Interrelated Dense Pattern Detection in Multilayer Networks

Wenjie Feng, Li Wang, Bryan Hooi, See Kiong Ng, and Shenghua Liu

***Abstract*—Given a heterogeneous multilayer network with various connections in pharmacology, how can we detect components with intensive interactions and strong dependencies? Can we accurately capture suspicious groups in a multi-lot transaction network under camouflage? These challenges related to dense subgraph detection have been extensively studied in simple graphs (such as bipartite graph, multi-view network) but remain under-explored on complex networks. Existing methods struggle to effectively handle the *intricate dependencies*, let alone accurately identify the *interrelated dense connected patterns* within a series of complex heterogeneous networks. In this paper, we propose INDUEN, a novel algorithm designed to detect interrelated densest subgraphs in multilayer networks through joint optimization of coupled factorization and local search for an elaborate-designed joint density measure. It is (a) effective for both large synthetic and real networks, (b) resistant to camouflage for anomaly detection, and (c) linearly scalable. Experimental results demonstrate that INDUEN outperforms the state-of-the-art baselines in accurately detecting interrelated densest subgraphs under various settings. Furthermore, INDUEN uncovers some intriguing patterns in real-world data, i.e., closely cooperated academic groups and interrelated dependent functional components in biology-net. INDUEN achieves more than 35× speedup compared to the SOTA method DESTINE.**

***Index Terms*—Multilayer network, dense subgraph detection, interrelated pattern mining, algorithm design.**

I. INTRODUCTION

CONSIDERING a collection of complex networks that rely upon one another, such as the intricate associations among genes, chemicals, and various diseases for humans, how can we detect heterogeneous components with solid interactions and mutual dependencies relevant to pharmacology? Also, how can we capture related party transactions or collusion fraud when facing elaborate camouflage in a suspicious multi-lot transnational money transfer network for banks?

Prevalent among the various data in the natural, social, and information sciences are intricate interactions and extensive interconnections, which leads to *complex, heterogeneous, and multilayer* networks [1], [2], including relations among {gene, protein, phenotype, disease} in life science [3], dependencies of {molecule, cell, targeted drug} in network pharmacology [4], and interrelations between {energy-sites, transportation, food-demand} in a physical society [5], etc. As a common knowledge, one can gain richer information about structures and knowledge by fusing different sources and related data simultaneously [6], [7]. Howbeit, modelling and capturing the high-order dependencies [8] in a complex homogeneous graph is notoriously challenging or becomes even worse for joint analysis of plenty of various sources [9].

Dense subgraph detection, aiming to extract tight-connected nodes from graphs, is a primary task in data mining with numerous applications, such as detecting communities [10] and spotting fraudulent users/behaviours [9], [11]; finding such intensive, interactive components is also essential for exploring several properties or detecting interesting patterns in real-world scenarios. However, most existing researches designed different algorithms for detecting dense subgraphs [12] but mainly focus on single graphs [11], [13], [14] or a pair of networks [15], [16]; others also extend to detecting dense blocks in tensors to utilize side information [17], [18]. In the case of complex networks, [19], [20], [21], [22] can find some dense patterns w.r.t. expert-designed motifs or communities in heterogeneous networks; almost all existing research [12], [23], [24], [25] that claims to work on multilayer networks is actually multi-view networks, which is just a simplified form that includes different types of edges over the same set of vertices, and they *cannot be extended to more complex graphs where nodes and edges are heterogeneous*. As far as we know, DESTINE [26] is the only one currently available that can indeed detect dense subgraphs in a multilayer network by considering cross-layer dependencies, while it is *neither memory-efficient nor efficient for large networks and without an interrelation guarantee for the detected subgraphs*. Hence, general and efficient dense pattern detection in heterogeneous multilayer networks remains a great challenge.

Moreover, in various real applications, e.g., social networks and marketing campaigns, we are not only concerned about the connections within a subgraph component but are more

Manuscript received 18 November 2023; revised 1 May 2024; accepted 6 May 2024. Date of publication 22 May 2024; date of current version 27 September 2024. This work was supported in part by the National Science Foundation of China under Grant U21B2046 and Grant 6237075198, in part by the National Research Foundation Singapore and DSO National Laboratories through AI Singapore Programme (AISG) under Award AISG2-RP-2020-018, and in part by the National Key R&D Program of China under Grant 2023YFC3305303. Recommended for acceptance by S. Whang. *(Wenjie Feng and Li Wang contributed equally to this work.) (Corresponding authors: Wenjie Feng and Shenghua Liu.)*

Wenjie Feng is with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China, and also with the Institute of Data Science, National University of Singapore, Singapore 117602 (e-mail: wenchiehfeng.us@gmail.com).

Li Wang and Shenghua Liu are with the CAS Key Laboratory of AI Security, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China, and also with the University of Chinese Academy of Sciences, Beijing 100190, China (e-mail: wl.tsinghua.ee@gmail.com; liushenghua@ict.ac.cn).

Bryan Hooi and See Kiong Ng are with the Institute of Data Science, National University of Singapore, Singapore 117602 (e-mail: dcsbhk@nus.edu.sg; seekiong@nus.edu.sg).

Digital Object Identifier 10.1109/TKDE.2024.3398683

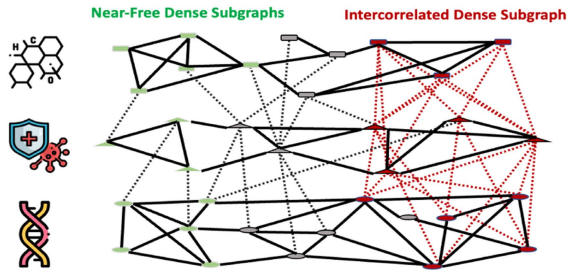


Fig. 1. A toy example of a multilayer network of {Gene, Disease, Chemical} in bio-medicine, where links of within-layer denote similarities (solid lines) and of cross-layers are 'which chemical-cures-which disease', 'which gene-related to- which disease' (dashed lines). The subgraph marked with red constitutes the '*interrelated dense subgraph*' with higher density when considering all links, while the subgraphs marked with green correspond to some '*near-free*' dense subgraphs for all layers for very few inter-layer links between them.

inclined to detect components with strong interrelationships if they are interdependent. Such *interrelated dense subgraphs* benefit to study strong complex relationships and downstream tasks, e.g., targeted drug research & development and effective therapies, while near-free dense subgraphs in each within-layer only reflect the great similarities of the local structure inside it but cannot reflect interactions or dependencies between different entities. Considering a multilayer bio-medicine network for the relationship among {Gene, Disease, Chemical} in Fig. 1, it contains useful interrelated dense subgraph patterns with strong dependencies among all types of nodes in three layers.

To this end, we propose INDUEN, a novel approach for the *interrelated densest subgraphs detection* (IDSD) in a multilayer network. INDUEN is a flexible framework with configurable components, and it adopts coupled factorization to maintain intricate dependencies. Our algorithm consistently optimizes the objective by virtue of a joint of continuous and discrete optimization with a guaranteed convergence in theory. Resorting to the elaborate-designed density measure and algorithm design, INDUEN becomes highly efficient, effective for the IDSD task, and resistant to camouflage attacks. In addition, INDUEN outperforms baselines for both synthetic and real networks; it uncovers various intriguing patterns in real-world data, e.g., close cooperation relations in academic groups and different interrelated functional components in biological networks. In addition, INDUEN is scalable, with near-linear runtime in the number of non-zeros of a network, and achieves more than $35\times$ speedup compared to the SOTA approach DESTINE.

