In addition to conducting a qualitative analysis of existing methods, we highlight potential research directions in GCCL by outlining key open issues and related challenges.

The remainder of this article is organized as follows. The introduction to GCCL is given in Section 2. Section 3 presents the graph representation approaches with CCL for undirected graph, attributed graph as well as uncertain graph, respectively. Then, the graph cognition mechanisms with CCL is presented in Section 4 via two equivalence relations between concept/concept stability and graph topology. The applications of graph CCL, datasets and platforms are examined in Sections 5 and 6. Some open challenges as well as future directions are discussed in Section 7. Finally, Section 8 concludes the article. The overall framework of the is shown in Fig. 1.

2. Graph concept-cognitive learning

Graph data is becoming more and more common in life, and its forms are diversified. This section first reviews the graph-related knowledge [39,49–52].

2.1. Graph-related definitions

Definition 1 (Graph). A graph is normally represented as $G = (V, E)$, where $V = \{v_1, v_2, \cdots, v_n\}$ is the set of vertices, and $E = \{e_1, e_2, \cdots, e_m\}$ denotes the set of edges, i.e., $E \subseteq V \times V$. The topology of G can be formulated with an adjacency matrix $A \in \mathbb{R}^{n\times n}$, in which $A_{ij} = 1$ *iff* $e(v_i, v_j) \in E$.

A graph is called to be connected if and only if it starts from a node, passes through one or more intermediate nodes, and can reach any other node. That is, there is a path between any two different nodes.

Definition 2 (Dynamic Graph). A dynamic graph can be represented as a sequence of graphs $G_t = (V_t, E_t)$ where each graph G_t is a snapshot of the graph at a given time $t \in \{1, 2, \cdots, T\}$. Formally, a dynamic graph is formulated as follows:

$$\widetilde{G} = \{G_1 = (V_1, E_1), G_2 = (V_2, E_2), \cdots, G_T = (V_T, E_T)\}, \quad (1)$$

here, V_t is the set of vertices at time t; $E_t \subseteq V_t \times V_t$ is the set of edges at time t; T is the total number of discrete time steps.

An attributed graph is a graph that each vertex (node) and/or edge is associated with a set of attributes or features. These attributes can represent various characteristics of the nodes or edges, such as labels, numerical values, or categorical properties.

Definition 3 (Attributed Graph). An attributed graph is defined as $G = (V, E, \mathbf{X}_V)$, where V is the set of vertices, $E \subseteq V \times V$ is the set of edges, and $\mathbf{X}_V : V \to \mathbb{R}^d$ assigns a d-dimensional attribute vector to each vertex.

A fuzzy graph is an extension of classical graph theory that incorporates concepts from fuzzy set theory to represent uncertainty and partial relationships between elements.

Definition 4 (Fuzzy Graph). A fuzzy graph is defined as $G = (V, E, \sigma, \mu)$, where V is the set of vertices, $E \subseteq V \times V$ is the set of edges, $\sigma : V \times V \to [0, 1]$ denotes a membership function that assigns a value to each edge, representing the degree of connectivity or strength of the relationship between two vertices, and μ ($\mu \in [0, 1]$) indicates a membership function that assigns a value to each vertex, indicating the degree of membership or importance of the vertex.

Definition 5 (Incomplete Graph). An incomplete graph can be represented as $\mathcal{G} = (V, E, p)$, where V is the set of vertices, $p : E \to \{1, ?, 0\}$ is a function that assigns the signs to the edges of the graph.

Definition 6 (Node Centrality). The number of edges directly connected to node x_i, and it also known as the number of nodes of node x_i:

$$c_D(i) = \sum_{j=1}^{N} a_{ij}.$$

The node with the largest node centrality is the network center, and the above formula is the absolute centrality of the node. The relative centrality of the node can also be obtained, that is, the ratio of the number of nodes of the node to the maximum possible total number of connections. The relative centrality of node x_i can be expressed as:

$$c'_D(i) = \frac{c_D(i)}{N - 1},$$

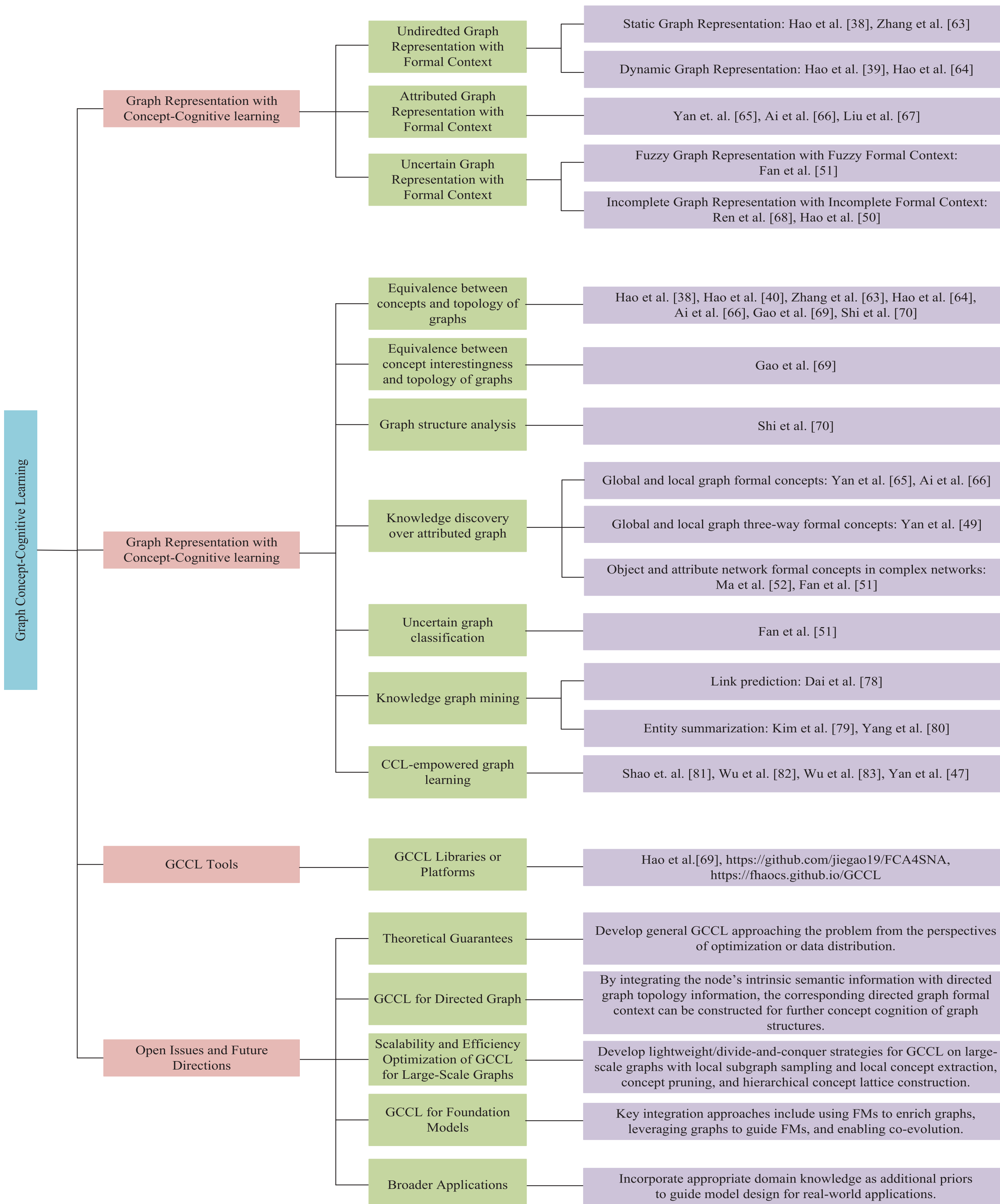


Fig. 1. Research overview of GCCL.

where N denotes network size, and the maximum degree of any node is $N-1$.

Definition 7 (Central Potential).

$$c_D' = \frac{\sum_{i=1}^{N}[c_{D_{max}} - c_D(i)]}{N-1},$$

where $c_{D_{\max}} = max\{c_D(i)\}$ denotes the largest value of the centrality of all nodes. The central potential reflects the importance difference within the network. The greater the c_D', the greater the importance difference between nodes in the network. On the contrary, the importance difference between nodes in the network is small.

2.2. Concept-cognitive learning

Concept-Cognitive Learning (CCL) is a data analysis method developed in combination with fields such as machine learning and cognitive computing, based on Formal Concept Analysis. It is used to simulate the cognitive processes involved in memory [21], learning [53], thinking, and activation [54], and aims to represent high-level knowledge structures through concept analysis. CCL serves as an important tool for studying how different cognitive learning methods can represent and analyze conceptual structures.

Due to its ability to mimic human learning mechanisms from a conceptual structure perspective, CCL has received widespread attention and research in recent years. The classic examples are listed as follows:

- Semantic analysis of images based on symbolic concept networks [55];
- Knowledge discovery using the conceptual structure-based abilities of CCL to uncover logical and decision relationships [56,57];
- Obtaining system information from an ontological perspective to effectively discover implicit needs and contradictions, while exploring natural language analysis based on ontology and three-way concept stability [58];
- Designing and implementing perceptual system models for formal contexts, utilizing CCL's ability to discover and recognize knowledge in unknown domains [59];
- Social network analysis for social media mining and human behavior modelling [41,60–62].

The concept-cognitive process is a computational process based on the concept-cognitive system, using specific cognitive models such as granular computing system models and information similarity matching to obtain corresponding concepts from given clues. Its purpose is to simulate the brain's concrete implementation of concept learning. As shown in Fig. 2, it represents the CCL process using multi-source heterogeneous data. The process begins with different data sources, each undergoing data transformation to convert raw data into a standardized form. From the standardized data, clues are extracted, which are then used to form individual sub-concepts. These sub-concepts, derived from the various data sources, are combined through concept-cognitive fusion, resulting in a unified and comprehensive resulting concept. This final concept reflects the integration of knowledge from multiple heterogeneous sources, simulating the way humans process and learn concepts.

FCA as a typical concept-cognitive learning model, has been widely used for data analysis and mining [38]. Importantly, this methodology could efficiently characterize the relationships between objects and attributes via formal concepts which are represented and formed with objects as well as attributes. Central to FCA are the notations of formal context, concept and concept lattice, thus this section mainly provides the basics of FCA.

Definition 8 (Formal Context). A formal context is formally represented as a three tuple, denoted by $K = (V, A, I)$, which is composed of a set of formal objects $V = \{v_1, v_2, \cdots, v_n\}$, a set of formal attributes $A = \{a_1, a_2, \cdots, a_m\}$ and the binary relation $I \subseteq V \times A$ between objects and attributes. Here, if $(v_i, a_j) \in I$, then it implies that the object v_i has the attribute a_j; and if $(v_i, a_j) \notin I$, then it implies that v_i does not have the attribute a_j, where $v_i \in V$ and $a_j \in A$. We normally mark the possessing relation between objects and attributes with "×" or "1", and the non-possessing relation between objects and attributes with " " or "0".

Table 1
A Formal Context.

V/A	a_1	a_2	a_3	a_4	a_5
v_1	×	×		×	×
v_2	×	×	×		
v_3					×
v_4	×	×	×		

Definition 9 (Operators ↑ and ↓). For a given formal context $K = (V, A, I)$, FCA methodology defines two key operators ↑ and ↓ for extracting the set of common attributes and the set of common objects, respectively. Formally, they are defined as:

$$X^{\uparrow} = \{a \in A | \forall x \in X, (x, a) \in I\}, \tag{2}$$

$$B^{\downarrow} = \{x \in V | \forall a \in B, (x, a) \in I\}, \tag{3}$$

where $X \subseteq V$ and $B \subseteq A$.

Definition 10 (Formal Concept). For a formal context $K = (V, A, I)$, if a pair (X, B) satisfies $X^{\uparrow} = B$ and $B^{\downarrow} = X$, then the pair (X, B) is a concept, where X is the extent of the concept, and B is the intent of the concept.

Definition 11 (Super-concept and Sub-concept). Given a formal context $K = (V, A, I)$, and two concepts $C_1 = (X_1, B_1)$ and $C_2 = (X_2, B_2)$. The partial order $C_1 \le C_2$ indicates that concept C_2 is the super-concept of concept C_1, and C_1 is the sub-concept of C_2. Formally, the following inequations holds.

$$(X_1, B_1) \le (X_2, B_2) \Leftrightarrow X_1 \subseteq X_2 (\Leftrightarrow B_1 \supseteq B_2), \tag{4}$$

then, we call "≤" is a partial relation of $C(K)$.

Definition 12 (Concept Lattice). Formally, a concept lattice, represented as $L = (C(K), \le)$, can be obtained by all formal concepts $C(K)$ of a context K with partial order "≤". Its graphical representation is a Hasse diagram.

Example 1. For a given formal context K which contains 4 objects $\{v_1, v_2, v_3, v_4\}$ and 5 attributes $\{a_1, a_2, a_3, a_4, a_5\}$. The binary relations between objects and attributes are shown in Table 1.

By using concept lattice generation algorithm and lattice visualization software *Galicia* [1], the corresponding concept lattice is shown in Fig. 3.

As can be seen from Fig. 3, each node indicates a formal concept including extent and intent, such as a concept $(\{v_1, v_2, v_4\}, \{a_1, a_2\})$, the extent is $\{v_1, v_2, v_4\}$ and intent is $\{a_1, a_2\}$. It is interpreted as the objects $\{v_1, v_2, v_4\}$ own the common attributes $\{a_1, a_2\}$. These formal concepts follow the partial order within the concept lattice.

2.3. The framework on graph concept-cognitive learning

Graph concept-cognitive learning (GCCL) is a special CCL paradigm on graph data. The core idea of GCCL is to implement the graph cognition tasks based on FCA, and further provide the relevant graph-related applications, such as social media mining, anomaly detection, community discovery and knowledge graph.

Fig. 4 shows a GCCL framework which is divided into four layers from bottom to top, namely the representation layer, concept layer, cognition layer, and application layer.

The specific functions of each layer are as follows:

[1] http://www.iro.umontreal.ca/~galicia/

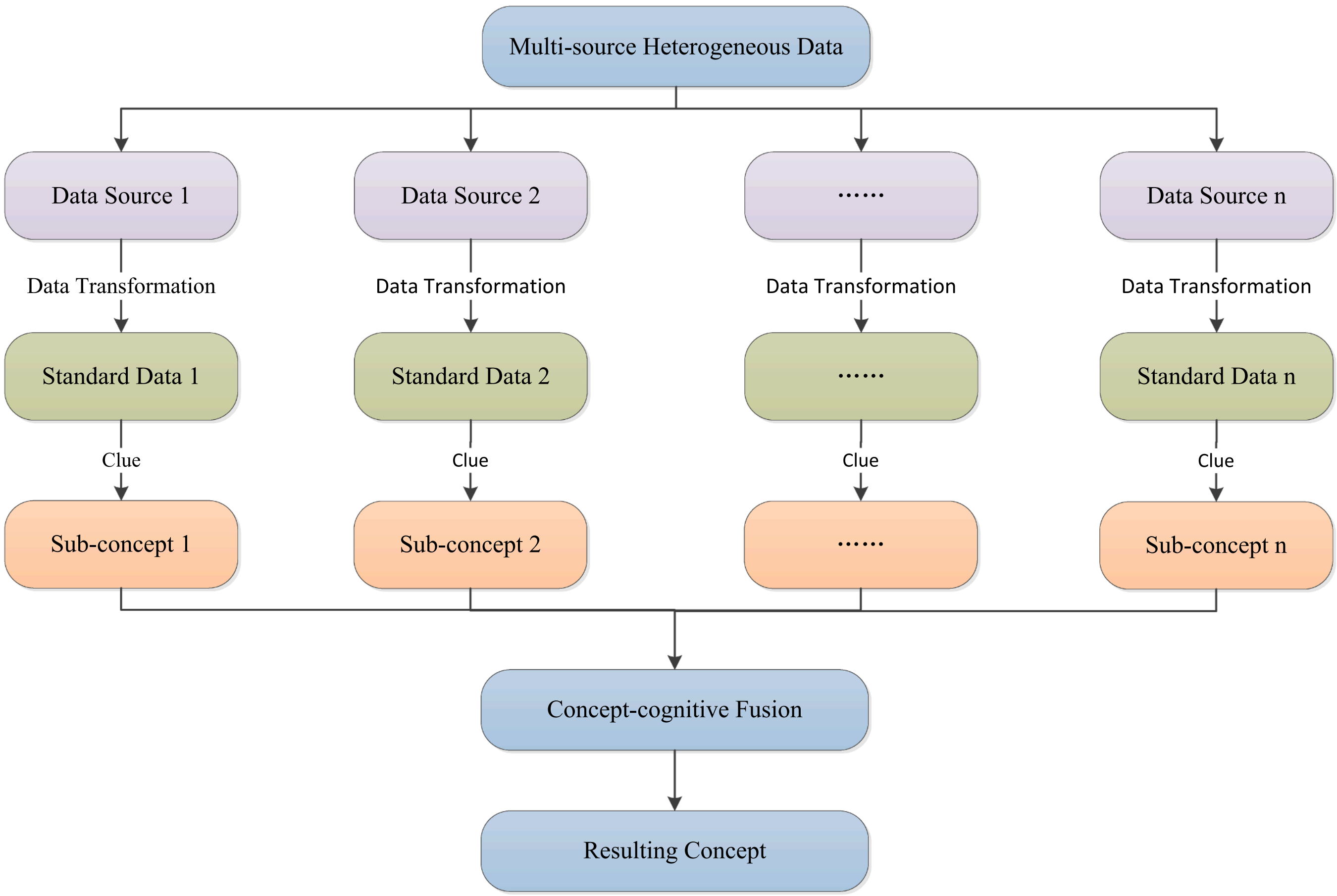


Fig. 2. The principle of CCL.