In summary, our main contributions are as follows:

- *Problem Formulations:* We formulate the interrelated dense subgraph detection problem in multipartite graphs and multilayer networks, and design an effective density metric adapted to the case.
- *Algorithm and Analysis:* We propose INDUEN, a fast and effective approach for the IDSD problems, and analyse its optimization convergence and complexity.
- *Evaluations:* We conduct extensive experiments to verify the efficiency, effectiveness, and linear scalability of INDUEN. It can also find intriguing and valuable patterns across various real application scenarios.

Reproducibility: Code, datasets, and supplementary for proofs and more experimental results are available online.[1]

II. RELATED WORK

A. Multilayer Networks

Rich cross-domain interactions induce extensive research for complex networks, including inter-dependent, multi-modal, multiplex, multi-view, multidimensional, multilayer, etc. We refer interested readers to the comprehensive survey [1] for detailed information and their differences. Chen et al. [7], [25] proposed MULAN to model the intricate dependencies among layers in multilayer networks. NONCLUS [27] clusters multiple domain-specific networks, utilizing non-negative matrix factorization. Also, multilayer networks have various applications, such as biomedicine [3], biological [4], [28], and human behaviour [29], etc.

B. Dense Subgraph Detection (DSD)

The detection of dense subgraphs has been extensively studied [12], [30], [31]. Finding the densest subgraphs is polynomial-time solvable with linear programming [13] or maximum flow algorithms [32]. Instead of detecting the exact densest subgraph, Greedy algorithms give solutions within a $1/2$ factor of optimum [33], [34]. These methods have been applied to detect community structure [10], [35], protein complexes [36], and anomaly [11], [37], e.g., follower-buying service [11], ill-gotten "likes" on Facebook [38], and money laundering of transaction flows [9], [39]. It has been extended to detect dense blocks in multidimensional data (tensor) [17], [18], [40], [41], which actually correspond to multi-view graph — one special type of multilayer network.

Beyond the single graph, detecting dense components in multiple graphs aims to leverage the rich side information (i.e., attributes). [42] maximized the minimum density of a subgraph sharing the same vertex set. [9] extended Greedy algorithm for detecting dense patterns as suspicious users in a multipartite graph. [43] proposed to find the densest connected subgraph in dual networks, where the subgraph needs to be connected in the physical network and has the largest density in the conceptual network. [44] proposed using core decomposition to discover the densest subgraphs in multi-view networks. [45] introduced a method that enforces k-edge-connectivity constraints on one graph while detecting dense components in another. [16] puts forth the generalized densest subgraph problem and designed a unified SPECGREEDY by leveraging the spectral graph properties to detect dense subgraphs.

Based on the most recent exhaustive survey [12], *almost all works focus on detecting dense subgraphs on the multi-view graph with some edge sets encoding different types of connections and/or time-dependent connections over the same set of vertices.* DESTINE [26] focuses on dense subgraph discovery

[1] http://github.com/wenchieh/induen.

in an actual multilayer network by considering the cross-layer dependencies and optimizes a non-strictly convex objective function based on the projected gradient descent. Another way to detect the dense subgraph in the multi-graph is based on some predefined motifs [46] while considering the label of node and edge, e.g., small clique, k-core, and k-truss; similar patterns and problems have been widely studied for homogeneous/bipartite/multi-view/directed/temporal graphs [47], [48], [49] and heterogeneous information networks [22], [50]. Also, [19], [20] studies the DSD problem in heterogeneous networks to consider different types of nodes and edges based on expert-designed meta-paths. The tutorial [51] is a comprehensive survey for cohesive subgraph mining in heterogeneous information networks. As we can imagine, it is hard to pre-define proper motifs or enumerate all motifs, considering the combinations of different node and edge labels, and it becomes even worse for complex graphs with more layers and more types of relations. *Those motif-based or meta-path-based methods can only find some fixed-type connection patterns; however, they cannot count the different contributions of various kinds of connections or edge weights* (e.g., the similarity and the number of interactions).

Moreover, the dense subgraph detection problem is usually deemed an unsupervised task, and few works focus on the subgraph topics [52], [53] but none on the DSD problem. Zhao et al. [54] try to use some detection results from the dense subgraph/block detection methods before detecting graph anomalies with the GNN model.

C. Coupled Factorization

To model the complex interactions in heterogeneous, multidimensional, and multiple source data, and to perform data fusion, coupled factorization has been developed significantly, which can gain insights by sharing the same latent factors. There are different applications, including recommendation systems [55], [56], pattern mining [57], forecasting [58], link prediction [59], community detection [60], [61], etc. Acar et al. [62] proposed CMTF-OPT, a first-order optimization algorithm for Coupled Matrix-Tensor Factorization. [63] proposed coupled non-negative matrix factorization for hyper-spectral and multi-spectral data fusion. TensorCast [58] leveraged coupled tensor-tensor factorization for forecasting with auxiliary information, enhancing accuracy and efficiency. [61] found tightly connected subgroups of nodes that exhibit similar node-specific time series by the coupled clustering.

III. Notions & Preliminary

We summarize the notions used in the paper, introduce coupled factorization and dense subgraph detection methods.

We use bold uppercase for matrices (e.g., $\mathbf{A}$), bold lowercase for vectors (e.g., $\boldsymbol{x}$), normal lowercase for scalars (e.g., a), and calligraphic letters for sets (e.g., $\mathcal{A}$). For indexing the elements of a matrix, we follow the Matlab settings. The i-th row of the matrix $\mathbf{A}$ is denoted as $\mathbf{A}(i,:)$, the j-th column as $\mathbf{A}(:,j)$, and the (i,j)th entry as $\mathbf{A}(i,j)$; $\mathbf{A}'$ denotes the transpose of the matrix $\mathbf{A}$, and $\mathbf{tr}(\mathbf{A})$ denotes the trace of $\mathbf{A}$ if it is a square matrix.

TABLE I
Symbols and Definitions Used in the Paper

Symbols	Definition
$\bar{G}=(\mathcal{V},\emptyset,\bar{\mathcal{C}},\bar{\mathbf{G}})$	Multipartite graph
$G=(\mathcal{V},\mathcal{A},\mathcal{C},\mathbf{G})$	multilayer network
$\mathcal{V}$	Node-set and $\mathcal{V}=\{\mathcal{V}_i \mid i\in[g]\}$
$\mathcal{A}$	Set of matrices of within-layer graphs
$\mathcal{C}$	Set of matrices of cross-layers dependencies
$\mathbf{G}$	Layer-layer dependency matrix
$\mathbf{A},\mathbf{C}$	Adjacency matrices of within, cross layers
N	Number of graphs in the multilayer networks.
$\otimes$	Hadamard (entrywise) product

The Frobenius norm of the matrix $\mathbf{A}$ is $||\mathbf{A}||_F$, and the $L_{1,1}$ norm is $||\mathbf{A}||_{1,1}=\sum_{i,j}|\mathbf{A}(i,j)|$. We use $\mathbf{0}$ and $\mathbf{J}$ to denote an all-zero and an all-one matrix, respectively. $[K]\equiv\{1,\ldots,K\}$ for brevity. Table I summarizes the main symbols used in the paper.

A. Multilayer Network & Multipartite Graphs

1) Multilayer Networks: Considering a typical multilayer network with g layers, we use $\mathbf{G}$ to denote the $g\times g$ layer-layer dependencies, $\mathbf{G}(i,j)=1$ if the j-th layer has links with the i-th layer, otherwise $\mathbf{G}(i,j)=0$. Assume the nodeset of the i-th layer is $\mathcal{V}_i$ with size $n_i>0$, and m_i edges between nodes of $\mathcal{V}_i$ forms a homogeneous graph; we use $\mathcal{A}=\{\mathbf{A}_1,\ldots,\mathbf{A}_g\}$ as a set of g adjacency matrices with $\mathbf{A}_i\in\mathbb{R}_+^{n_i\times n_i}$ for the i-th layer, $\mathcal{C}=\{\mathbf{C}_{i,j}\in\{0,1\}^{n_i\times n_j}|\forall i,j\in[g]\wedge i\neq j\}$ as the cross-layers network dependencies, and $\mathbf{C}_{i,j}=\mathbf{0}$ iff $\mathbf{G}(i,j)=0$. We denote the number of graphs in the multilayer networks as N, where $N=|\mathcal{A}|+|\mathcal{C}|/2$. Considering the undirected dependencies, $\mathbf{C}_{i,j}(s,t)=1$ if the node s in the i-th layer and the node t in the j-th layer depend on each other, which naturally leads to $\mathbf{C}_{j,i}(t,s)=1$ as a consequence.

Hence, a multilayer network is denoted as $G=(\mathcal{V},\mathcal{A},\mathcal{C},\mathbf{G})$ where $\mathcal{V}=\cup_{i=1}^{g}\mathcal{V}_i$ is the universal nodeset with size $|\mathcal{V}|=\sum_{i=1}^{g}n_i$ and $|\mathcal{C}|\leq g(g-1)$. Besides, we use $G(\mathcal{S})$ to denote the multilayer subgraph of G derived by a node subset $\mathcal{S}\subseteq\mathcal{V}$ and all corresponding connections.

2) Multipartite Graphs: Under the above paradigm, each non-missing cross-layer matrix $\mathbf{C}_{i,j}$ actually represents a bipartite graph if $\mathcal{V}_i\cap\mathcal{V}_j=\emptyset$, and G is a g-partite graph if $\{\mathbf{A}_i=\mathbf{0}\,|\,\forall\mathbf{A}_i\in\mathcal{A}\}$, we use $\bar{G}=(\mathcal{V},\emptyset,\bar{\mathcal{C}},\bar{\mathbf{G}})$ to denote a g-partite graph, $\mathbf{C}_{i,i+1}\in\bar{\mathcal{C}}$ is cross-partite connections between the i-th and next part.

B. Coupled Factorization

For a matrix $\mathbf{X}\in\mathbb{R}^{M\times N}$, matrix decomposition finds the factor matrices $\mathbf{U}\in\mathbb{R}^{M\times K}$ and $\mathbf{V}\in\mathbb{R}^{N\times K}$ to minimize the squared reconstruction error, $\min_{\mathbf{U},\mathbf{V}}||\mathbf{X}-\sum_{i=1}^{K}\mathbf{U}(:,i)\cdot\mathbf{V}(:,i)'||_F^2$, with $0<K\ll N$ as the number of latent factors. The optimal solution of this low-rank approximation is given by the Singular Value Decomposition (SVD) of $\mathbf{X}$ based on the Eckart–Young–Mirsky theorem [64].

To include other related resource data, e.g., profile information, for co-analysis and correlated prediction, *coupled matrix*

factorization naturally extends the standard matrix decomposition. The factorization of $\mathbf{X} \in \mathbb{R}^{M\times N}$ coupled with $\mathbf{Y} \in \mathbb{R}^{M\times T}$ on the first mode is obtained by minimizing

$$\min_{\mathbf{U},\mathbf{V},\mathbf{W}} ||\mathbf{X} - \mathbf{U}\mathbf{V}'||_F^2 + \beta||\mathbf{Y} - \mathbf{U}\mathbf{W}'||_F^2;$$

where $\mathbf{U} \in \mathbb{R}^{M\times K}, \mathbf{V} \in \mathbb{R}^{N\times K}$, and $\mathbf{W} \in \mathbb{R}^{T\times K}$, β is the hyperparameter as the strength of the coupling for the task.

Many techniques have been proposed to solve the above optimization problem, like multiplicative update rules and additive update rules (e.g., projected SGD) [65].

C. Densest Subgraph Detection (DSD)

Given an undirected graph $G = (\mathcal{V}, \mathcal{E})$ with edgeset $\mathcal{E}$, the commonly used average degree density metric for the DSD problem is defined as $\rho(G) = \frac{|\mathcal{E}|}{|\mathcal{V}|}$. As for detection methods, the Max-Flow-based algorithm [32] and the LP-based algorithm [13] can give exact solutions in polynomial time. The greedy algorithm [66] is computationally efficient and has a $\frac{1}{2}$-approximation optimal guarantee, i.e., $\rho^* \geq \frac{1}{2}\rho_{opt}$; Greedy++ [67] is also a variant. We use MAXFLOWSOLVER and GREEDYSOLVER to denote these algorithms, respectively.

Besides, given the adjacency matrix $\mathbf{A}$ of G, there are some matrix decomposition based approximate detection methods (named MF-DSD), including

- *EIGENSOLVER:* using eigen-decomposition of matrix, $\mathbf{A} \approx \mathbf{U}\mathbf{\Lambda}\mathbf{U}'$ with $\mathbf{\Lambda} = diag(\lambda_1, \ldots, \lambda_K)$ as the non-decreasing top K ($\ll |\mathcal{V}|$) eigenvalues.[2]
- *NMFSOLVER:* using the symmetric non-negative matrix decomposition $\min_{\mathbf{U}\geq\mathbf{0}} ||\mathbf{A}_G - \mathbf{U}\cdot\mathbf{U}'||_F^2$.

where $\mathbf{U} \in \mathbb{R}^{|\mathcal{V}|\times K}$ is the top-K decomposition factors.

Considering the fact that vertices with a larger value in the top decomposition factors manifest their essential contribution to the graph and usually make up some densely connected groups, SPOKEN [68] and FBOX [69] utilize such an idea to detect suspicious behaviour patterns in the graph.

IV. PROBLEM FORMULATIONS

The densest subgraph in a graph is determined by the density measure in theory, and different forms of density definitions will derive distinct densest subgraphs.

In contrast to the simple graph, a valid dense subgraph in a multilayer network should cover all layers as we expect, not just some of them, and should have strong interconnections through cross-layer dependencies between nodes of different layers, rather than being loosely connected. To that end, we account for the intricate contributions of the within layers and even more complicated cross-layer dependencies and propose a flexible density measure for dense subgraphs in a multilayer network as follows,

Definition 1 (Joint Density): Given a g-layered network $G = (\mathcal{V}, \mathcal{A}, \mathcal{C}, \mathbf{G})$ and a nodeset $\mathcal{S} = \{\mathcal{S}_i \,|\mathcal{S}_i \subseteq \mathcal{V}_i, \forall\, i \in [g]\}$, let $\mathbf{A}_i^{\mathcal{S}}$ be the i-th within-layer's adjacency matrix derived by $\mathcal{S}$ and $\mathbf{C}_{i,j}^{\mathcal{S}}$ is similar w.r.t. the cross-layer dependencies. $||\mathbf{A}_i||_{1,1}$ equals twice the number of edges in the corresponding graph w.r.t. $\mathbf{A}_i$, $||\mathbf{C}_{i,j}||_{1,1}$ is the number of connections in the graph of $\mathbf{C}_{i,j}$.