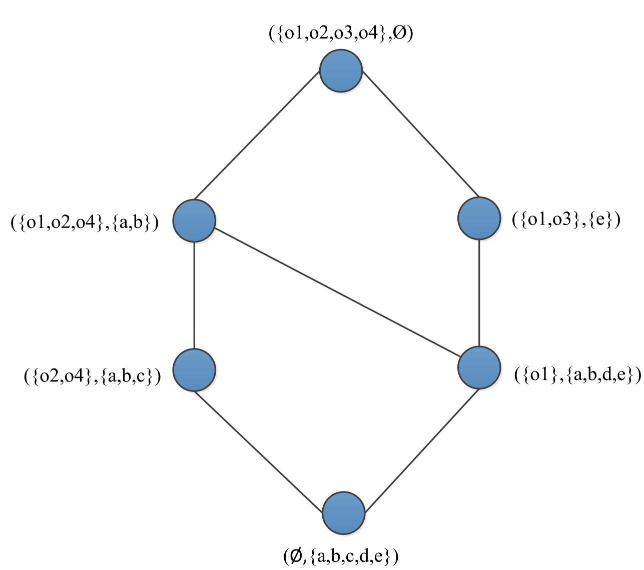


Fig. 3. The concept lattice of formal context K.

- The representation layer is responsible for representing a given graph in the form of a modified adjacency matrix, which serves as the formal context required by the FCA model.
- The concept layer is in charge of generating a concept lattice from the formal context. At this layer, the graph structure is represented in the form of concepts.
- The cognition layer is responsible for learning the implicit knowledge hidden in the concept lattice, including intuitive knowledge that is easy to obtain, such as extent and intent, as well as hidden knowledge obtained through concept-cognitive learning metrics like concept interestingness.
- The application layer applies the concept cognition results from the cognition layer to various specialized application domains, such as structure recognition, social recommendation, and data mining.

3. Graph representation with concept-cognitive learning

In this section, we mainly discuss the graph representation approaches with concept-cognitive learning for various graphs.

3.1. Undirected graph representation with formal context

In this section, we will present the representation approaches for static graph and dynamic graph, respectively.

3.1.1. Static graph representation

Generally, a static graph is represented as $G = (V, E)$ where V indicating the set of vertices and E indicating the set of relationships between vertices. Note that, the vertices are regarded as both objects and attributes in the constructed formal context. Hence, a formal context of a static network can be formalized as $K^G = (V, V, I)$ by the following modified adjacency matrix, in which I is the binary relationship between vertices [38,63].

Definition 13 (Modified Adjacency Matrix)**.** Let G be a graph with n vertices that are assumed to be ordered from v_1 to v_n. The $n \times n$ matrix

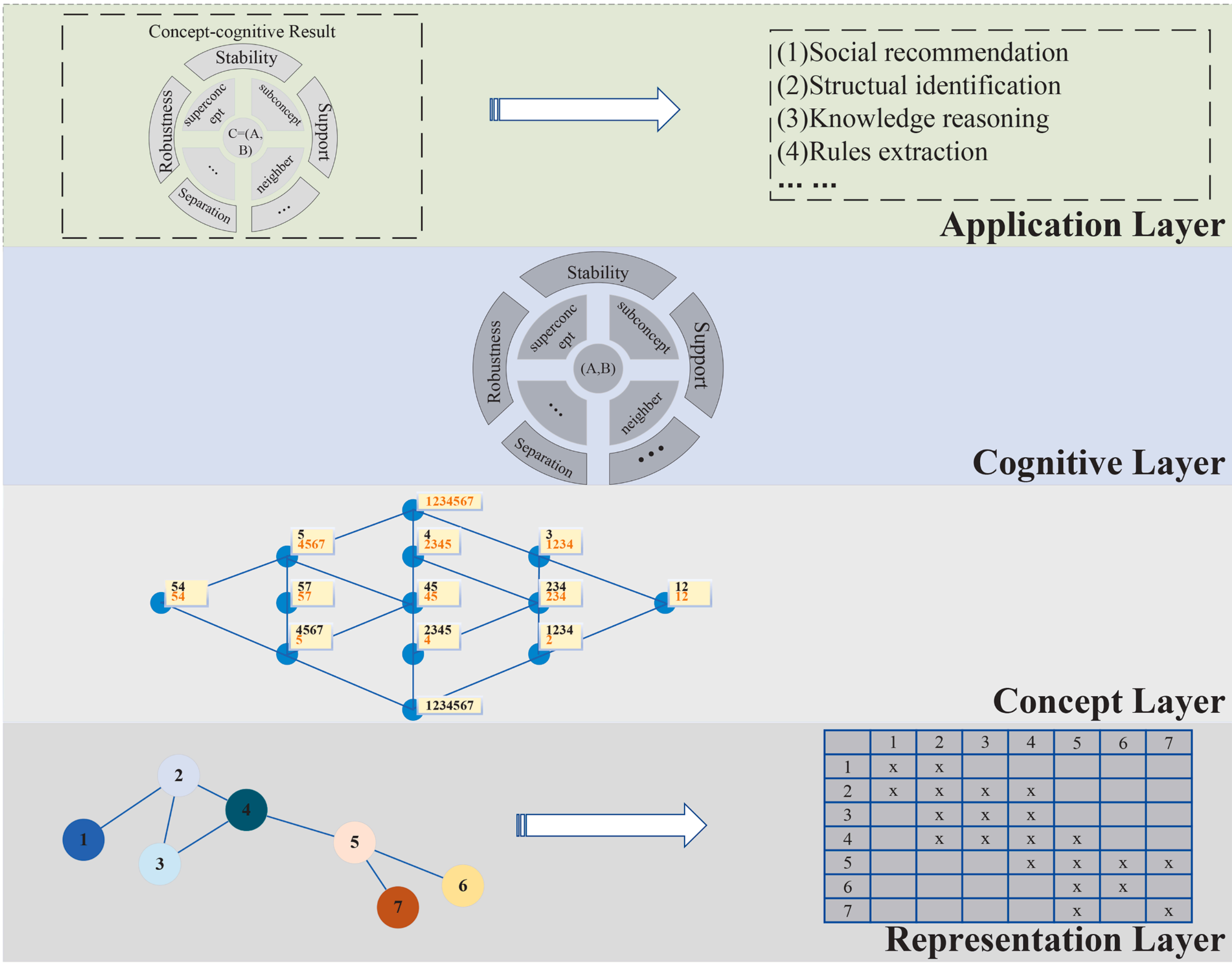


Fig. 4. The framework of GCCL.

A' is called a modified adjacency matrix, in which

$$A' = \begin{cases} a_{ij} = 1, & \text{if } (v_i, v_j) \in E, \\ a_{ij} = 1, & \text{if } i = j, \\ a_{ij} = 0, & \text{otherwise.} \end{cases}$$

Hence, K^G is equivalent to the modified adjacency matrix of G, i.e., $K^G \equiv A'$. Besides, a formal context of a social network K^G has the following properties.

- K^G is symmetry. The social relationships between individual are reciprocal, the relationships are naturally symmetry.
- All the diagonal elements are marked with “1”.

We would like to demonstrate the representation procedure for a given graph with formal context via the following Example 2.

Example 2. Fig. 5 shows a graph G with vertices indicating vertices and edges indicating the relationships between vertices, and a formal context of G is constructed in Table 2 according to Definition 13.

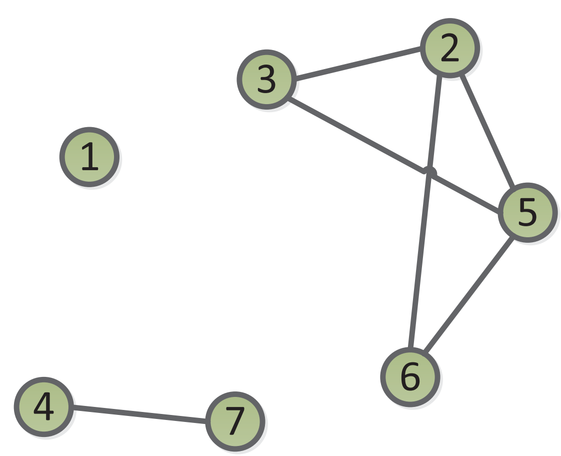


Fig. 5. A graph G.

3.1.2. Dynamic graph representation

Thanks to the Triadic Formal Concept Analysis, the dynamic graph representation is represented as triadic formal context, i.e., $K^{G_t} = (V, V, T, I)$, in which the set of vertices V is regarded as both objects and attributes, time T is viewed as the condition, and I denotes the connections between vertices at time T [39,64]. For any two vertices $v_i, v_j \in V$, if there exists a connection between them, then $I(v_i, v_j) = 1$, otherwise, $I(v_i, v_j) = 0$;

Table 2
A Formal Context of Graph G.

User	v_1	v_2	v_3	v_4	v_5	v_6	v_7
v_1	1	0	0	0	0	0	0
v_2	0	1	1	0	1	1	0
v_3	0	1	1	0	1	0	0
v_4	0	0	0	1	0	0	1
v_5	0	1	1	0	1	1	0
v_6	0	1	0	0	1	1	0
v_7	0	0	0	1	0	0	1

Table 3
An Attribute Graph Formal Context (V, A, A', I).

User	v_1	v_2	v_3	v_4	v_5	v_6	v_7	a_1	a_2	a_3	a_4	a_5
v_1	1	0	0	0	0	0	0	1	1	1	0	0
v_2	0	1	1	0	1	1	0	1	1	0	1	0
v_3	0	1	1	0	1	0	0	0	0	1	1	0
v_4	0	0	0	1	0	0	1	1	0	0	1	1
v_5	0	1	1	0	1	1	0	0	1	1	1	1
v_6	0	1	0	0	1	1	0	0	1	0	1	1
v_7	0	0	0	1	0	0	1	0	1	1	0	1

Table 4
A Graph Formal Context.

	A'				⋯	A^K				A			
User	v_1	v_2	⋯	v_n	⋯	v_1	v_2	⋯	v_n	a_1	a_2	⋯	a_m
v_1	1	0	⋯	0	⋯	0	0	⋯	1	1	0	⋯	1
v_2	0	1	⋯	0	⋯	1	0	⋯	1	0	1	⋯	1
⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯
v_n	0	0	⋯	1	⋯	0	1	⋯	0	0	1	⋯	1

3.2. Attributed graph representation with formal context

As mentioned above, an attributed graph is represented as $G = (V, E, A)$, where V is the set of vertices, $E \subseteq V \times V$ is the set of edges, and A: $V \to \mathbb{R}^d$ assigns a d-dimensional attribute vector to each vertex. Here, we express it in the form of formal context[65–67].

Definition 14 (Attributed Graph Formal Context)**.** A quadruples (V, A, A', I) is an attributed graph formal context, where $V = \{v_1, v_2, \cdots, v_n\}$ denotes an object set, $A = \{a_1, a_2, \cdots, a_m\}$ denotes an attribute set, A' denotes a modified adjacency matrix between objects and the binary relation $I \subseteq V \times A$ between objects and attributes.

We also give an example to help understand attributed graph formal context.

Example 3. In Fig. 5, we suppose each object has its attribute information, and 1–7 have attribute $a_1a_2a_3$, $a_1a_2a_4$, a_3a_4, $a_1a_4a_5$, $a_2a_3a_4a_5$, $a_2a_4a_5$, $a_2a_3a_5$, respectively. According to Definition 14, we can represent these information as an attributed graph formal context in Table 3.

It can be simply understood from Table 3 that the graph formal context is the simple splicing of the modified adjacency matrix and the formal context.

Definition 14 only considers the first-order adjacency matrix. In order to better mine the hidden information between them, the adjacency matrix can also be extended to the k-order case. Generally expressed as Table 4.

3.3. Uncertain graph representation with formal context

This section is devoted to presenting the representation approaches for fuzzy graph and incomplete graph, respectively.

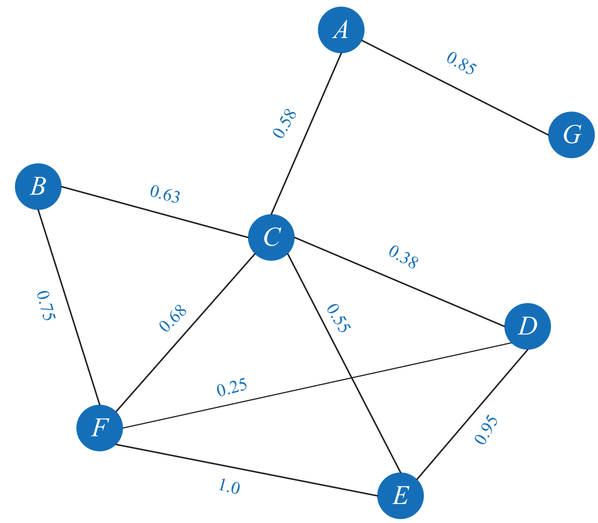


Fig. 6. A typical fuzzy graph.

Table 5
The fuzzy formal context for Fig. 6.

Node	A	B	C	D	E	F	G
A	1	0.58	0	0	0	0	0.85
B	0.58	1	0.63	0	0	0.75	0
C	0	0.63	1	0.38	0.55	0.68	0
D	0	0	0.38	1	0.95	0.25	0
E	0	0	0.55	0.95	1	1.0	0
F	0	0.75	0.68	0.25	1.0	1	0
G	0.85	0	0	0	0	0	1

3.3.1. Fuzzy graph representation with fuzzy formal context

Similar to the static graph representation, we take the nodes as the objects and attributes and adopt the fuzzy relation matrix R^* to construct the fuzzy formal context of the given fuzzy graph $\mathcal{G}$, denoted as $FFC(\mathcal{G}) = (V, V, R^*)$ [51].

Definition 15 (Fuzzy Relation)**.** A fuzzy matrix $R^* = (r_{ij})_{m \times n}$ is a matrix with dimension $m \times n$, if

$$R^* = \begin{cases} r_{ij} = \mu(e_{ij}), & \text{if there exists an edge from } v_i \text{ to } v_j \text{ and } i \neq j, \\ r_{ij} = 1, & \text{if } i = j, \\ r_{ij} = 0, & \text{otherwise,} \end{cases} \tag{5}$$

where, $r_{ij} = \mu(e_{ij})$, i.e., r_{ij} is the degree of membership between v_i and v_j.