Then, we define the joint density of the subgraph $G(\mathcal{S})$ as

$$\rho_G(\mathcal{S};\gamma_\cdot) = \sum_{i=1}^{g} \frac{||\mathbf{A}_i^{\mathcal{S}}||_{1,1} + \sum_{j=1}^{g} \gamma_{i,j}\cdot \mathbf{G}(i,j)\cdot ||\mathbf{C}_{i,j}^{\mathcal{S}}||_{1,1}}{2\cdot|\mathcal{S}|} \cdot \prod_{i=1}^{g} \mathbb{1}(|\mathcal{S}_i| > 0), \quad (1)$$

where $\gamma_{i,j}$ is a preset parameter to balance the importance of $\mathbf{C}_{i,j}$ w.r.t. the within-layer connections, $\mathbb{1}(\cdot)$ is the indicator function.

As for (1), the first part is the sum of γ-weighted average density involving all connections in the derived multilayer subgraph, considering both within-layer and cross-layer connections (γ for different importance); and $\prod_{i=1}^{g} \mathbb{1}(|\mathcal{S}_i| > 0)$ as penalization to ensure the none-empty of node subset in each layer. *Different γs will lead to distinct densest subgraph structures because of the different emphasis on within and cross-layers.* We can also naturally define other measures similar to (1) for multilayer networks based on other commonly used density measures [12], [70], e.g., edge ratio, triangle density, and triangle ratio, etc.

Furthermore, the majority of current densest subgraph detection methods primarily concentrate on the monopartite or bipartite graph; *how to handle the heterogeneous multilayer networks is still under-explored*. To that end, we propose a generalized dense subgraph detection problem as,

Problem 1 (Interrelated Densest Subgraph Detection (IDSD) in Multilayer Networks). Given: a g-layer network $G = (\mathcal{V}, \mathcal{A}, \mathcal{C}, \mathbf{G})$ and the density measure $\rho(\cdot)$;

Find: a set of nodes $\mathcal{S}^* = \{\mathcal{S}_i^* \,|\, \emptyset \subsetneq \mathcal{S}_i^* \subseteq \mathcal{V}_i, \forall\, i \in [g]\}$, which composes the target interrelated densest subgraph, that is, $\mathcal{S}^* = \text{argmax}_{\mathcal{S}\triangleq\{\mathcal{S}_i\subseteq\mathcal{V}_i, \forall\, i\in[g]\}} \rho_G(\mathcal{S})$.where $\rho_G(\mathcal{S})$ measures the density of the subgraph derived by the subset $\mathcal{S}$ over $\bar{G}$.

As we can see, when all cross-layer dependencies are absent, the network G degenerates into g separate graphs, $\{\mathbf{A}_1, \ldots, \mathbf{A}_g\}$, the DSD problem in this case can be solved independently, resulting in some unrelated (near-free) results.

Consider that a good deal of cascade behaviour pattern mining problem actually corresponds to the densest subgraph detection in a multipartite graph, e.g., money laundering behaviour detection in transaction flows [9] and heterogeneous information integration (i.e., gene-protein-disease in functional genomics [71]). Hence, we formalise the following problem as the other special case of Problem 1,

Problem 2 (Densest Subgraph Detection in Multipartite Graphs). Given: a g-partite graph $\bar{G} = (\mathcal{V}, \emptyset, \bar{\mathcal{C}}, \bar{\mathbf{G}})$ and the density measure $\rho(\cdot)$;

Find: a set of independent node-sets $\bar{\mathcal{S}}^* = \{\bar{\mathcal{S}}_i^* \,|\, \emptyset \subsetneq \bar{\mathcal{S}}_i^* \subseteq \mathcal{V}_i, \forall\, i \in [g]\}$, which composes the detected connected densest subgraph, i.e., $\bar{\mathcal{S}}^* = \text{argmax}_{\bar{\mathcal{S}}\triangleq\{\bar{\mathcal{S}}_i\subseteq\mathcal{V}_i, \forall\, i\in[g]\}} \rho_{\bar{G}}(\bar{\mathcal{S}})$.

[2] If $\mathbf{A}_G \in \mathbb{R}^{M\times N}$, SVD is used as $\mathbf{A} \approx \mathbf{U}\mathbf{\Sigma}\mathbf{V}'$, where $\mathbf{\Sigma} = diag(\sigma_1, \ldots, \sigma_K)$ is the diagonal matrix of top-K singular values (non-negative), $\mathbf{U} \in \mathbb{R}^{M\times K}$ and $\mathbf{V} \in \mathbb{R}^{N\times K}$.

V. Proposed Methods

A. MF-DSD Over the Aggregated Graph

For a multipartite graph $\hat{G}$ with none-overlapping within-layer nodesets, by treating all nodes equally, even if they could be heterogeneous, we can represent their connections with an aggregated big matrix $\mathbf{A}_{\bar{G}} \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$. The following shows an example of a three-partite graph $\mathbf{A}_{\bar{G}}$.

$$\mathbf{A}_{\bar{G}} = \begin{pmatrix} \mathbf{0} & \mathbf{C}_{1,2} & \\ \mathbf{C}_{2,1} & \mathbf{0} & \mathbf{C}_{2,3} \\ & \mathbf{C}_{3,2} & \mathbf{0} \end{pmatrix}, \hat{\mathbf{G}} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}.$$

Similarly, we can use a big symmetric block matrix $\mathbf{A}_G \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$ to represent a multilayer network G, where $\mathbf{C}_{i,j}$ as the off-diagonal submatrices can be a zero-matrix if the corresponding dependency is absent. The following shows an example of $\mathbf{A}_G$ for a three-layer network.

$$\mathbf{A}_G = \begin{pmatrix} \mathbf{A}_1 & \mathbf{C}_{1,2} & \mathbf{C}_{1,3} \\ \mathbf{C}_{2,1} & \mathbf{A}_2 & \mathbf{C}_{2,3} \\ \mathbf{C}_{3,1} & \mathbf{C}_{3,2} & \mathbf{A}_3 \end{pmatrix}. \tag{2}$$

Thus, $\mathbf{A}_G$ ($\mathbf{A}_{\bar{G}}$) signifies an undirected connected graph.

Superficially, we could detect the dense subgraph over this aggregated graph directly; the detection results given by GreedySolver, MaxFlowSolver, and MF-DSD, however, will be problematic because they *cannot guarantee to satisfy the constraints that* $\mathcal{S}_i^* \neq \emptyset$ *for* $\forall\, i \leq g$ *and their interrelationship* when the connections are dominated by some partite, some within-layers, and/or cross-layers. Moreover, detecting the dense subgraph over all $\mathbf{A}_i$ separately *cannot guarantee the subgraphs' close interrelation between layers*, which *results in (near) free dense subgraphs for all layers*.

Consequently, all the above approaches and adopted density metrics cannot be used to find the interrelated dense subgraphs we expect in a multilayer or multipartite graph.

B. Proposed InDuen Algorithm

To solve Problems 1–2 efficiently while addressing the above issues, hereby, we devise a new heuristic algorithm InDuen, Interrelated Densest Subgraph Detection on Networks, which consists of the following three components,

i) *Coupled Non-negative Factorization (CoNF):* the factorization will tie together the within-layer graphs and cross-layer dependencies and find their best rank-1 factors.
ii) *IDSD Detector:* we use some efficient detectors to find the interrelated densest subgraph with the above factors.
iii) *Boosting Expander (Expander):* we exploit the local structure by neighbour expansion for the rough results to boost the fine detection subgraph.

In the following sections, we describe each step in detail and illustrate the complete algorithm at the end.