Hence, $FFC(\mathcal{G})$ is equivalent to the fuzzy matrix of $\mathcal{G}$, that is to say $FFC(\mathcal{G}) \equiv R^*$.

Example 4. Fig. 6 shows a fuzzy graph containing 7 vertices and edges with degrees of membership. All vertices $\{A, B, C, D, E, F, G\}$ are considered as both objects and attributes, and the weight (membership) values on the edges are in fuzzy form. The relationship value between objects and attributes in the formal context. Finally, the fuzzy formal context construction of Fig. 6 is shown in Table 5.

3.3.2. Incomplete graph representation with incomplete formal context

An incomplete graph can be represented as $G^p = (V, E, p)$, where V is the set of vertices, $p : E \to \{1, ?, 0\}$ is a function that assigns the signs to the edge of the graph. Obviously, the incomplete graph is quite different from the other types of graph, here is a natural question, is there any solution for representing the incomplete graph by formal context?

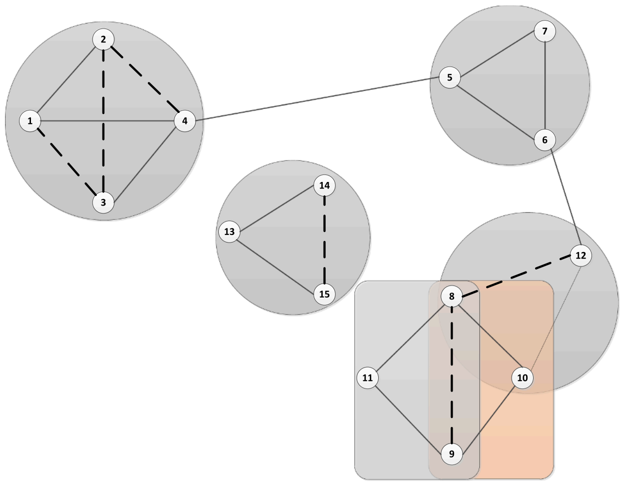


Fig. 7. A typical incomplete graph.

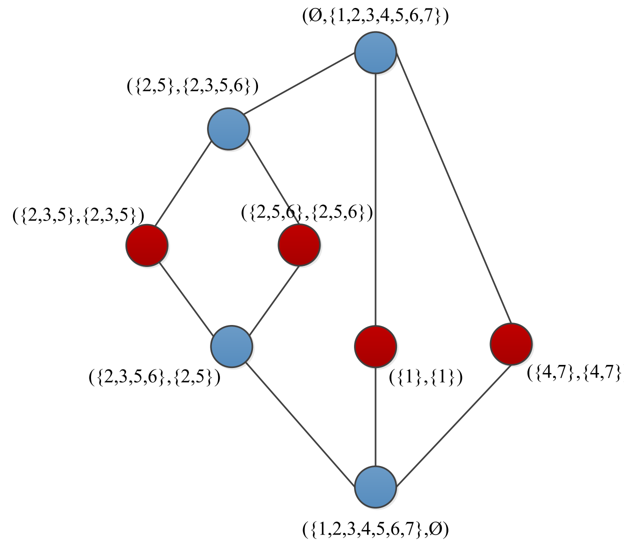


Fig. 8. Concept lattice of graph G.

Fortunately, Partially-known Concept Analysis methodology defines the incomplete formal context which can be well characterize the incomplete graph [50,68].

Definition 16 (Incomplete Formal Context)**.** An incomplete formal context can be represented as a 4-tuple $IK = (V, A, \{+, ?, -\}, I)$, where $V = \{x_1, x_2, \cdots, x_n\}$ is a set of objects, $A = \{a_1, a_2, \cdots, a_m\}$ is a set of attributes, and I is a mapping: $O \times A \rightarrow \{+, ?, -\}$. To be specific, $I(v, a) = +$ indicates the object v has the attribute a; $I(v, a) = -$ denotes the objects v does not have the attribute a; and $I(v, a) =?$ is unknown whether or not the object v has the attribute a. In general, we write $\{+, ?, -\}$ as E, then $IK = (V, A, \{+, ?, -\}, I)$ is rewritten as $IK = (V, A, E, I)$.

Similar with the approach for a graph representation based on FCA, an incomplete graph G^p can be represented by the incomplete formal context of G^p by taking the vertices as the objects and attributes, as well as the edges as the relations, namely $IK(G^p) = (V, V, \{+, ?, -\}, I)$, in which I denotes the relations between two vertices. For convenience, we replace "$+$" and "-" with "1" and"0", i.e., $IK(G^p) = (V, V, \{1, ?, 0\}, I)$.

To better understand how does it work for representing incomplete graph with incomplete formal context, the following example is given.

Example 5. Fig. 7 shows an incomplete graph including 15 vertices, and dotted lines refer to the uncertain links between vertices and solid lines are the links between vertices.

Technically, the vertices $v_1, v_2, \cdots, v_{15}$ are taken as the objects and attributes, and the edges are viewed as the relations, in the constructed incomplete formal context as shown in Table 6.

Clearly, this incomplete formal context contains 12 notations "?", that is to say, there exist 6 unknown edges in the G^p since G^p is an undirected graph.

4. Graph cognition with concept-cognitive learning

In this section, we first present two important equivalence relations between concepts/concept stability and topology of graphs. Then, the methodologies on graph cognition tasks including key structures identification, link prediction, community detection and other graph-related applications in undirected graphs, directed graphs, and uncertain graphs are briefly elaborated.

4.1. Equivalence between concepts and topology of graphs

To bridge two disciplines, i.e, Graph data mining and FCA technically, Hao et.al. [38] firstly introduced an important definition, called Equiconcept, which has the same extent and intent. They proved that the Equiconcepts are equivalent to the cliques in the graph. It is also proved that all cliques detection is composed of basic cliques obtained from the equiconcepts as well as derived cliques obtained from other high-order equiconcepts. With these findings, Hao et al. [40] proved that the equiconcepts are exactly mapping to the maximal cliques in the graph.

Example 6. Let us take Fig. 5 as an example. The corresponding concept lattice is presented as shown in Fig. 8. Clearly, 4 equiconcepts $(\{v_4, v_7\}, \{v_4, v_7\})$, $(\{v_2, v_{5,6}\}, \{v_2, v_5, v_6\})$, $(\{v_2, v_3, v_5\}, \{v_2, v_3, v_5\})$, $(\{v_1\}, \{v_1\})$ are identified from the concept lattice of social network G since their extent equals to intent. That is to say, we can easily obtain the maximal cliques, i.e., $(\{v_4, v_7\}, \{v_4, v_7\})$, $(\{v_2, v_5, v_6\}, \{v_2, v_5, v_6\})$, $(\{v_2, v_3, v_5\}, \{v_2, v_3, v_5\})$, $(\{v_1\}, \{v_1\})$.

4.2. Equivalence between concept interestingness and topology of graphs

In practice, real data sets may generate a large number of formal concepts and contain noisy data, which limits the effective use of FCA technology. Concept interestingness is an important metric used to prune concept lattices and obtain accurate and effective information. The concept interestingness mentioned in this paper includes stability and separation.

First, concept stability measures the degree of dependence of a concept on some special objects in its extent set, that is, the probability that the intent of a concept remains consistent when the extent set of the concept excluding some elements. The calculation of concept stability for a formal concept (A, B) requires traversing all subsets P of the extension set A in turn and checking whether the operation $P^{\uparrow}$ on it is equal to the intent set B. Gao et.al [69] proposed a novel approach for handling key structures identification tasks by concept interestingness. Specifically, they introduced stability and separation to interpret the hidden association between the concept and the key structures. Then, the mapping relations between particular concepts and key structures are proved and obtained as shown in Table 7, where σ is the intentional stability, that is, the probability of keeping the intent fixed in the case of disturbing the extent.

Example 7. Fig. 9 shows a simple graph including 15 vertices.

According to Definition 13, the following graph formal context (i.e., Table 8) can be easily constructed.

Table 6
Incomplete formal context of G^p:$IK(G^p) = (V, V, \{+, ?, -\}, I)$.

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}	v_{11}	v_{12}	v_{13}	v_{14}	v_{15}
v_1	1	1	?	1	0	0	0	0	0	0	0	0	0	0	0
v_2	1	1	?	?	0	0	0	0	0	0	0	0	0	0	0
v_3	?	?	1	1	0	0	0	0	0	0	0	0	0	0	0
v_4	1	?	1	1	1	0	0	0	0	0	0	0	0	0	0
v_5	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0
v_6	0	0	0	0	1	1	1	0	0	0	0	1	0	0	0
v_7	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0
v_8	0	0	0	0	0	0	0	1	?	1	1	?	0	0	0
v_9	0	0	0	0	0	0	0	?	1	1	1	0	0	0	0
v_{10}	0	0	0	0	0	0	0	1	1	1	0	1	0	0	0
v_{11}	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0
v_{12}	0	0	0	0	0	1	0	?	0	1	0	1	0	0	0
v_{13}	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
v_{14}	0	0	0	0	0	0	0	0	0	0	0	0	1	1	?
v_{15}	0	0	0	0	0	0	0	0	0	0	0	0	1	?	1

Table 7
The Mapping Relational Table.

Key Structure	Concept (A, B)
Maximal clique	$A = B$
Isolated maximal clique	$A = B, \sigma = \frac{2^{\lvert A \rvert}-1}{2^{\lvert A \rvert}}, \xi = 1$
Bridge	$A = B, \sigma = 0.25, \xi \leq 0.5$ $A = B, \sigma = 0.5, \xi \leq 2/3$
Structural hole	$\lvert A \rvert = 1$, $\sigma = 0.5$, $\xi \leq 0.5$

σ: the intentional stability
$\lvert A \rvert$: the element number of the object set
ξ: the separation index

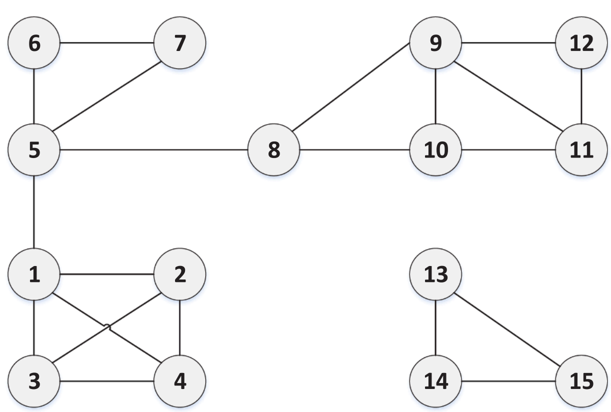


Fig. 9. A simple graph G.

4.3. Graph structure analysis

Shi et al. [70] focused on investigating the relationships between structure of graphs and formal concepts and further providing alternative approach for graph topology analysis. The following are the details.

4.3.1. Cliques detection

In graph theory, a clique represents a fundamental concept defined as a complete subgraph within an undirected graph. This cohesive structure requires that every pair of distinct vertices within it must be directly connected by an edge. The detection and analysis of cliques are computationally challenging yet vital for understanding complex network properties, with practical applications extending to community detection in social networks, relationship prediction, and analysis of biological systems such as protein interactions.

Definition 17 (Clique)**.** In a graph G, a clique is a subset $S \subset V$ such that for any two vertices $v_i, v_j \in S$, there exists an edge $(v_i, v_j) \in E$. Obviously, a clique is a complete connected subgraph in the graph.

Definition 18 (k-Clique)**.** In a graph G, a clique is a subset $S \subset V$ and $|S| = k$ such that for any two vertices $v_i, v_j \in S$, there exists an edge $(v_i, v_j) \in E$.

Theorem 1 ([38])**.** *Given a graph G, the k-cliques detection problem is equivalent to finding the set of k-equiconcepts from the concept lattice of G.*

Lemma 1. *Let (A, B) be a k-equiconcept, the number of derived (k-1)-cliques from (A, B) equals C_k^{k-1}.*

Example 8. In Example 2, we can easily find the four equiconcepts, *i.e.*, $(\{1\}, \{1\}), (\{4, 7\}, \{4, 7\}), (\{2, 5, 6\}, \{2, 5, 6\})$ and $(\{2, 3, 5\}, \{2, 3, 5\})$. Actually, in the social network G, these equiconcepts correspond to 1-clique, 2-clique, 3-clique, and 3-clique, respectively. In fact, we can derive more 2-cliques from $(\{2, 5, 6\}, \{2, 5, 6\})$ and $(\{2, 3, 5\}, \{2, 3, 5\})$, such as $(\{2, 3\}, \{2, 3\})$, $(\{2, 5\}, \{2, 5\})$, $(\{2, 6\}, \{2, 6\})$, $(\{3, 5\}, \{3, 5\})$, and $(\{5, 6\}, \{5, 6\})$. But, they are not concepts, thus they do not appear in the concept lattice of graph G.

Theorem 2. *Given a graph G, all k-cliques detection is composed of following parts: a) basic cliques are generated from the k-equiconcepts; b) remaining cliques are derived from the $(k+1)$-equiconcepts, $(k+2)$-equiconcepts, $\cdots$, M-equiconcepts. ($M > k$). M is the number of maximum extent or intent of maximum equiconcepts.*

4.3.2. Maximal cliques detection

The purpose of maximal clique detection is to enumerate all maximal cliques from a given graph. Importantly, enumerating the maximal clique has numerous potential applications in biology, chemistry, sociology, and graph modeling [71,72].

Definition 19 (Maximal Clique [73])**.** A maximal clique is a clique that cannot be extended by including one more adjacent vertex. Formally, for a clique c if there exists no clique c' in G such that $c \subset c'$. Then, the clique c is called as maximal clique.

Theorem 3 (Equivalence theorem between maximal clique and equiconcept)**.** *Given a graph $G = (V, E)$, the maximal cliques existing in graph G exactly match with the equiconcepts in the formal concept lattice of G. Formally, an equivalence relation holds.*

$$maximal.cliques(G) \equiv EC(K^G), \quad (6)$$

where K is the formal context of G; $maximal.cliques(G)$ returns a set of maximal cliques of G; $EC(K^G)$ indicates the set of equiconcepts with respective to the formal context K.

Actually, a formal concept is a maximal pair of set of extent and its intent closed with Galois connection. That is to say, the equiconcept as a special formal concept, is also a maximal pair of set of extent and its intent (Note that, a maximal pair is corresponding to a maximal clique).

Table 8
Formal context of graph G (The number denotes the node ID). Construct formal context K of social network G.

G	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	×	×	×	×	×										
2	×	×	×	×											
3	×	×	×	×											
4	×	×	×	×											
5	×				×	×	×	×							
6					×	×	×								
7					×	×	×								
8					×			×	×	×					
9								×	×	×	×	×			
10								×	×	×	×				
11									×	×	×	×			
12									×		×	×			
13													×	×	×
14													×	×	×
15													×	×	×

Example 9. To better understand the proposed equivalence theorem. On one hand, we run the command "***maximal.cliques(g)***" in R language compiler. Eventually, we obtain 9 maximal cliques as shown in Fig. 10(b). On the other hand, a formal context of $g = (V, E)$ can be constructed, *i.e.*, $K^g = FC(g) = (V, V, I)$ by our proposed representation model. Then, the corresponding concept lattice $L(C(K^g), \leq)$ is built according to Definition 12. Then, the set of equiconcepts existing in L is denoted as $EC(K^g)$ as shown in Fig. 10(a).

As can be seen from Fig. 10, there exists an equivalence relation or one to one mapping from equiconcepts to maximal cliques. In other words, there exist 9 mappings that match the detected 9 equiconcepts with the mined 9 maximal cliques in the graph.

4.3.3. Clique communities detection

A k-clique community [74] is defined as the union of all k-cliques (i.e., complete subgraphs of size k), which can be reached from one or another through a series of adjacent k-cliques (where adjacency means sharing k-1 vertices).

This section introduces a novel FCA-based overlapping community detection method, termed the weak equiconcept for overlapping community detection (WEOCD) method [70]. Not only does the WEOCD method eliminate the requirement for pre-defined parameters, but it also effectively extracts more representative nodes as the seed set.

Definition 20 (Neighboring Concepts)**.** For the equiconcepts $H_1 = (A_1, B_1)$ and $H_2 = (A_2, B_2)$, if $A_1 \cap A_2 \neq \emptyset$, then H_1 and H_2 are called neighboring concepts.

Definition 21 (Weak Equiconcept)**.** In the network formal context (U, A, A', I), for neighboring concepts H_1 and H_2, if the following holds:

$$A_3 = A_1 \cup A_2, \tag{7}$$

$$B_3 = B_1 \cap B_2, \tag{8}$$

then $H_3 = (A_3, B_3)$ is referred to as the weak equiconcept.

Denote $\delta = \frac{|A_3|}{|B_3^{\downarrow}|}$, where $0 < \delta < 1$, and (A_3, B_3) is a weak equiconcept at the degree of δ.

Similarly, suppose (A_3, B_3) and (A_4, B_4) are weak equiconcepts. For the concept (C, D), $C \subseteq A_3 \cup X_4$, $D \subseteq A_3 \cap B_4$, if it follows that:

$$C^{\diamond} = \left\{ c_i \in U \mid 0 < \frac{|C|}{|D^{\uparrow}|} < \delta \right\}, \tag{9}$$

$$D^{\diamond} = \left\{ d_i \in B \mid 0 < \frac{|D|}{|C^{\downarrow}|} < \delta \right\}, \tag{10}$$

then concept (C, D) is also a weak equiconcept.

Where k is the number of objects in an equiconcept, a $(k+1)$-weak equiconcept is a parent concept of a k-equiconcept, and a k-equiconcept is a child concept of a $(k+1)$-weak equiconcept.

Definition 22 (Stability Coefficient)**.** For an equiconcept $H_1 = (A_1, B_1)$, the stability coefficient of the equiconcept is:

$$EC(H_1) = \frac{|A_1| \cdot (|A_1| - 1)}{\sum_{v_i \in A_1}^{|A_1|} \deg(v_i)}, \tag{11}$$

which is to measure the internal cohesion of the cluster formed by the objects of the equiconcept and the extent of connections outside the cluster.

The stability coefficient of the weak equiconcept $H_3 = (A_3, B_3)$ is:

$$EC_w(H_3) = \frac{2 \times links(H_3)}{\sum_{v_j \in A_3}^{|A_3|} \deg(v_j)}, \tag{12}$$

where $links(H_3)$ represents the number of edges among the objects within the concept H_3, and $\deg(v_i)$ represents the degree of the object v_i.

Property 1. *In the context of k-equiconcepts, a higher stability coefficient implies that the extent of the concept is more likely to belong to a singular community, leading to smaller communities with clearer boundaries. When these are used as seed sets, they impact the quality of the communities formed during expansion. Conversely, a lower stability coefficient results in more ambiguous community boundaries, and objects within these concepts are more likely to be part of multiple communities.*

Property 2. *The formation of a $(k+1)$-weak equiconcept, constructed from k-equiconcepts, is characterized by changes in the stability coefficient. An increase in the stability coefficient indicates that the newly added objects strengthen the connections among existing internal objects, thereby enhancing the clarity of community boundaries in the resulting communities. Conversely, a decrease in the stability coefficient suggests that the addition of new objects negatively impacts the stability of the internal community, leading to more ambiguous community structures.*

Example 10. Fig. 11 is a social network, which has 10 users and the development of initial and final communities, highlighting friendships

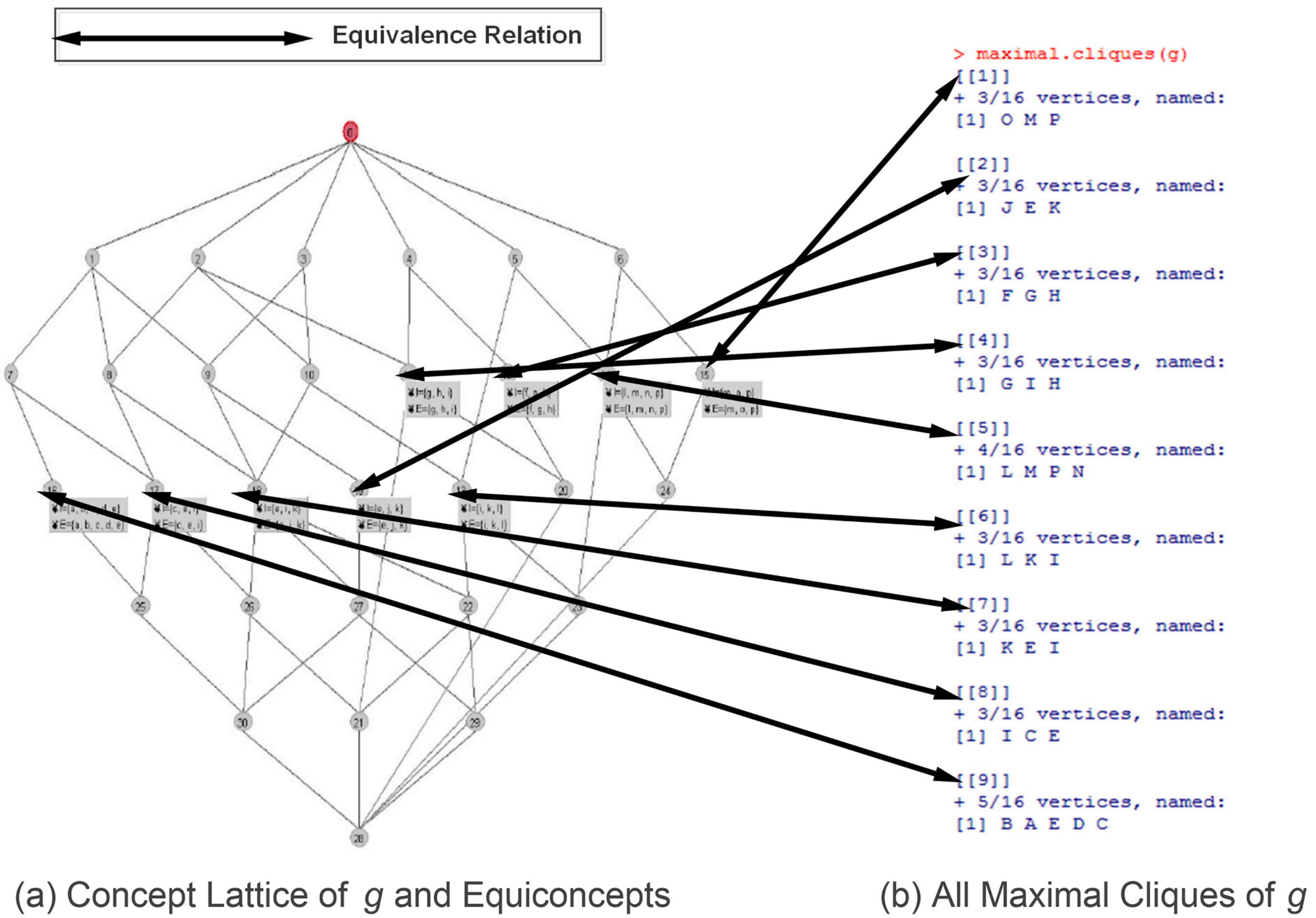


Fig. 10. Equivalence relation between equiconcepts and maximal cliques.

among these users through connecting edges. Fig. 12 presents the process of the formal context transformation in Fig. 8. Here, the symbol "1" in the table indicates that users 1 and 2 are first-order adjacent. At the same time, "0" means that there is no direct connection between them, suggesting that users 1 and 2 are not first-order adjacent. Fig. 13 is the corresponding Hasse diagram.

Fig. 12(a) indicates its corresponding simplified network formal context, considering only the 1-th association matrix A'. The equiconcepts in Fig. 12(b) are generated by definitions. We see the extent of each equiconcept as an initial seed set. Objects 1, 3, and 4 are part of more than one seed set, with object 3 belonging to three seed sets. Based on the decomposition of the concept subspaces in Fig. 10, we observe changes in the stability coefficient. Equiconcepts 1 and 2 merge into weak equiconcept 7, with the stability value of concept 7 being higher than both 1 and 2, thus selecting concept 7. Similarly, concepts 1 and 3 form weak equiconcept 11, which is also a weak equiconcept of concepts 7 and 8. However, its stability coefficient is lower than that of concept 7, so concept 7 is chosen. Weak equiconcept 12, formed by concepts 9 and 10, is selected, along with concept 6. In Fig. 11(c), there are three small communities in different colors, which are red, yellow, and green, and they are composed of objects from concepts 7, 12, and 6, respectively. They form three seed sets. Object 3, functioning as an overlapping node in two seed sets, showcases its diverse social connections by belonging to two different social circles.

Fig. 14 depicts the workflow of the WEOCD method, which consists of four key stages: building the network formal context, selecting seeds, expanding seeds, and optimizing communities. In the first stage, the algorithm creates an equivalent network formal context based on the attributes of a complex network. During seed selection, the most suitable seeds are chosen by filtering through the weak equiconcept. The seed expansion phase can be seen as an improvement to the clustering process initiated by forward propagation, where attribute preferences between nodes are incorporated to support seed growth. In the final stage, the algorithm identifies and refines overlapping communities by reducing conductance and information entropy within the network formal context.

Hao et al. [38] addressed the issue of detecting k-clique communities by introducing the k-intent concept and proposing a theorem for effective k-clique community detection. In other words, identifying k-clique communities is equivalent to finding the k-intent concepts, and the extents of each k-intent concept share at least $k-1$ vertices.

4.4. *Knowledge discovery over attributed graph*

4.4.1. *Global and local graph formal concepts*

In formal concepts, the relationship between objects and attributes is fully considered. In the graph, how to take the structure information between nodes into account is a problem worth exploring. Therefore, Yan et al.[65] defined graph formal concept.

Definition 23 (Global Graph Formal Concept)**.** Let (V, A, A', I) be a graph formal context, $X \subseteq V$ and $B \subseteq A$. If $X^{\uparrow} = B$, $B^{\downarrow} = X$, and X is connected, then the pair (X, B) is a global graph formal concept, where X is the extent of the concept, and B is the intent of the concept.

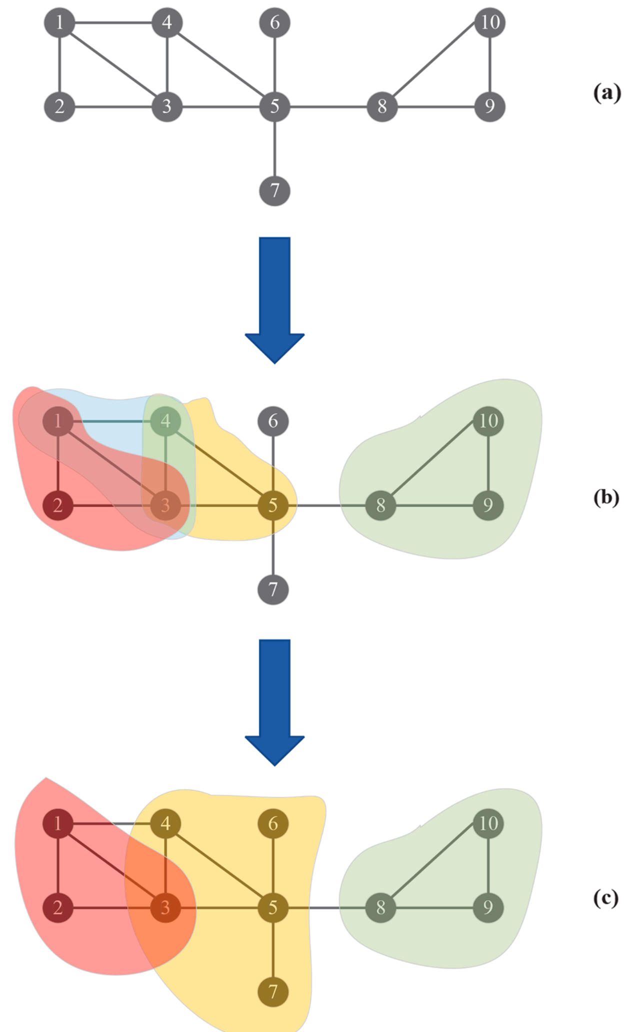


Fig. 11. In the social network graph of 10 users, the equiconcept evolves into a weak equiconcept.

	M_1									
	1	2	3	4	5	6	7	8	9	10
1	1	1	1	1	0	0	0	0	0	0
2	1	1	1	0	0	0	0	0	0	0
3	1	1	1	1	1	0	0	0	0	0
4	1	0	1	1	1	0	0	0	0	0
5	0	0	1	1	1	1	1	1	0	0
6	0	0	0	0	1	1	0	0	0	0
7	0	0	0	0	1	0	1	0	0	0
8	0	0	0	0	1	0	0	1	1	1
9	0	0	0	0	0	0	0	1	1	1
10	0	0	0	0	0	0	0	1	1	1

(a)

	{1,2,3}	{1,3,4}	{3,4,5}	{8,9,10}
{1,2,3}	3	2	1	0
{1,3,4}	2	3	2	0
{3,4,5}	1	2	3	0
{8,9,10}	0	0	0	3

(b)

	{1,2,3,4}	{3,4,5,6,7}	{8,9,10}
{1,2,3,4}	3	2	0
{3,4,5,6,7}	2	3	0
{8,9,10}	0	0	3

(c)

Fig. 12. The process of seed sets selection based on the changes in the network formal context corresponding to Fig. 11.

It should be pointed out that a formal concept can be regarded as a class, and considering the connectivity between objects based on the formal concept can only judge whether objects with the same attributes in a given graph formal context belong to the same class, which is equivalent to global connectivity. However, it is also common for partially connected objects with the same attributes to form the same class in graph formal context. If only global connectivity is considered, the conceptual knowledge containing local structure information will not be obtained. Therefore, the local graph formal context is further proposed.

Definition 24 (Local Graph Formal Concept)**.** Let (V, A, A', I) be a graph formal context, $X \subseteq V$ and $B \subseteq A$. If $X^{\uparrow} = B$, X is connected and and there is no connection between $a \in B^{\downarrow} - X$ and X, then the pair (X, B) is a local graph formal concept.

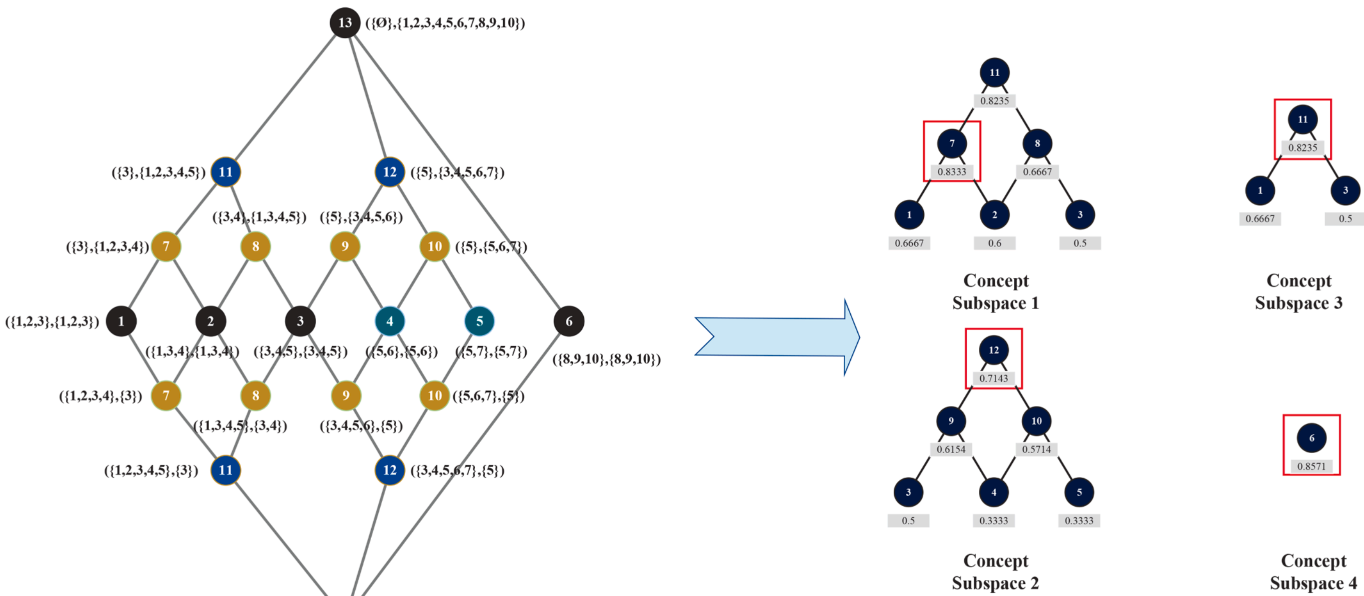


Fig. 13. The hasse diagram.