1) Coupled Non-Negative Factorization:

a) DSD over Multipartite Graphs: For a g-partite graph $\bar{G}$, the nodeset $\mathcal{V}_i$ is the bridge connecting $\mathbf{C}_{i-1,i}$ and $\mathbf{C}_{i,i+1}$ for $1 < i < g$; and $\mathcal{V}_1$ ($\mathcal{V}_g$) is only affiliated with $\mathbf{C}_{1,2}$ ($\mathbf{C}_{g-1,g}$). Therefore, let $\mathbf{U}_i \in \mathbb{R}^{n_i\times K}$ be the factors associated with $\mathcal{V}_i$ for any $i \in [g]$, we formulate the coupled factorization for all $\mathbf{C}_{i,i+1} \in \bar{\mathcal{C}}$ of $\bar{G}$ as

$$\begin{aligned} \min_{\{\mathbf{U}_i,\mathbf{\Sigma}_i\}} \quad & \sum_{i=1}^{g-1} \alpha_i \cdot ||\mathbf{C}_{i,i+1} - \hat{\mathbf{C}}_{i,i+1}||_F^2 \\ \text{s.t.} \quad & \hat{\mathbf{C}}_{i,i+1} = \mathbf{U}_i\mathbf{\Sigma}_i\mathbf{U}'_{i+1},\ \alpha_i > 0,\ \mathbf{\Sigma}_i > \mathbf{0}, \forall\, i \in [g-1] \\ & \mathbf{U}'_j\mathbf{U}_j = \mathbf{I},\ \mathbf{U}_j \geq \mathbf{0},\ \forall j \in [g]. \end{aligned} \tag{3}$$

where $\mathbf{\Sigma}_i$ is a positive diagonal matrix and $\alpha_i > 0$ is the hyperparameter denoting the importance of the i-th reconstruction term for the corresponding cross-layer dependence. Non-negative constraints are applied for all factors.

b) IDSD over Multilayer Networks: For a g-layer network $G = (\mathcal{V}, \mathcal{A}, \mathcal{C}, \mathbf{G})$, the nodeset $\mathcal{V}_i$ affiliates with $\mathbf{A}_i$ and also connects with $\mathbf{C}_{i,j}$ if $\mathbf{G}_{i,j} = 1$ for any $j \neq i$ and $j \leq g$. Let $\mathbf{U}_i \in \mathbb{R}^{n_i\times K}$ be the factors associated with $\mathcal{V}_i$ for any $i \in [g]$, we formulate the coupled factorization for all $\mathbf{A}_i \in \mathcal{A}$ and $\mathbf{C}_{i,j} \in \mathcal{C}$ as follows,

$$\begin{aligned} \min_{\{\mathbf{U}_i,\mathbf{\Lambda}_i,\mathbf{\Sigma}_{i,j}\}} \quad & \sum_{i=1}^{g} \Bigg(||\mathbf{A}_i - \hat{\mathbf{A}}_i||_F^2 \\ & + \sum_{j>i}^{g} \beta_{i,j} ||\mathbf{C}_{i,j} - \hat{\mathbf{C}}_{i,j}||_F^2 \Bigg) \\ \text{subject to} \quad & \hat{\mathbf{A}}_i = \mathbf{U}_i\mathbf{\Lambda}_i\mathbf{U}'_i,\ \hat{\mathbf{C}}_{i,j} = \mathbf{U}_i\mathbf{\Sigma}_{i,j}\mathbf{U}'_j, \\ & \mathbf{U}_i \geq \mathbf{0},\ \mathbf{U}'_i\mathbf{U}_i = \mathbf{I},\ \mathbf{\Lambda}_i \geq \mathbf{0},\ \mathbf{\Sigma}_{i,j} \geq \mathbf{0}, \\ & \beta_{i,j} \geq 0,\ \forall\, i,j \in [g]. \end{aligned} \tag{4}$$

where $\mathbf{\Lambda}_i$ and $\mathbf{\Sigma}_{i,j}$ are non-negative diagonal matrices, $\beta_{i,j} \geq 0$ is the hyperparameter to define the importance of the i-th reconstruction term of the cross-layer dependencies $\mathbf{C}_{i,j}$. $\beta_{i,j} = 0$ means ignoring the dependencies of $\mathbf{C}_{i,j}$ or it is absent. Non-negative constraints are introduced for all $\mathbf{U}_i$.

Obviously, (4) is more general than (3), and it can handle any absent of the submatrix of $\mathbf{A}_G$ in (2) while taking the constraints for each layer (partite) into consideration.

In addition, to prevent overfitting and produce sparse low-rank latent factors, we add the $L_{1,1}$ regularization in (4), which leads to

$$\begin{aligned} \min_{\{\mathbf{U}_i,\mathbf{\Lambda}_i,\mathbf{\Sigma}_{i,j}\}} \sum_{i=1}^{g} \Bigg(& ||\mathbf{A}_i - \hat{\mathbf{A}}_i||_F^2 + \sum_{j>i}^{g} \beta_{i,j} ||\mathbf{C}_{i,j} - \hat{\mathbf{C}}_{i,j}||_F^2 \\ & + \lambda(||\mathbf{U}_i||_{1,1} + ||\mathbf{\Lambda_i}||_{1,1} + \sum_{j>i}^{g} ||\mathbf{\Sigma}_{i,j}||_{1,1}) \Bigg), \end{aligned} \tag{5}$$

where λ controls the importance of the $L_{1,1}$ regularization terms, while other constraints remain as-is.

To prevent the objective function from being overwhelmed by the reconstruction error of some $\mathbf{A}_i$ or $\mathbf{C}_{i,j}$, i.e., when they are overly denser or larger than others, we set $\beta_{i,j}$ as dispersion

Algorithm 1: CoNF: Coupled Non-Negative Factorization for Multilayer Networks.

Input: A g-layer Network $G = (\mathcal{V}, \mathcal{A}, \mathcal{C}, \mathbf{G})$; factorization rank K; hyperparameters βs, λ.
Output: Decomposed factor matrices $\{\mathbf{U}_1, \ldots, \mathbf{U}_g\}$.

1 **for** $i \leftarrow 1$ **to** g **do** $\mathbf{U}_i \in \mathbb{R}_+^{n_i \times K}$;
2 Initialize $\mathbf{\Lambda}s, \mathbf{\Sigma}s$ as identity matrix $\mathbf{I}_K$
3 **repeat**
4 **for** $i \leftarrow 1$ **to** g **do**
5 Update $\mathbf{U}_i$ with Equation (7).
6 Update $\mathbf{\Lambda}_i$ and $\mathbf{\Sigma}_{i,j}$ with Equation (8).
7 $\mathbf{\Lambda}_i \leftarrow \mathbf{\Lambda}_i \otimes diag(||\mathbf{U}_i(:,1)||_2^2, \ldots, ||\mathbf{U}_i(:,K)||_2^2)$
8 $\mathbf{\Sigma}_{i,j} \leftarrow \mathbf{\Sigma}_{i,j} \otimes diag\left([||\mathbf{U}_i(:,k)||_2 ||\mathbf{U}_j(:,k)||_2]; \forall k \in [K]\right)$
9 $\mathbf{U}_i(:,k) = \frac{\mathbf{U}_i(:,k)}{||\mathbf{U}_i(:,k)||_2}, \forall k \in [K]$.
10 **until** *all* $\mathbf{U}_i$ *are stable*;
11 **return** $\{\mathbf{U}_1, \ldots, \mathbf{U}_g\}$.

parameters [72] by

$$\beta_{i,j} = \frac{\alpha_i \cdot ||\mathbf{A}_i||_F^2 + \alpha_j \cdot ||\mathbf{A}_j||_F^2}{(\alpha_i + \alpha_j) \cdot ||\mathbf{C}_{i,j}||_F^2}, \tag{6}$$

it can be also customized according to the user's needs.

With the well-studied multiplicative update equations [65], [73], we find the solution of (4) by iterative updating rule,

$$\mathbf{U}_i \leftarrow \mathbf{U}_i \otimes \left(\frac{\sum_{j=1}^{g} \beta_{i,j} \mathbf{C}_{i,j} \mathbf{U}_j \mathbf{\Sigma}_{i,j} + 2\mathbf{A}_i \mathbf{U}_i \mathbf{\Lambda}_i}{\mathbf{U}_i \mathbf{U}_i' \left(\sum_{j=1}^{g} \beta_{i,j} \mathbf{C}_{i,j} \mathbf{U}_j \mathbf{\Sigma}_{i,j} + 2\mathbf{A}_i \mathbf{U}_i \mathbf{\Lambda}_i \right) + \lambda \mathbf{J}_i} \right)^{\frac{1}{2}}, \tag{7}$$

and the update of the diagonal matrices $\mathbf{\Lambda}_i$ and $\mathbf{\Sigma}_{i,j}$ as,

$$\mathbf{\Lambda}_i \leftarrow \mathbf{\Lambda}_i \otimes \left(\frac{\mathbf{U}_i' \mathbf{A}_i \mathbf{U}_i}{\mathbf{U}_i' \mathbf{U}_i \mathbf{\Lambda}_i \mathbf{U}_i' \mathbf{U}_i + \lambda \mathbf{J}_K} \right)^{\frac{1}{2}},$$

$$\mathbf{\Sigma}_{i,j} \leftarrow \mathbf{\Sigma}_{i,j} \otimes \left(\frac{\mathbf{U}_i' \mathbf{C}_{i,j} \mathbf{U}_j}{\mathbf{U}_i' \mathbf{U}_i \mathbf{\Sigma}_{i,j} \mathbf{U}_j' \mathbf{U}_j + \lambda \mathbf{J}_K} \right)^{\frac{1}{2}}, \tag{8}$$

where $\mathbf{J}_i$ has the same size as $\mathbf{U}_i$, $\mathbf{J}_K$ is an all-one $K \times K$ matrix. They can avoid the denominator being zero during the iteration updates.

The CoNF in Algorithm 1 shows the above factorization process for the multilayer networks. The updating of those factors will continue until convergence (Lines 4-9).

2) IDSD Detector:

a) Skewed distribution of values in Factors: As [16] shows, the scores in eigenvectors and singular vectors of the adjacency matrix of large real-world graphs are highly skewed and decreases sharply, which does benefit efficient dense subgraph detection in big graphs. Similar properties are also found in the community, Kronecker graphs, and multiple generative models in [74], [75], [76]. However, it is unclear whether the factors of coupled factorization for related graphs follow similar properties. Hence, we explore the properties of scores of the factor vectors from real-world multilayer networks.

Algorithm 2: EXPANDER: Neighbor Booster for DSD.

Input: Undirected graph $G = (\mathcal{V}, \mathcal{E})$; initial node subset $\mathcal{V}_0$; density metrc $\rho(\cdot)$.
Output: Nodeset of the densest subgraph of G.

1 $\mathcal{V}^* \leftarrow \mathcal{V}_0$
2 **while** *true* **do**
3 $\mathcal{N} \leftarrow \{v \,|\, (u,v) \in \mathcal{E} \wedge v \notin \mathcal{V}^*, \forall u \in \mathcal{V}^*\}$
4 $v^* = \arg\max_{v \in \mathcal{N}} |\{(u,v) \,|\, (u,v) \in \mathcal{E}, \forall u \in \mathcal{V}^*\}|$
5 **if** $\rho_G(\mathcal{V}^* \cup \{v^*\}) > \rho_G(\mathcal{V}^*)$ **then**
6 $\mathcal{V}^* \leftarrow \mathcal{V}^* \cup \{v^*\}$
7 **else break** ;
8 **return** $\mathcal{V}^*$

Taking AMINER network for the {author, paper, venue} relationship as an example, Fig. 2 shows the distributions of the factorization components from CoNF. We can see that the within-layer connections and cross-layer dependencies of real-world networks are sparse and have power-law degree distributions; the score of decomposition factors is also highly-skewed. Moreover, the distributions of the top diagonal elements of $\mathbf{\Lambda}$s and $\mathbf{\Sigma}$s are also skewed.

b) Densest Subgraph Detectors: Owing to the cross-layer dependencies and shared-association constraints during the coupled factorization and their skewed distribution properties, we can significantly reduce the size of the candidates by truncating with some thresholds while ensuring the completeness of the target solution as much as possible. Therefore, we can keep only those nodes with top-ranked scores in decomposed factors based on a threshold value, i.e., $\Delta = {}^1/_{\sqrt{n_k}}$ for the factor $\mathbf{U}_k$, to construct the candidate solution $\hat{\mathcal{S}} = \{\hat{\mathcal{S}}_1, \ldots, \hat{\mathcal{S}}_g\}$ with $\hat{\mathcal{S}}_k$ obtained from $\mathbf{U}_k$. It is worth noting that the size of $\hat{\mathcal{S}}$ is only $\mathbf{5\%} \sim \mathbf{15\%}$ of the original nodeset $\mathcal{V}$ in G empirically.

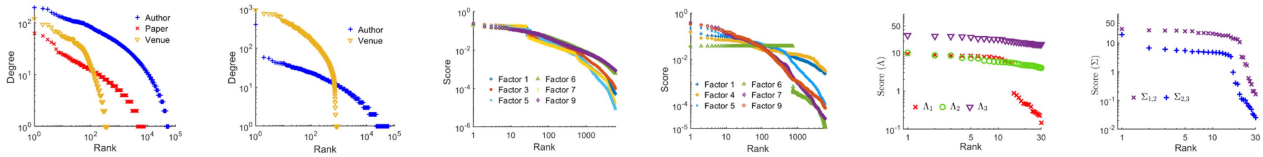
Given a multilayer network G and a candidate nodeset $\hat{\mathcal{S}}$, let G_i be the graph w.r.t. $\mathbf{A}_i$, we obtain a refined solution by the following strategies with some DSDSOLVER, which can be GREEDYSOLVER, MAXFLOWSOLVER, MAXCLIQUESOLVER, and so on.

1) *Layer-By-Layer detection:* detect the densest subgraph for each $G_i(\hat{\mathcal{S}}_i)$ and return the merged nodeset.
2) *Aggregation detection:* detect the densest subgraph over the aggregated graph w.r.t. $\{\mathbf{A}_i^{\hat{\mathcal{S}}}\} \cup \{\gamma_{i,j} \mathbf{C}_{i,j}^{\hat{\mathcal{S}}}\}$ as (2), where $\gamma_{i,j}$ re-weight the edge of cross-layers.

3) Boosting Expander: Considering the approximation results from CoNF and DSDSOLVER (maybe some approximation algorithms), we exploit the local structure of the solution returned by the previous step to boost the result. Algorithm 2 summarises the proposed EXPANDER. It tries to improve the density by greedily introducing some neighbours that haven't been in the current nodeset, until there is no update.

Similar hill climbing and local search approaches, as a type of discrete optimization technique, have also been widely used in various graph problems, e.g., subgraph patterns mining [77], influence maximization [78], anomaly detection [38], etc.

a) Complete Detection Algorithm: Algorithm 3 summarizes the complete structure of INDUEN for interrelated densest subgraph detection in multilayer networks.