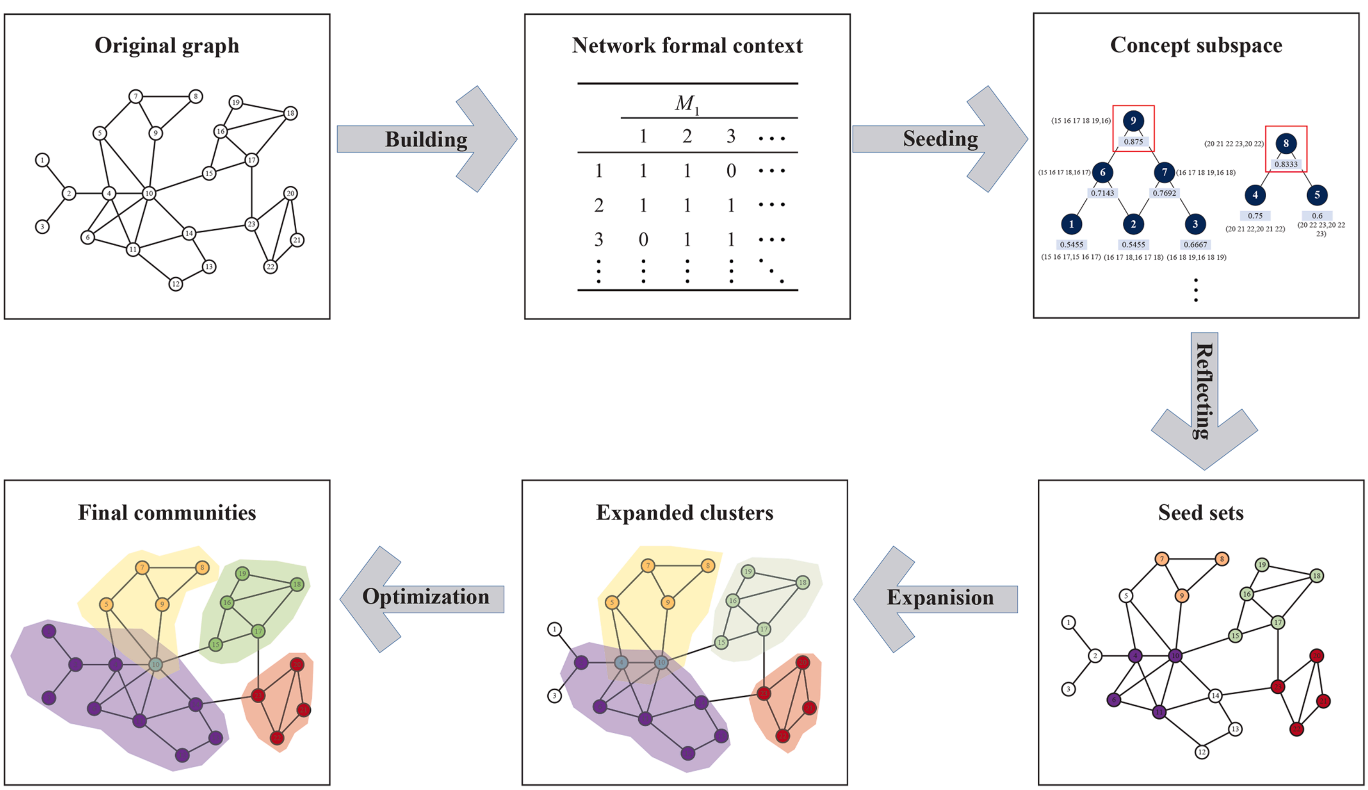


Fig. 14. WEOCD method overview diagram.

Definition 24 shows that the local graph formal concepts only need all the objects connected with X to be included in X, which can dig out the concept that is only established in the local range. In fact, the global graph formal concept is a concept with connectivity, while the local graph formal concept allows the disconnected objects in the concept to be free. The following is a specific example to illustrate the difference and connection between global graph formal concepts and local graph formal concepts.

Example 11. Fig. 15 is a diagram of an attributed graph, middle nodes 1–6 in subgraphs (a) and (b) represent 6 individuals, a, b, $\cdots$, e represents the interests of each individual, and the connected edges between objects represent the knowledge between two individuals. According to the definition of the formal concept, (134, a) and (126, b) satisfy the conditions of the formal concept, The red circle in figure (a) shows that the formal concept objects 1, 3 and 4 have a connected path, so the formal concept (134, a) is a global graph formal concept. In the yellow circle,

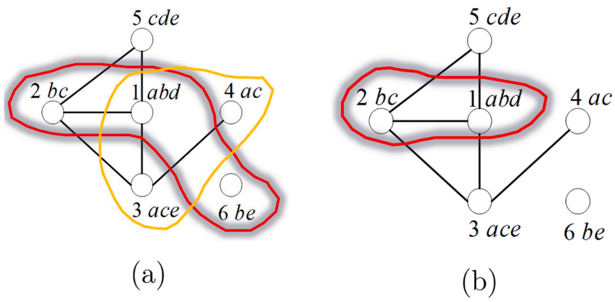


Fig. 15. The diagram of global and local graph formal concepts.

object 6 has no edge connection with objects 1 and 2, that is, objects 1, 2 and 6 do not have a connected path, so the formal concept (126, b) is not a global graph formal concept. However, if we only judge the global graph formal concept, we will lose a lot of local information. For example, although objects 1, 2, and 6 are not connected, objects 1 and 2 are connected, as shown in the red circle in figure (b), at this time, only considering the global connectivity will lose the local information, while the local graph formal concept makes up for this deficiency. (12, b) can constitute the local graph formal concept, so we can mine this local information, which makes full use of the attributed graph information to obtain the local concept knowledge.

4.4.2. Global and local graph three-way formal concepts

According to global and local graph formal concepts, Yan et al. [49] further proposed global and local graph three-way formal concepts.

Definition 25 ([22] Negative Operators)**.** Let $k = (V, A, I)$ be a formal context and $I^c = (V \times A) - I$. For any object set $x \subseteq V$ and attribute set $B \subseteq A$,

$$X^{\uparrow c} = \{a \in B | x I^c a, \forall x \in X\},$$

$$B^{\downarrow c} = \{x \in V | x I^c a, \forall a \in B\}.$$

Compared with the operator $\uparrow$ and $\downarrow$ in the formal context $K = (V, A, I)$ that embodies the semantics of "commonly possessed", the operators $\uparrow c$ and $\downarrow c$ reflect the semantics of "commonly unpossessed". Therefore, the operators $\uparrow c$ and $\downarrow c$ are called negative operators. For our purpose, the operators $\uparrow$ and $\downarrow$ are called positive operators.

Based on the positive and negative operators, the notions of three-way operators can be defined from the aspects of "commonly possessed" relationship and "commonly unpossessed" relationship between an object set and an attribute set.

Definition 26 ([75] OE-operators)**.** Let (V, A, I) be a formal context. For any $X \subseteq V$, $B_1, B_2 \subseteq A$, we define $X^< = (X^{\downarrow}, X^{\downarrow c})$, and $(B_1, B_2)^> = \{x \in V | x \in B_1^{\downarrow}, x \in B_2^{\downarrow c}\} = B_1^{\downarrow} \cap B_2^{\downarrow c}$. The operators $<$ and $>$ are called object induced three-way operators, or OE-operators for short.

Definition 27 ([75] Three-way Concepts)**.** Let (V, A, I) be a formal context, $X \subseteq V$, $B_1, B_2 \subseteq A$. If $X^< = (B_1, B_2)$ and $(B_1, B_2)^> = X$, then $(X, (B_1, B_2))$ is called object induced three-way concept, or OE-concept for short. Here X is the extent of the concept $(X, (B_1, B_2))$, and (B_1, B_2) is the intent of the concept $(X, (B_1, B_2))$. The set of all the OE-concepts of the formal context (V, A, I) is denoted as OEL(V, A, I).

The object-attribute relation and object-object relation are taken into account at the same time to extract global graph OE-concepts and local graph OE-concepts from a graph formal context through the connectivity in graph theory.

Definition 28 (Global Graph OE-concepts)**.** Let (V, A, A', I) be a graph formal context, $X \subseteq V$ $(X \neq \emptyset)$, $B_1, B_2 \subseteq A$ $(B_1, B_2 \neq \emptyset)$. If $X^< = (B_1, B_2)$, $(B_1, B_2)^> = X$ and X is connected, then the order pair $(X, (B_1, B_2))$ is called a global graph OE-concept. The set of all global graph OE-concepts in (V, A, A', I) is denoted as $N_{TA}(V, A, A', I)$.

Table 9
A graph formal context (V, A, A', I).

	x_1	x_2	x_3	x_4	x_5	x_6	a_1	a_2	a_3	a_4	a_5	a_6
x_1	1	1	0	1	1	0	0	0	1	1	0	0
x_2	1	1	1	0	0	1	0	1	1	1	0	1
x_3	0	1	1	0	1	0	1	0	0	1	1	0
x_4	1	0	0	1	0	1	0	0	1	0	1	1
x_5	1	0	1	0	1	0	1	1	0	0	0	0
x_6	0	1	0	1	0	1	1	0	0	0	1	0

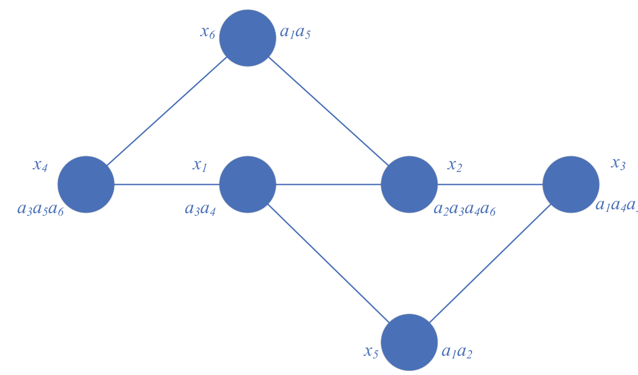


Fig. 16. The graphical representation of (V, A, A', I).

Definition 29 (Local Graph OE-concepts)**.** Let (V, A, A', I) be a graph formal context, $X \subseteq V$ $(X \neq \emptyset)$, $B_1, B_2 \subseteq A$ $(B_1, B_2 \neq \emptyset)$. If $X^< = (B_1, B_2)$, X is connected and there is no connection between $x \in (B_1, B_2)^> - X$ and X, then the order pair $(X, (B_1, B_2))$ is called a local graph OE-concept. The set of all local graph OE-concepts in (V, A, A', I) is denoted as $N_{TL}(V, A, A', I)$.

Example 12. Table 9 is a graph formal context (V, A, A', I), where $U = \{x_1, x_2, \ldots, x_6\}$ represents 6 individuals, $A = \{a_1, a_2, \ldots, a_6\}$ represents 6 sports: badminton, tennis, volleyball, table tennis, running and rope skipping. In addition, $a_{ij} = 1$ represents there is communication between x_i and x_j, while $a_{ij} = 0$ represents there is no communication between x_i and x_j; $(x_i, m_j) = 1$ represents the object x_i likes the sport a_j, while $(x_i, a_j) = 0$ represents the object x_i does not like the sport a_j. Fig. 16 is the graphical representation of the graph formal context (V, A, A', I) in Table 9.

Global graph OE-concepts of Table 9 are as follows: $(x_1x_2x_3, (a_4, \emptyset))$, $(x_1x_2x_4, (a_3, a_1))$, $(x_1x_2x_5, (\emptyset, a_5))$, $(x_1x_2, (a_3a_4, a_1a_5))$, $(x_1x_4, (a_3, a_1a_2))$, $(x_1x_5, (\emptyset, a_5a_6))$, $(x_3, (a_1a_4a_5, a_2a_3a_6))$, $(x_1, (a_3a_4, a_1a_2a_5a_6))$, $(x_2, (a_2a_3a_4a_6, a_1a_5))$, $(x_5, (a_1a_2, a_3a_4a_5a_6))$, $(x_4x_6, (a_5, a_2a_4))$, $(x_4, (a_3a_5a_6, a_1a_2a_4))$, $(x_6, (a_1a_5, a_2a_3a_4a_6))$.

Local graph OE-concepts of Table 9 are as follows: $(x_1x_2x_3, (a_4, \emptyset))$, $(x_1x_2x_4, (a_3, a_1))$, $(x_1x_2x_5, (\emptyset, a_5))$, $(x_1x_2, (a_3a_4, a_1a_5))$, $(x_1x_4, (a_3, a_1a_2))$, $(x_1x_5, (\emptyset, a_5a_6))$, $(x_3, (a_1a_4a_5, a_2a_3a_6))$, $(x_1, (a_3a_4, a_1a_2a_5a_6))$, $(x_2, (a_2a_3a_4a_6, a_1a_5))$, $(x_5, (a_1a_2, a_3a_4a_5a_6))$, $(x_4x_6, (a_5, a_2a_4))$, $(x_4, (a_3a_5a_6, a_1a_2a_4))$, $(x_6, (a_1a_5, a_2a_3a_4a_6))$, $(x_3x_5, (a_1, a_3a_6))$, $(x_1x_4x_6, (\emptyset, a_2))$, $(x_1x_3x_5, (\emptyset, a_6))$.

4.4.3. Object and attribute network formal concepts in complex networks

The following describes network formal concepts represented by characteristic parameters [52].

Definition 30 (Node Centrality)**.**

$$c_D(i) = \sum_{j=1}^{N} (|x_j| + \sum_{k=1}^{L} |a_{ijk}|),$$

where $|x_j|$ represents the number of edges adjacent to node $|x_i|$, $\sum_{k=1}^{L}|a_{ijk}|$ Indicates the number of common attributes between nodes $|x_i|$ and $|x_j|$, that is, if $a_{ijk}=0$, it means that node $|x_i|$ and $|x_j|$ do not share attribute k, and if $a_{ijk}=1$, it means that node $|x_i|$ and $|x_j|$ share attribute k. N is the number of nodes in network, and L represents the total number of attributes in attribute set A in the network formal context.

Definition 31 (Relative Centrality of Nodes)**.**

$$c_D'(i)=\frac{c_D(i)}{(N-1)(L+1)},$$

where is the number of nodes in network, and L represents the total number of attributes in network, so the maximum degree of this node is $N-1+(N-1)L$, that is, $(N-1)(L+1)$.

Definition 32 (Central Potential)**.**

$$c_D'=\frac{\sum_{i=1}^{N}[c_{D_{max}}-c_D(i)]}{(N-1)(L+1)},$$

where $c_{D_{\max}}$ is the largest value of the centrality of all nodes.

Definition 33 (Object and Attribute Network Formal Concepts)**.** Let (O,A,A',I) be a graph formal context, then the concept $(\mathscr{M},X,X^{\uparrow})$ is called object network formal concepts, and the concept $(m,B^{\downarrow},B)$ is called attribute network formal concepts.

Here $\mathscr{M}$ denotes the characteristic parameters of network, $m=\{\mathscr{M}_1,\mathscr{M}_2\}$, $\mathscr{M}_1=\frac{\sum_{i=1}^{N}([c_{D_max}-c_D(i)])}{(N-1)(L+1)}$ denotes potential of concept, which indicates the degree of difference within a network concept, and $\mathscr{M}_2=\frac{\sum_{i=1}^{N}c_D(i)}{(N-1)(L+1)}$ denotes concept average degree, which indicates the relative importance of network concepts in the network.

Definition 34 (Strong/Weak Concepts of Objects and Attributes)**.** Let (V,A,A',I) be a graph formal context, and $X\subseteq V$, $B\subseteq A$,

(1) if $X^{\uparrow\downarrow}=X$, then $(\mathscr{M},X,X^{\uparrow})$ is called an object strong concept;
(2) if $X^{\uparrow\downarrow}\supset X$, then let $\beta=\frac{|X|}{|X^{\uparrow\downarrow}|}$. $(\mathscr{M},X,X^{\uparrow})_\beta$ is called an object weak concept when $0\le\beta<1$, while $(\mathscr{M},X,X^{\uparrow})_\beta$ is called an object weak concept under the degree of β^* when $\beta\ge\beta^*$;
(3) if $B^{\downarrow\uparrow}=B$, then $(m,B^{\downarrow},B)$ is called an attribute strong concept;
(4) if if $B^{\downarrow\uparrow}\supset B$, then let $\gamma=\frac{|B|}{|B^{\downarrow\uparrow}|}$, and $(m,B^{\downarrow},B)_\gamma$ is called an attribute weak concept when $0\le\gamma<1$, while $(m,B^{\downarrow},B)_\gamma$ is called an attribute weak concept under the degree of γ^* when $\gamma\ge\gamma^*$.

Here $|X^{\uparrow\downarrow}|$ denotes the number of objects corresponding to attribute $X^{\uparrow}$ shared by set X. β denotes the proportion of $|X|$ to $|X^{\uparrow\downarrow}|$, and if β is large and approaches to 1, it means that $(X,X^{\uparrow})$ is a highly representative concept, while β is small and approaches to 0, it means that $(X,X^{\uparrow})$ is not highly representative in the whole network. γ denotes the proportion of $|B|$ to $|B^{\downarrow\uparrow}|$, and if γ is large and approaches to 1, it means that $(B^{\downarrow},B)$ is a highly representative concept, while γ is small and approaches to 0, it means that $(B^{\downarrow},B)$ is not highly representative in the whole network.

4.5. Uncertain graph classification

As the feature information of the graph is extracted through graph convolution and graph pooling, it needs to be input it into a Multi-Layer Perceptron (MLP) for graph classification. To effectively utilize the feature information of the graph, Fan et al. [51] proposed employing the Fuzzy Entropy Causality Classification Method (FECCM) to select crucial graph features for the classification task, ultimately improving performance.

The essence of Pearl's Causality is to characterize the effectiveness of the rule through the rule's antecedent c_i, consequent d_j and their corresponding confidence level μ_{ij}. In graph classification, if a graph with a specific attribute is labeled as c_i, and the classification decision is marked as d_j , then the classification rule can be represented as $r(c_i,d_j)$: $c_i\xrightarrow{\mu_{ij}}d_j$. Let μ_{ij} (confidence) be the probability value of the ith graph with attribute c being assigned to the jth.

Given that the confidence level μ_{ij} of a classification rule reflects the degree to which the rule is considered accurate or reliable, it should be modeled as a fuzzy number. This approach enables a more effective representation of the uncertainty and fuzziness inherent in classification rules. Accordingly, this paper investigates classification rules and their confidence levels by applying definitions based on fuzzy sets, fuzzy entropy, and fuzzy cut sets. Utilizing these concepts, the research aims to construct a framework capable of managing the inherent vagueness and imprecision in classification rules.

Definition 35 ([76] Causality)**.** For rule $r(c_i,d_j)$: $c_i\xrightarrow{\mu_{ij}}d_j$ and its anti-rule $r(d_j,c_i)$: $dj\xrightarrow{\mu_{ji}}c_i$, the causality between c_i and d_j is defined as $CP=\mu_{ij}+\mu_{ji}-1(CP\in[-1,1])$.

Definition 36 (Fuzzy Entropy Causality)**.** For rule $r(c_i,d_j)$: $c_i\xrightarrow{\mu_{ij}}d_j$ and its anti-rule $r(d_j,c_i)$: $d_j\xrightarrow{\mu_{ji}}c_i$, the fuzzy entropy causality between c_i and d_j is defined as follows:

$$CP_E(r_{ij})=1-\frac{1}{log_2^K}(-\mu_{ij}log_2\mu_{ij}-\mu_{ji}log_2\mu_{ji}).$$

where K is the total number of classification labels for discrete graphs. If $K=2$, that is, the graph is a two-category graph and the graph has only two labels, then $CP_E(r_{ij})=1+\mu_{ij}log_2\mu_{ij}+\mu_{ji}log_2\mu_{ji})$.

Example 13. For binary classification, let's denote the classification rules for the graph as: r_{ij_1} : $\mu_{ij_1}=1.0,\mu_{j_1i}=0.2$ and r_{ij_2} : $\mu_{ij_2}=0.6,\mu_{j_2i}=0.6$. From Definition 35, we know that the causality of rules r_{ij_1} and r_{ij_2} is the same, that is $CP(r_{ij_1})=CP(r_{ij_2})$:

$$CP(r_{ij_1})=\mu_{ij_1}+\mu_{j_1i}-1=1.0+0.2-1=0.2,$$

$$CP(r_{ij_2})=\mu_{ij_2}+\mu_{j_2i}-1=0.6+0.6-1=0.2.$$

It can be seen from this $CP(r_{ij_1})=CP(r_{ij_2})$.

However, from Definition 36, we can know that the fuzzy entropy causality of rules r_{ij_1} and r_{ij_2} is different:

$$CP_E(r_{ij_1})=1-\frac{1}{log_2K}(-1\times-log_2(1)-0.2\times log_2(2))$$
$$\approx 1-0.464\approx 0.536,$$

$$CP_E(r_{ij_2})=1-\frac{1}{log_2K}(-0.6\times-log_2(0.6)-0.6\times log_2(0.6))$$
$$\approx 1-0.884\approx 0.116.$$

That is, when using rule r_{ij_1} to classify object c_i, the fuzzy entropy causality is greater, the greater the certainty, so it is classified into the j_1 category.

Theorem 4. *Let any r_{ij_l}, $r_{ij_q}\in R$, if $CP(r_ij_l)=CP$, then $CP_E(r_{ij_l})\le CP_E(r_{ij_q})$ or $CP_E(r_{ij_l})\ge CP_E(r_{ij_q})$ holds true.*

Theorem 5. *Let any r_{ij_l}, $r_{ij_q}\in R$, if $CP(r_ij_l)=CP$, and $\mu_{ij_l}=\mu_{ij_q}$ or $\mu_{j_li}=\mu_{j_qi}$, then $CP_E(r_{ij_l})=CP_E(r_{ij_q})$ holds.*

Theorem 6. *Let any r_{ij_l}, $r_{ij_q}\in R$, if $CP_E(r_ij_l)=CP_E(r_ij_q)$, then $CP(r_{ij_l})=CP(r_{ij_q})$ holds.*

From Theorems 5 and 6, we can see that the fuzzy entropy causality characterizes the classification rules more accurately.

Based on the aforementioned definitions and theorems, a fuzzy entropy causality classification method (FECCM) is introduced. This approach is designed to leverage the certainty of fuzzy entropy causality evaluation rules for selecting critical feature information essential for classification tasks. Finally, the integration of HAGP and FECCM is employed for hierarchical graph classification; a comprehensive description of the graph classification implementation is provided in the subsequent Fig. 17. The process can be divided into six steps: (1) The original

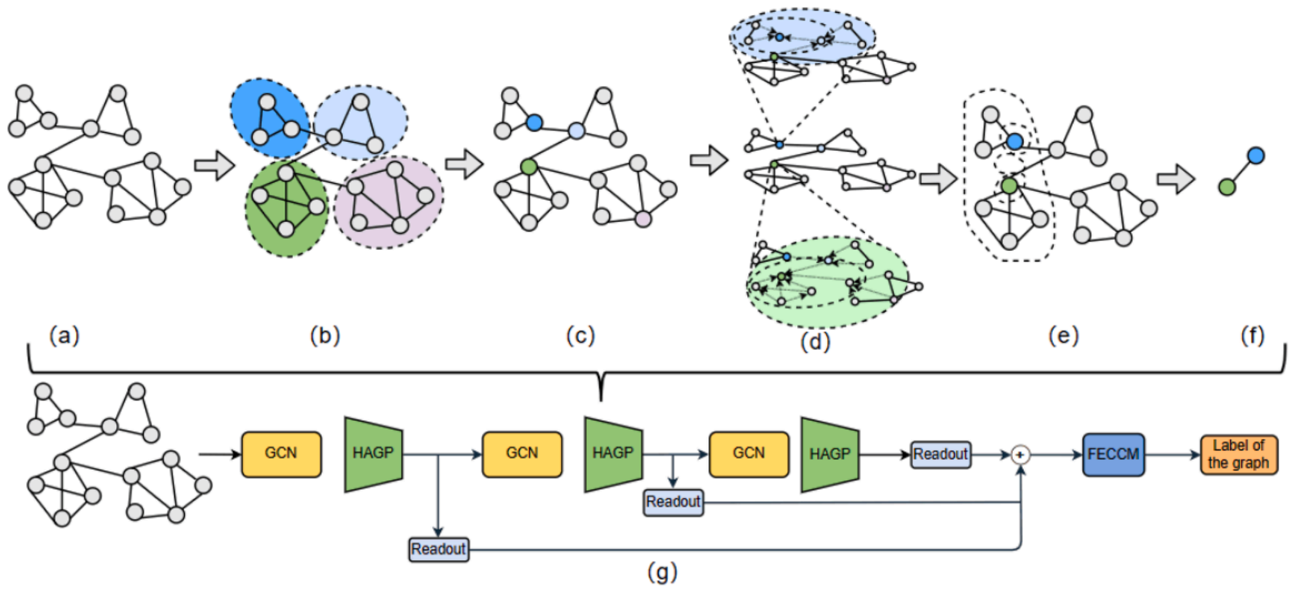


Fig. 17. Overview of graph classification model structure.

graph is input into the HAGP model. (2) Clusters of node neighborhoods are constructed. The Master2Token method is used to learn the representation of cluster attributes and obtain the corresponding assignment matrix. (3) Clusters are evaluated using LEConv. The intensity of a cluster's color in the graph corresponds to the magnitude of its score. (4) Neighborhood cluster information is aggregated. Two prominently colored clusters in the graph are selected, and their 2-hop neighborhood cluster information is aggregated using HO-CIA. Dashed arrows in the graph indicate the flow of first and second-order feature information within the clusters. (5) The top-k method is applied to identify clusters with higher scores. The adjacency matrix recalculates edge weights by incorporating neighboring nodes from the selected clusters. (6) A coarse-grained graph is output.

4.6. Knowledge graph mining

4.6.1. Link prediction

The Relational Graph Convolutional Network (R-GCN) [77] marks a notable advancement in graph neural networks, tailored specifically for multi-relational and heterogeneous graph-structured data. By iteratively aggregating features from neighboring nodes via relation-specific transformations, this model refines entity representations and effectively captures both direct and higher-order relational dependencies in the graph. In contrast to standard GCNs that treat all edges uniformly, the R-GCN uses separate weight matrices for each relation type, enabling it to distinguish and incorporate diverse relational semantics during information propagation. To prevent overfitting and control parameter growth as the number of relations increases, the model employs regularization methods such as basis or block-diagonal decomposition, facilitating weight sharing across relations and ensuring scalability. This architecture not only improves the expressiveness of node embeddings but also supports tasks such as entity classification and link prediction in complex knowledge graphs.

Dai et al. [78] integrated the granular concept into R-GCN model and constructed GCR-GCN model to improve the performance of link prediction model. The GCR-GCN model can not only enhance the interpretability, but also improve the performance of the model.

As illustrated in Fig. 18, the GCR-GCN model comprises two distinct modules. The first module is dedicated to computing relational weights based on granular concepts. This process involves four key steps: initially, a formal context is constructed utilizing entities and their adjacent relations; subsequently, granular concepts derived from objects (entities) are calculated within this formal context; then, a subset of more analogous objects (entities) is filtered by evaluating the similarity of attributes (relations) between concepts; ultimately, the information from these similar entities is fused into the updated entities to determine the importance of each relation. The second module simultaneously integrates information from adjacent relations and entities into the entity being updated. In this phase, the relational weights calculated by the first module allow different updated entities to effectively incorporate distinct information from the same relation.

4.6.2. Entity summarization

Entities in knowledge graphs are described using the Resource Description Framework (RDF), which represents various resources and their relationships in the form of "subject-predicate-object" triples. However, knowledge graphs often contain a vast amount of entity description information, leading to an information overload problem when users retrieve entity information.

For instance, the latest English version of the knowledge graph DBpedia contains 6.6 million entities and 1.7 billion RDF triples, with an average of 258 descriptive pieces of information per entity [79]. Given such an overwhelming volume of entity information, it becomes challenging for users to find useful information, which can result in information overload. Therefore, it is crucial to utilize entity summarization techniques to streamline knowledge graphs and better serve users.

Entity Summarization is to select representative Top-k triples from the vast and verbose triples in knowledge graphs, thereby streamlining entity information effectively. Given that the predicates and objects in RDF data for an entity can be transformed into a binary tabular format, it is reasonable to consider applying Formal Concept Analysis (FCA) to entity summarization. Kim et al. [79] introduced KAFCA for this purpose, which generates ranked RDF triples based on the weights of the extents of concepts within a concept lattice. However, their approach neglects the nature of dynamics of knowledge graph which limits the applicability of the above approach. To this end, our previous work proposes an incremental approach to enhance the efficiency of entity summarization using Formal Concept Analysis (FCA)[80]. Additionally, we refined the ranking algorithm by incorporating factors such as the importance, redundancy, and uniqueness of triples to achieve improved summarization outcomes.

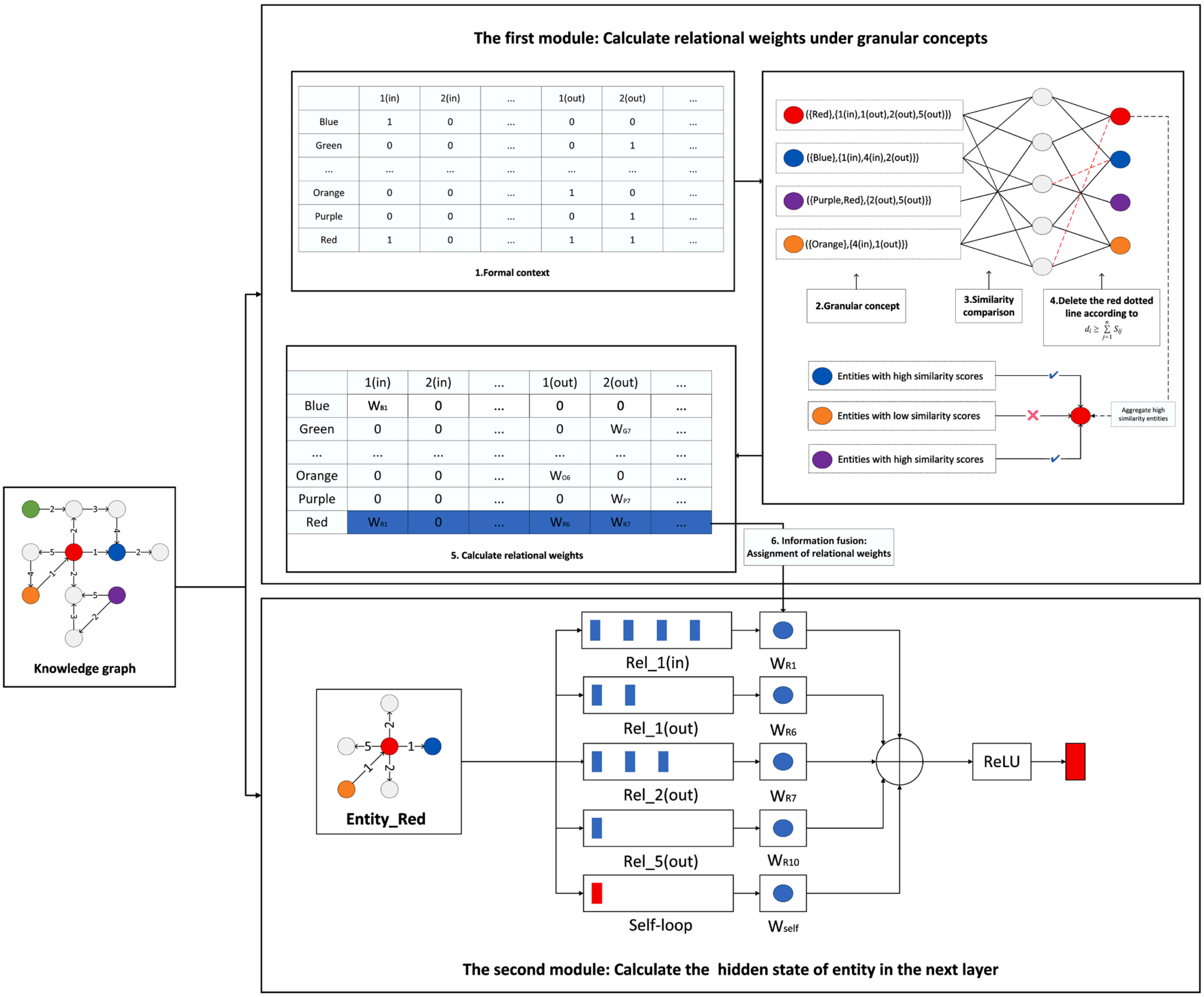


Fig. 18. Overall framework of GCR-GCN.