(a) (b) (c) (d) (e) (f)

Fig. 2. *Some statistical properties of the multilayer networks (AMINER) are highly skewed.* (a)-(b) show the power-law distribution of node degrees based on the relations from within-layer (coauthor, paper-citation) and cross-layer ('author-write-paper', 'paper-public to-venue'). (c)-(d) show the skewed distributions of scores in top-10 factors from the non-negative coupled factorization for 'Author' and 'Paper', respectively. (e)-(f) show the distribution of diagonal elements of $\mathbf{\Lambda}$s and $\mathbf{\Sigma}$s, resulting from the factorization of within-layer adjacency matrices and cross-layers dependencies.

Algorithm 3: INDUEN: Solving the Optimal IDSD in a Multilayer Network.

Input: A g-layer Network $G = (\mathcal{V}, \mathcal{A}, \mathcal{C}, \mathbf{G})$; DSDSOLVER$(\cdot)$; density metric $\rho(\cdot)$; αs, βs, λ.
Output: Nodeset w.r.t. the densest subgraph of G.

$\mathcal{V}* \leftarrow \emptyset$
$\mathbf{U}_1, \ldots, \mathbf{U}_g \leftarrow$ CONF$(G, K, \{\alpha\}, \{\beta\}, \lambda)$
for $k \leftarrow 1$ **to** K **do**
  for $i \leftarrow 1$ **to** g **do**
    ▷ *candidate nodeset w.r.t. the factor* $\mathbf{U}_i(\cdot, k)$.
    $\hat{\mathcal{S}}_i = \{v \mid \mathbf{U}_i(v,k) \geq \frac{1}{\sqrt{n_i}}, \forall\, v \in [|\mathcal{V}|]\}$
  $\mathcal{V}' \leftarrow$ DSDSOLVER$(G, \{\hat{\mathcal{S}}_1, \ldots, \hat{\mathcal{S}}_g\}, \rho)$
  $\mathcal{V}'' \leftarrow$ EXPANDER$(G, \mathcal{V}', \rho)$
  if $\rho_G(\mathcal{V}'') > \rho_G(\mathcal{V}^*)$ **then**
    $\mathcal{V}^* \leftarrow \mathcal{V}''$
return $\mathcal{V}^*$

First, INDUEN utilizes CONF to obtain decomposition factors $\{\mathbf{U}_1, \ldots, \mathbf{U}_g\}$. Then, for each rank-1 component $\mathbf{U}_i(:,k)$, it finds a candidate $\hat{\mathcal{S}}_i$ and detects the densest subgraph nodeset $\mathcal{V}'$ with a DSDSOLVER based on the candidate solution $\hat{\mathcal{S}}^k = \{\hat{\mathcal{S}}_1, \ldots, \hat{\mathcal{S}}_g\}$; and a refined nodeset $\mathcal{V}''$ is obtained after exerting expander-boosting with EXPANDER. Finally, it returns the nodeset achieving the highest density score as the solution.

C. Optimization Rationale

We reveal the unified framework behind INDUEN for detecting interrelated densest subgraphs by group optimizations: the coupled decomposition actually finds the approximation solution for IDSD in the *continuous space*, which guarantees the non-emptiness of $\mathcal{S}^*$; DSDSOLVER and EXPANDER refine the above rough results with *discrete optimization*.

Theorem 1 (Consistent optimization): Using the *edge density* as density metric, the three steps of INDUEN maximize the subgraph density consistently to solve the IDSD problem.

Proof 1: Given an undirected graph $G = (\mathcal{V}, \mathcal{E})$ with the adjacency matrix $\mathbf{A}$ and a nonempty subset $\mathcal{S} \subseteq \mathcal{V}$, let the vector $\boldsymbol{x} \in \{0,1\}^{|\mathcal{V}|}$ with $\boldsymbol{x}_i = 1$ if $i \in \mathcal{S}$, otherwise $\boldsymbol{x}_i = 0$, then the edge density of the subgraph $G(\mathcal{S})$ is $\rho_G(\mathcal{S}) = \frac{1}{2} \cdot \frac{\boldsymbol{x}'\mathbf{A}\boldsymbol{x}}{\boldsymbol{x}'\boldsymbol{x}} = \frac{\boldsymbol{y}'\mathbf{A}\boldsymbol{y}}{2}$ with $\boldsymbol{y} = \boldsymbol{x}/\sqrt{|\mathcal{S}|}$ and $||\boldsymbol{y}||_2 = 1$.

In the general case, for any non-negative $\boldsymbol{x} \in \mathbb{R}_+^{|\mathcal{V}|}$, the densest subgraph detection problem in G can be formulated as $\max_{\boldsymbol{x} \in \mathbb{R}_+^{|\mathcal{V}|}, ||\boldsymbol{x}||_2 = 1} \boldsymbol{x}'\mathbf{A}\boldsymbol{x}$, by optimizing $\boldsymbol{x}$, and it will lead to an *imperfect solution*, i.e., $\boldsymbol{x}_i \neq \frac{1}{\sqrt{|\mathcal{S}^*|}}$ or 0, in the *continuous space*. Furthermore, we can search for obtaining a realistic solution as the final nodeset, which corresponds to the *discrete space* with other optimization techniques, such as local search, linear programming, projection, etc.

In terms of the coupled factorization objective function in (4) for a multilayer network G, that is,

$$\begin{aligned}\min_{\{\mathbf{U}_i, \mathbf{\Lambda}_i, \mathbf{\Sigma}_{i,j}\}} & \sum_{i=1}^{g} \|\mathbf{A}_i - \mathbf{U}_i \mathbf{\Lambda}_i \mathbf{U}_i'\|_F^2 \\ & + \sum_{i=1}^{g}\sum_{j>i}^{g} \beta_{i,j} \left\|\mathbf{C}_{i,j} - \mathbf{U}_i \mathbf{\Sigma}_{i,j} \mathbf{U}_j'\right\|_F^2 \\ \text{s.t. } & \mathbf{U}_i'\mathbf{U}_i = \mathbf{I}, \mathbf{U}_i \geq \mathbf{0}, \mathbf{\Lambda}_i > \mathbf{0}, \mathbf{\Sigma}_{i,j} > \mathbf{0}, \\ & \beta_{i,j} \geq 0, \forall i, j \in [g],\end{aligned}$$

which can be also reformulated as

$$\begin{aligned}\max_{\{\mathbf{U}_i, \mathbf{\Lambda}_i, \mathbf{\Sigma}_{i,j}\}} & \sum_{i=1}^{g}\sum_{k=1}^{K} \mathbf{\Lambda}_i(k,k) \mathbf{tr}\left(\mathbf{U}_i'(k,:)\mathbf{A}_i\mathbf{U}_i(:,k)\right) \\ & + \sum_{i=1}^{g}\sum_{j>i}^{g} \beta_{i,j} \sum_{k=1}^{K} \mathbf{\Sigma}_{i,j}(k,k) \mathbf{tr}\left(\mathbf{U}_i'(k,:)\right. \\ & \left.\mathbf{C}_{i,j}\mathbf{U}_j(:,k)\right) \\ & - \sum_{j=1}^{g}\left(\sum_{k=1}^{K} \mathbf{\Lambda}_i^2(k,k) + \sum_{j>i}^{g}\sum_{k=1}^{K} \beta_{i,j} \mathbf{\Sigma}_{i,j}^2(k,k)\right).\end{aligned}$$

In general, the first term $\mathbf{tr}(\mathbf{U}_i'(k,:)\mathbf{A}_i\mathbf{U}_i(:,k))$ aims at finding the densest subgraph in the i-th within layer based on the k-th factor $\mathbf{U}_i(:,k)$; the second term $\mathbf{tr}(\mathbf{U}_i'(k,:)\mathbf{C}_{i,j}\mathbf{U}_j(:,k))$ aims at finding the densest subgraph in a bipartite graph (cross-layer dependency) w.r.t $\mathbf{C}_{i,j}$ based on the k-th factors $\mathbf{U}_i(:,k)$ and $\mathbf{U}_j(:,k)$. Compared with the single graph case, all the objectives here are strongly coupled with each other.