4.7. CCL-empowered graph learning

Recently, there are emerging cutting-edge research areas about CCL-empowered graph learning. That is to say, CCL is used for assisting the tasks of graph learning.

Current graph neural networks (GNNs) map nodes in networks to specific samples in datasets, thereby overlooking the conceptual information embedded in object-attribute clusters within the data. Furthermore, GNNs face challenges when handling data lacking structural information, as such information serves as their core input. This paper seeks to incorporate conceptual information into the message-passing mechanism of GNNs. To achieve this, Shao et al. [81] integrated concept lattice theory with existing GNN architectures. Technically, a concept lattice is a robust tool for characterizing the generalization and specialization relationships between formal concepts. As the fundamental components of concept lattices, formal concepts effectively illuminate the dependencies between features and samples. Building on this, they presented a novel GNN framework derived from concept lattices, designed to address the inherent limitations of conventional GNNs. This innovative framework not only incorporates conceptual information into message passing but also empowers GNNs to process both structured and unstructured data.

Regarding the traffic flow prediction, our previous work [82,83] proposed a novel Formal Concept-enhanced Graph Convolutional Network (FC-GCN) model. This integration enhances the training accuracy of the models by leveraging FCA to distill structural features from traffic data and subsequently refining the feature representation of the traffic data. Further, under federated learning (FL), the GCNs model can be trained independently on different clients, and the local model is optimized by sharing model parameters. Consequently, a novel formal concept-enhanced federated graph learning paradigm is developed [83]. Coupled with the premise of protecting data privacy, the integrity of the data is guaranteed and the training accuracy of the GCNs model is improved.

Another important task of graph learning is node classification. Yan et al. [47] proposed a graph network semi-supervised concept-cognitive learning model for this task. It contains three main parts: the construction of concept space, the pseudo-labeled process, and concept generalization. This model not only takes a new perspective for node classification, but also has natural interpretability and shares certain advantages in classification accuracy.

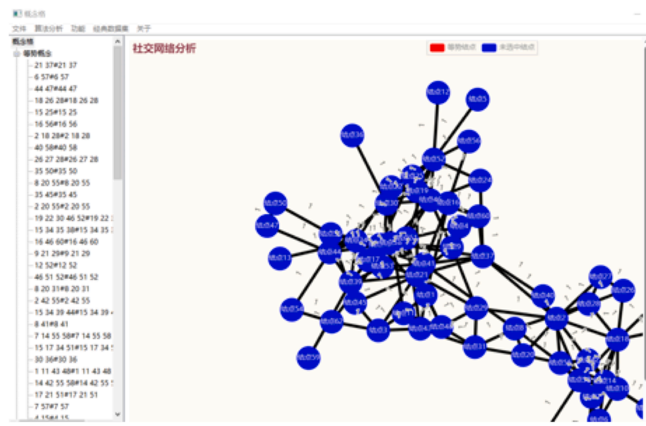
5. Datasets and platforms

There are several datasets and benchmarks used to evaluate the performance of graph concept-cognitive learning various down-streaming tasks such as link prediction, node classification, and community detec-

Table 10
Graph datasets for node, edge, and graph-level tasks.

Dataset Type	Dataset Name	Task	Nodes	Edges	Classes	Features
Node-level	Cora	Node Classification (paper topic)	2708	5429	7	1433
Node-level	CiteSeer	Node Classification	3327	4732	6	3703
Node-level	PubMed	Node Classification	19,717	44,338	3	500
Node-level	PPI	Node Classification (proteins)	56,944	818,716	121	50
Node-level	Reddit	Node Classification (subreddits)	232,965	11,606,919	41	602
Node-level	ogbn-arxiv	Node Classification (arXiv)	169,343	1,166,243	40	128
Node-level	Flickr	Node Classification	89,250	899,756	7	500
Node-level	DHFR	Node Classification	31,384	67,352	4	37
Node-level	Cuneiform	Node Classification	5677	23,922	4	3
Node-level	BZR	Node Classification	14,479	31,070	10	3
Edge-level	Facebook Page-Page	Link Prediction	22,470	170,912	N/A	None
Edge-level	PPI (Link Prediction)	Link Prediction (proteins)	Varies	Varies	N/A	N/A
Edge-level	CollabNet	Link Prediction	18,772	44,327	N/A	None
Edge-level	Amazon Co-purchase	Link Prediction (products)	334,863	925,872	N/A	None
Edge-level	YouTube Friendship	Link Prediction	1,134,890	2,987,624	N/A	None
Edge-level	ogbl-collab	Link Prediction (collaborations)	235,868	1,285,465	N/A	None
Graph-level	MUTAG	Graph Classification (toxicology)	188	Varies	2	Varies
Graph-level	PROTEINS	Graph Classification (functions)	1113	Varies	2	Varies
Graph-level	ENZYMES	Graph Classification (enzymes)	600	Varies	6	Varies

(a) Formal Context based Representation of Graph



(b) Visualization of Graph

Fig. 19. The interfaces of FCA4Graph.

tion [2] [3]. For example, Table 10 shows the commonly-used datasets in terms of node-level, edge-level and graph-level.

Recently, some GCCL libraries or platforms are developed for implementing simulation and basic graph-related tasks execution. For example, FCA4SNS [4] is a Python library for social network analysis based on concept-cognitive learning [69]. It aims to address the representation of social network, network topology mining, and social intelligence extraction. To extend the application range, we developed a novel GCCL simulation tool, named FCA4Graph, which is used for supporting data mining and analysis in signed network, attributed network, knowledge graph, dynamic graph, as well as fuzzy graph. Fig. 19 demonstrates the main interfaces of FCA4Graph, the users can manually create the formal context of the given graph (Fig. 19(a)), then the graph cognition results are displayed in Fig. 19(b).

In addition, a cloud-based graph concept-cognitive learning simulation platform, termed GCCL[5], is developed, as shown in Fig. 20. It aims to facilitate efficient graph structure mining, understanding, and application by exploring the mapping relationships between subgraph structures and formal concepts. This simulation platform emphasizes leveraging the intuitiveness and structural characteristics of graphs to decompose complex formal concepts into more comprehensible units, presenting their interactive relationships in a graphical manner.

6. Open issues and future directions

GCCL is an emerging research topic. Although significant progresses have been made for GCCL, there still remain plenty of research directions worthy of future explorations.

6.1. *Theoretical guarantees*

While numerous GCCL methods have been put forward and shown empirical effectiveness, there remains a need for deeper exploration into more fundamental theoretical analyses of graph CL. A particularly promising avenue is to develop such theoretical frameworks by drawing inspiration from general GCCL approaching the problem from the perspectives of optimization or data distribution. This would facilitate a clearer understanding of the underlying mechanisms and effectiveness of GCCL methods.

[2] https://www.start.umd.edu/data-tools/GTD
[3] https://chrsmrrs.github.io/datasets/docs/datasets/
[4] https://github.com/jiegao19/FCA4SNA
[5] https://fhaocs.github.io/GCCL

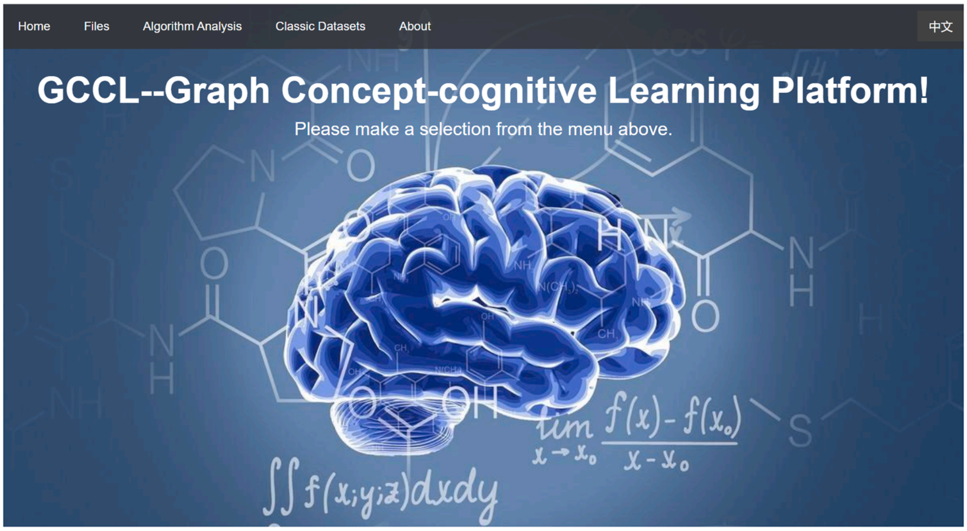


Fig. 20. The interface of GCCL.

6.2. GCCL for directed graph

The above literature review has verified the GCCL has a promising potential to learning various down-streaming graph tasks. However, the research work about GCCL on directed graphs has not yet been studied. The difficulty of applying GCCL to directed graph modeling primarily stems from the distinction between binary attributes and directed relations, as well as how to address issues of complexity and directionality. To overcome this limitation, a cognitive operator based on the dual “in-degree–out-degree” neighborhood of directed edges: the out-degree neighborhood characterizes a node’s “active influence” attributes, while the in-degree neighborhood represents its “passive association” attributes. By integrating the node’s intrinsic semantic information with directed graph topology information, the corresponding directed graph formal context can be constructed for further concept cognition of graph structures.

6.3. Scalability and efficiency optimization of GCCL for large-scale graphs

This open issue addresses the insufficient exploration of scalability and efficiency of GCCL for large-scale (million-node) graphs. Rooted in theoretical basis of GCCL—FCA—the core bottleneck is concept lattice explosion (exponential concept growth with graph scale), leading to prohibitive computational costs (e.g., excessive memory, long runtime) and limiting practicality of GCCL in real-world applications. Therefore, the following key research directions can be further investigated. First, reviewing and integrating existing FCA acceleration techniques, such as distributed computing [84], incremental updates [85], approximation algorithms [86] into GCCL, is under-discussed. Second, it is crucial to verify the feasibility and develop the lightweight/divide-and-conquer strategies for GCCL on large-scale graphs with local subgraph sampling and local concept extraction, concept pruning, and hierarchical concept lattice construction.

6.4. GCCL for foundation models

The integration of GCCL and foundation models (FMs) merges structured relational reasoning with large-scale generalizable learning, addressing their complementary limitations: FMs excel at processing unstructured data and generalization but lack explicit relational structure and interpretability [87], while GCCL offers structured concept-relation modeling but struggles with static, manually curated graphs and unstructured data. Key integration approaches include using FMs to enrich graphs (e.g., extracting concepts/relations from text, contextualizing nodes), leveraging graphs to guide FMs (e.g., knowledge-augmented training, enhancing interpretability), and enabling co-evolution (e.g., feedback loops for dynamic adaptation). This synergy enables applications like semantic search, cognitive robotics, and personalized education, through challenges remain in aligning representations, ensuring scalability, adapting to dynamic knowledge, and achieving cognitive plausibility. Future directions include multi-modal graphs, neural-symbolic hybrids, human-in-the-loop systems, and efficient training methods, aiming to create more robust, interpretable, and human-aligned AI.

6.5. Broader applications

While GCCL methods have been applied to several tasks as discussed in this work, their potential in more diverse real-world applications—such as traffic flow prediction [83], QA answering system [80], edge computing tasks allocation [88], and AI inference [52]—warrants further exploration to achieve more effective and satisfying predictive outcomes. A key challenge lies in how to incorporate appropriate domain knowledge as additional priors to guide model design.

7. Conclusion

This paper introduces an emerging paradigm, Graph Concept-cognitive Learning (GCCL), which incorporates concept-cognitive learning into graph analysis. A comprehensive framework for GCCL is proposed, detailing techniques for integrating concept-cognitive learning with various graph representations. These representations enable a range of graph cognition tasks, including graph structure analysis, knowledge discovery, and graph classification. To support further research in this area, relevant datasets and simulation platforms are also introduced. This study provides foundational insights into GCCL, facilitating its application and exploration in advanced graph learning scenarios.

CRediT authorship contribution statement

Fei Hao: Writing – original draft, Supervision, Methodology, Investigation, Formal analysis, Conceptualization; **Mengyu Yan:** Writing – original draft, Methodology, Investigation, Formal analysis; **Jinhai Li:** Writing – review & editing, Validation, Resources, Formal analysis; **Weihua Xu:** Writing – review & editing, Validation, Data curation.

Data availability and access

The authors declare that all datasets can be obtained from https://networkrepository.com/index.php.

Data availability

Data will be made available on request.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was funded in part by the National Natural Science Foundation of China (Grant Nos. 62477029, 61702317), the Ministry of Education Humanities and Social Sciences Research Youth Fund Project (Grant No. 22YJCZH046), the Fundamental Research Funds for the Central Universities (Grant No. GK202505028), and Yunnan Fundamental Research Projects (Grant Nos. 202401AV070009, 202501AU070156).

References

[1] S.S. Singh, S. Muhuri, S. Mishra, D. Srivastava, H.K. Shakya, N. Kumar, Social network analysis: a survey on process, tools, and application, ACM Comput. Surv. 56 (8) (2024) 1–39.
[2] C. Li, L. Xia, X. Ren, Y. Ye, Y. Xu, C. Huang, Graph transformer for recommendation, in: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2023, pp. 1680–1689.
[3] S. Motie, B. Raahemi, Financial fraud detection using graph neural networks: a systematic review, Expert Syst. Appl. (2023) 122156.
[4] M. Réau, N. Renaud, L.C. Xue, A.M. Bonvin, DeepRank-GNN: a graph neural network framework to learn patterns in protein–protein interfaces, Bioinformatics 39 (1) (2023) btac759.
[5] J. Li, R. Zheng, H. Feng, M. Li, X. Zhuang, Permutation equivariant graph framelets for heterophilous graph learning, IEEE Trans. Neural Netw. Learn. Syst. 35 (9) (2024) 11634–11648.
[6] Z. Li, G. Zhang, J. Yu, L. Xu, Dynamic graph structure learning for multivariate time series forecasting, Pattern Recognit. 138 (2023) 109423.
[7] L. Pasa, N. Navarin, W. Erb, A. Sperduti, Empowering simple graph convolutional networks, IEEE Trans. Neural Netw. Learn. Sys. 35 (11) (2024) 4385–4399.
[8] K. Sun, Z. Huang, H. Mao, A. Qin, X. Li, W. Tang, J. Xiong, Multi-scale cluster-graph convolution network with multi-channel residual network for intelligent fault diagnosis, IEEE Trans. Instrum. Meas. 71 (2022) 2502612.
[9] W. Ju, Y. Wang, Q. Yifang, Z. Mao, X. Zhiping, J. Luo, J. Yang, Y. Gu, D. Wang, Q. Long, S. Yi, X. Luo, M. Zhang, Towards graph contrastive learning: a survey and beyond, (2024). https://doi.org/10.48550/arXiv.2405.11868
[10] Y. Liu, Z. Li, S. Pan, C. Gong, C. Zhou, G. Karypis, Anomaly detection on attributed networks via contrastive self-supervised learning, IEEE Trans. Neural Netw. Learn. Syst. 33 (6) (2022) 2378–2392.
[11] L. Yu, L. Sun, B. Du, T. Zhu, W. Lv, Label-enhanced graph neural network for semi-supervised node classification, IEEE Trans. Knowl. Data Eng. 35 (11) (2023) 11529–11540.
[12] M. Gong, H. Zhou, A.K. Qin, W. Liu, Z. Zhao, Self-paced co-training of graph neural networks for semi-supervised node classification, IEEE Trans. Neural Netw. Learn. Syst. 34 (11) (2023) 9234–9247.
[13] J. Jin, D. Rong, T. Zhang, Q. Ji, H. Guo, Y. Lv, X. Ma, F. Wang, A GAN-based short-term link traffic prediction approach for urban road networks under a parallel learning framework, IEEE Trans. Intell. Transp. Syst. 23 (9) (2022) 16185–16196.
[14] Z. Fang, S. Tan, Y. Wang, J. Lv, Elementary subgraph features for link prediction with neural networks, IEEE Trans. Knowl. Data Eng. 35 (4) (2023) 3822–3831.
[15] Y. Xie, S. Lv, Y. Qian, C. Wen, J. Liang, Active and semi-supervised graph neural networks for graph classification, IEEE Trans. Big Data 8 (4) (2022) 920–932.
[16] J. Ji, H. Jia, Y. Ren, M. Lei, Supervised contrastive learning with structure inference for graph classification, IEEE Trans. Network Sci. Eng. 10 (3) (2023) 1684–1695.
[17] L. Wu, P. Cui, J. Pei, L. Zhao, X. Guo, Graph neural networks: foundation, frontiers and applications, in: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2022, pp. 4840–4841.
[18] W. Zhang, W. Xu, Cognitive model based on granular computing, Chinese J. Eng. Math. 24 (6) (2007) 957–971.
[19] W. Xu, D. Guo, J. Mi, Y. Qian, K. Zheng, W. Ding, Two-way concept-cognitive learning via concept movement viewpoint, IEEE Trans. Neural Netw. Learn. Syst. 34 (10) (2023) 6798–6812.
[20] D. Guo, W. Xu, Y. Qian, W. Ding, M-FCCL: memory-based concept-cognitive learning for dynamic fuzzy data classification and knowledge fusion, Inf. Fusion 100 (2023) 101962.
[21] D. Guo, W. Xu, W. Ding, Y. Yao, X. Wang, W. Pedrycz, Y. Qian, Concept-cognitive learning survey: mining and fusing knowledge from data, Inf. Fusion 109 (2024) 102426.
[22] R. Wille, Restructuring lattice theory: an approach based on hierarchies of concepts, in: RIVAL I. Ordered Sets., 1982, pp. 445–470.
[23] Z. Pawlak, Rough sets, Int. J. Comput. Inf. Sci. 11 (1982) 341–356.
[24] L.A. Zadeh, Fuzzy sets and information granularity, in: Advances in Fuzzy Set Theory and Applications, 1979, pp. 3–18.
[25] X. Li, Multi-modal cognitive computing, Sci. Sinica Inf. 53 (1) (2023) 1–32.
[26] Y. Wang, On concept algebra: a denotational mathematics structure for knowledge and software modeling, Int. J. Cognitive Inf. Natural Intell. 2 (2) (2008) 1–19.
[27] J. Ma, W. Zhang, Axiomatic characterizations of dual concept lattices, Int. J. Approximate Reasoning 54 (5) (2013) 690–697.
[28] Y. Yao, Interpreting concept learning in cognitive informatics and granular computing, IEEE Trans. Sys. Man, Cybern.-Part B: Cybern. 39 (4) (2009) 855–866.
[29] J. Li, M. Changlin, W. Xu, Y. Qian, Concept learning via granular computing: a cognitive viewpoint, Inf. Sci. 298 (2015) 447–467.
[30] C. Kumar, I. Ms, C. Loo, Formal concept analysis approach to cognitive functionalities of bidirectional associative memory, Biol. Inspired Cognit. Archit. 12 (2015) 20–33.
[31] Y. Shi, Y. Mi, J. Li, W. Liu, Concept-cognitive learning model for incremental concept learning, IEEE Trans. Sys. Man, Cybern. Syst. 51 (2) (2021) 809–821.
[32] Y. Mi, Y. Shi, J. Li, W. Liu, M. Yan, Fuzzy-based concept learning method: exploiting data with fuzzy conceptual clustering, IEEE Trans. Cybern. 52 (1) (2022a) 582–593.
[33] Y. Mi, W. Liu, Y. Shi, J. Li, Semi-supervised concept learning by concept-cognitive learning and concept space, IEEE Trans. Knowl. Data Eng. 34 (5) (2022b) 2429–2442.
[34] J. Niu, D. Chen, J. Li, H. Wang, Fuzzy rule based classification method for incremental rule learning, IEEE Trans. Fuzzy Syst. 30 (9) (2022) 3748–3761.
[35] X. Zhang, D. Chen, J. Mi, Fuzzy decision rule-based online classification algorithm in fuzzy formal decision contexts, IEEE Trans. Fuzzy Syst. 31 (9) (2023) 3263–3277.
[36] Z. Liu, J. Li, X. Zhang, X. Wang, Incremental incomplete concept-cognitive learning model: a stochastic strategy, IEEE Trans. Neural Netw. Learn. Syst. Early Access (2023) https://doi.org/10.1109/TNNLS.2023.3333537.
[37] X. Deng, J. Li, Y. Qian, J. Liu, An emerging incremental fuzzy concept-ccognitive learning model based on granular computing and conceptual knowledge clustering, IEEE Trans. Emerg. Top. Comput. Intell. 8 (3) (2024) 2417–2432.
[38] F. Hao, G. Min, Z. Pei, D.-S. Park, L.T. Yang, K-clique community detection in social networks based on formal concept analysis, IEEE Syst. J. 11 (1) (2017) 250–259.
[39] F. Hao, D.-S. Park, G. Min, Y.-S. Jeong, J.-H. Park, K-cliques mining in dynamic social networks based on triadic formal concept analysis, Neurocomputing 209 (2016) 57–66.
[40] F. Hao, Z. Pei, L.T. Yang, Diversified top-k maximal clique detection in social internet of things, Future Gener. Comput. Syst. 107 (2020) 408–417.
[41] F. Hao, Y. Yang, G. Min, V. Loia, Incremental construction of three-way concept lattice for knowledge discovery in social networks, Inf. Sci. 578 (2021) 257–280.
[42] D. Guo, W. Xu, Y. Qian, W. Ding, Fuzzy-granular concept-cognitive learning via three-way decision: performance evaluation on dynamic knowledge discovery, IEEE Trans. Fuzzy Syst. 32 (3) (2023) 1409–1423.
[43] N. Xie, E. Yan, X. Yao, Q. Chen, T. Hao, A concept-cognitive learning approach from the perspective of causal reasoning, Neurocomputing (2025) 131876.
[44] L. Zhang, E. Yan, T. Hao, Fuzzy concept-cognitive learning based on attention mechanism and three-way partial order structure, Appl. Soft. Comput. (2026) 114664.
[45] Y. Lou, C. Jia, F. Hao, L. Guo, J. Li, Cognitive memory enhancement based on concept reduction, Int. J. Mach. Learn. Cybern. 17 (2) (2026) 42.
[46] M. Tao, F. Hao, L. Wei, H. Zhi, S.O. Kuznetsov, G. Min, Fairness-aware maximal cliques identification in attributed social networks with concept-cognitive learning, IEEE Trans. Comput. Social Syst. 11 (6) (2024) 7373–7385.
[47] M. Yan, J. Li, X. Wang, GNS2CCL: a graph network semi-supervised concept-cognitive learning model for node classification, Pattern Recognit. 174 (2026) 112958.
[48] T. Zhang, M. Rong, H. Shan, M. Liu, Causal asymmetry analysis in the view of concept-cognitive learning by incremental concept tree, Cognit. Comput. 13 (5) (2021) 1274–1286.
[49] M. Yan, J. Li, Knowledge discovery and updating under the evolution of network formal contexts based on three-way decision, Inf. Sci. 601 (2022) 18–38.
[50] F. Hao, Y. Sun, Y. Lin, Rough maximal cliques enumeration in incomplete graphs based on partially-known concept learning, Neurocomputing 496 (2022) 96–106.
[51] M. Fan, H. Li, Graph classification method based on fuzzy entropy causality in a network formal context, IEEE Trans. Fuzzy Syst. 32 (9) (2024) 5018–5032.
[52] N. Ma, M. Fan, J. Li, Concept-cognitive learning under complex network, J. Nanjing Univ. (Natural Sciences) 55 (4) (2019) 609–623.
[53] Y. Zhao, J. Li, W. Liu, W. Xu, Cognitive concept learning from incomplete information, Int. J. Mach. Learn. Cybern. 8 (2017) 159–170.
[54] Y. Wang, G. Baciu, Y. Yao, W. Kinsner, K. Chan, B. Zhang, S. Hameroff, N. Zhong, C.-R. Hunag, B. Goertzel, et al., Perspectives on cognitive informatics and cognitive computing, Int. J. Cognit. Inf. Natural Intell. (IJCINI) 4 (1) (2010) 1–29.
[55] I. Higgins, N. Sonnerat, L. Matthey, A. Pal, C.P. Burgess, M. Botvinick, D. Hassabis, A. Lerchner, Scan: learning abstract hierarchical compositional visual concepts, arXiv:1707.03389 (2017).
[56] J. Liu, J. Mi, D. Niu, A novel method for generating a canonical basis for decision implications based on object-induced three-way operators, Knowl. Based Syst. 283 (2024) 111161.

[57] E. Yan, S. Hao, T. Zhang, T. Hao, Q. Chen, J. Yu, Graph representation learning method based on three-way partial order structure, Int. J. Approximate Reasoning 165 (2024) 109104.
[58] F. Hao, J. Gao, C. Bisogni, G. Min, V. Loia, C. De Maio, Stability of three-way concepts and its application to natural language generation, Pattern Recognit. Lett. 149 (2021) 51–58.
[59] W. Xu, W. Li, Granular computing approach to two-way learning based on formal concept analysis in fuzzy datasets, IEEE Trans. Cybern. 46 (2) (2014) 366–379.
[60] P. Cordero, M. Enciso, A. Mora, M. Ojeda-Aciego, C. Rossi, Knowledge discovery in social networks by using a logic-based treatment of implications, Knowl. Based Syst. 87 (2015) 16–25.
[61] Aufaure, M.-A.s. Grand, B. Le, Advances in FCA-based applications for social networks analysis, Int. J. Conceptual Struct. Smart Appl. 1 (2013) 73–89.
[62] J. Poelmans, D.I. Ignatov, S.O. Kuznetsov, G. Dedene, Formal concept analysis in knowledge processing: a survey on applications, Expert Syst. Appl. 40 (2013) 6538–6560.
[63] T. Zhang, Z. Xue, H. Lu, S. Li, J. Zhang, X. Liu, Incremental equiconcept calculation based on graph attibute topology, Control Decision 39 (10) (2024) 3328–3336.
[64] F. Hao, D.-S. Park, D.-S. Sim, M.J. Kim, Y.-S. Jeong, J.-H. Park, H.-S. Seo, An efficient approach to understanding social evolution of location-focused online communities in location-based services, Soft Comput. 22 (2018) 4169–4174.
[65] M. Yan, J. Li, W. Liu, W. Zhang, Conceptual knowledge discovery and evolution in formal context with object structure information, Acta Electronica Sinica 51 (1) (2023) 1–17.
[66] S. Ai, Q. Wan, J. Li, Network formal concepts acquisition based on equiconcepts, Pattern Recognit. Artif. Intell. 38 (5) (2025) 412–424.
[67] M. Liu, P. Zhu, Fuzzy object-induced network three-way concept lattice and its attribute reduction, Int. J. Approximate Reasoning 173 (2024) 109251.
[68] R. Ren, L. Wei, Y. Yao, An analysis of three types of partially-known formal concepts, Int. J. Mach. Learn. Cybern. 9 (2018) 1767–1783.
[69] J. Gao, F. Hao, Z. Pei, G. Min, Learning concept interestingness for identifying key structures from social networks, IEEE Trans. Network Sci. Eng. 8 (4) (2021) 3220–3232.
[70] S. Shi, M. Yan, J. Li, Overlapping community detection based on weak equiconcept, IEEE Access 12 (2024) 42147–42162.
[71] P.M. Pardalos, J. Xue, The maximum clique problem, J. Global Optim. 4 (1994) 301–328.
[72] Q. Wu, J.-K. Hao, A review on algorithms for maximum clique problems, Eur. J. Oper. Res. 242 (3) (2015) 693–709.
[73] C. Lu, J.X. Yu, H. Wei, Y. Zhang, Finding the maximum clique in massive graphs, Proc. VLDB Endowment 10 (11) (2017) 1538–1549.
[74] G. Palla, I. Derényi, I. Farkas, T. Vicsek, Uncovering the overlapping community structure of complex networks in nature and society, Nature 435 (7043) (2005) 814–818.
[75] J. Qi, L. Wei, Y. Yao, Three-way formal concept analysis, in: Proceedings of Rough Sets and Knowledge Technology, 2014, pp. 732–741.
[76] B. Scholkopf, Causality for machine learning, in: Probabilistic and Causal Inference: The Works of Judea Pearl, 2022, pp. 765–804.
[77] M. Schlichtkrull, T.N. Kipf, P. Bloem, R. Van, I. Titov, M. Welling, Modeling relational data with graph convolutional networks, in: The 15th International Conference on Semantic Web, 2018.
[78] Y. Dai, M. Yan, J. Li, Granular concept-enhanced relational graph convolution networks for link prediction in knowledge graph, Inf. Sci. 694 (2025) 121698.
[79] E.-k. Kim, K.-S. Choi, Entity summarization based on formal concept analysis, in: 1st International Workshop on Entity Retrieval, EYRE, 2018.
[80] E. Yang, F. Hao, Y. Yang, C. De Maio, A. Nasridinov, G. Min, L.T. Yang, Incremental entity summarization with formal concept analysis, IEEE Trans. Serv. Comput. 15 (6) (2021) 3289–3303.
[81] M. Shao, Z. Hu, W. Wu, H. Liu, Graph neural networks induced by concept lattices for classification, Int. J. Approximate Reasoning 154 (2023) 262–276.
[82] K. Wu, C. Zhang, F. Hao, J. Li, Q. Wan, K. Park, X. Qin, V. Loia, Fc-gcn: a formal concept-enhanced graph convolution network model, Soft Comput. 29 (6) (2025a) 2715–2725.
[83] K. Wu, F. Hao, R. Yao, J. Li, G. Min, S.O. Kuznetsov, Traffic prediction based on formal concept-enhanced federated graph learning, IEEE Trans. Intell. Transp. Syst. 26 (5) (2025) 6936–6948.
[84] A. Khaund, A.M. Sharma, A. Tiwari, S. Garg, S. Kailasam, RD-FCA: A resilient distributed framework for formal concept analysis, J. Parallel Distrib. Comput. 179 (2023) 104710.
[85] Y. Hu, Y.Z. Hu, Z. Su, X.L. Li, Z. Meng, W.J. Tian, Y.Y. Yang, J.F. Chai, An incremental algorithm for concept lattice based on structural similarity index, Soft Comput. 26 (21) (2022) 11409–11423.
[86] L. Kovács, Conceptual clustering with application on FCA context, Expert Syst. Appl. 245 (2024) 123013.
[87] M. Awais, M. Naseer, S. Khan, R.M. Anwer, H. Cholakkal, M. Shah, M.-H. Yang, F.S. Khan, Foundation models defining a new era in vision: a survey and outlook, IEEE Trans. Pattern Anal. Mach. Intell. (2025).
[88] F. Hao, G. Pang, Z. Pei, K. Qin, Y. Zhang, X. Wang, Virtual machines scheduling in mobile edge computing: a formal concept analysis approach, IEEE Trans. Sustainable Comput. 5 (3) (2019) 319–328.