Therefore, the decomposed factors from CONF compose K solutions in the *continuous space*, as approximations for $\mathcal{S}^*$ for IDSD problem, it ensures the interrelation between nodeset of different layers. Constructing candidate nodeset with thresholds leads to rough solutions in the *discrete space* and guarantees the non-empty for all $\hat{\mathcal{S}}_i$; DSDSOLVER and EXPANDER,

as discrete optimizations, refine the results consequently by improving the density by updating neighbour nodes until convergence.

Even with other density metrics, decomposed factors from CoNF also provide sets of good approximate solutions since those dense subgraphs under different density metrics consist of those nodes with a high score in factors w.h.p.

D. Algorithm Analysis

Here, we analyse the proposed algorithms in terms of their effectiveness as well as efficiency. All proofs are given in the supplementary material.

Theorem 2: The fixed points of (7) and (8) satisfy the KKT condition for their optimality.

The convergence of the CoNF is given by Lemma 1.

Lemma 1: The objective functions in (4) and (5) decrease monotonically under the updating rules in (7), and (8).

Moreover, since $\beta_{i,j}$ is the coefficient of the term $\|\mathbf{C}_{i,j} - \mathbf{U}_i\mathbf{\Sigma}_{i,j}\mathbf{U}_j'\|_F^2$ in (4) while $\gamma_{i,j}$ is the coefficient of $||\mathbf{C}_{i,j}||_{1,1}$ in (1), we can initialize $\beta_{i,j} = \gamma_{i,j}^2$ for CoNF or set $\gamma_{i,j} = \sqrt{\beta_{i,j}}$ for the density definition to keep it simple.

According to Theorem 2 and Lemma 1, Algorithm 1 converges to a local minimal solution for each decomposed factor.

In terms of efficiency, we analyze the time complexity and space complexity of the proposed INDUEN algorithm as the following Theorems 3 and 4.

Given a g-layered network, $G = (\mathcal{V}, \mathcal{A}, \mathcal{C}, \mathbf{G})$, let $\texttt{NNZ} = \sum_{\mathbf{A}_i \in \mathcal{A}} nnz(\mathbf{A}_i) + \sum_{\mathbf{C}_{i,j} \in \mathcal{C}} nnz(\mathbf{C}_{i,j})$, K be the factorization rank, and $\hat{\mathcal{S}} = \text{argmax}_{k \in [K], \hat{\mathcal{S}}^k = \{\hat{\mathcal{S}}_1, \ldots, \hat{\mathcal{S}}_g\}} |\hat{\mathcal{S}}^k|$, and use GREEDYSOLVER as DSDSOLVER.

Theorem 3 (Time Complexity of: **INDUEN** *)* Let T be the number of iterations in CoNF, then the time complexity of INDUEN is $O(T \cdot (\texttt{NNZ} \cdot K + |\mathcal{V}| \cdot K^2) + |\hat{\mathcal{S}}| \log |\hat{\mathcal{S}}|)$.

Theorem 4 (Space Complexity of: **INDUEN** *)* The space complexity of INDUEN is $O(\texttt{NNZ} + K \cdot (|\mathcal{V}| + g + |\mathcal{C}|))$.

VI. EXPERIMENT

We design experiments to answer the following questions:

- *Effectiveness for IDSD:* How accurately does INDUEN detect interrelated densest subgraphs of different settings? Does it outperform the state-of-the-art methods?
- *Real Pattern Discovery:* Does INDUEN discover interrelated densest subgraphs in real-world multilayer networks? And what patterns can be recognized?
- *Apply to Anomaly Detection:* how is the performance of INDUEN for the complex anomaly detection tasks?
- *Ablation study:* How do the components and parameters influence the detection results of INDUEN?
- *Scalability:* Is INDUEN scalable with the size of the input multilayer networks?

We provide detailed information about the dataset and different settings, additional experimental results and analysis, and more case studies in the supplementary.

TABLE II
STATISTICS OF REAL-WORLD DATASETS

Dataset	AMINER	BIO	INFRA-3	DBLP
# of layers (g)	3	3	3	3
# of nodes ($\|\mathcal{V}\|$)	125,344	35,631	15,126	5,015
# of within-layer links	214,134	252,749	29,861	10,039
# of cross-layer links	188,844	75,456	28,023,500	8,627

A. Datasets

Real-world: We employ various real-world datasets from different domains as [26], including AMINER and DBLP for academia (author, paper, venue) [79], BIO for biology (chemical, gene, and disease) [80], and INFRA-3 [6] for infrastructure (airport, autonomous system, power grid). The statistics of the datasets are summarised in Table II.

Synthetic: We build some networks with network generation models and test whether algorithms can detect the injected dense subgraphs in each layer of a multilayer network. The 3-layer networks have a number of nodes $(1800, 2400, 3000)$. *Erdős-Rényi (ER) network* is generated with Erdős-Rényi model [81] where within-layer nodes are connected with probability p. *Scale-Free (SF) network* is generated with Barabási-Albert (BA) preferential attachment model [82] for within-layer links with the number of edges controlled by parameters m.

We randomly select 10% of nodes in each layer as the ground-truth nodeset for injected interrelated densest subgraph and generate links of within-layer and cross-layers with probabilities of p_{gt} and 0.05, respectively; randomly create cross-layer dependencies of the remaining nodes between layers with a probability of 0.001.

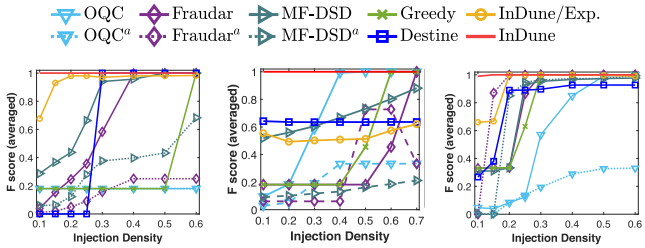
B. Experimental Setup

Baselines: We compare INDUEN with the following methods, including Greedy [13] (GREEDYSOLVER), OQC [14], which detects dense subgraphs by optimizing the edge-surplus density framework via greedy, COREAPP [83] for efficient dense subgraph discovery, NMF [84] for maximal clique detection, FRAUDAR [11] with column-weighting as g_{log}, FlowScope [9] detecting dense subgraphs in the multipartite graphs, KPCORE [21] searching cohesive communities incorporating meta-path over large heterogeneous information networks (HINs), and DESTINE [26]. We transform the multilayer network to a HIN and carefully design the meta-paths for each network.[3] All are implemented in Python, but KPCORE in Java. Except for FlowScope, KPCORE, and DESTINE, we apply the others, designed only for a single graph, to the multilayer networks by Layer-By-Layer detection or Aggregation detection.

Metrics: F-score is used if the ground truth is available; the size and density of the detected subgraphs are given.

Machine: All the experiments are performed on a machine with 2.1 GHz Intel(R) Skylake CPU and 64 GB of RAM.

[3]The meta-path information: 'A-P-V-P-A' for AMINER and DBLP where 'A', 'P', 'V' denote Author, Paper, and Venue resp., 'C-G-D' for BIO denotes 'Chemical', 'Gene', and 'Disease' resp., 'P-AS-A' for INFRA-3 denoting 'Power', 'Autonomous System', and 'Airport', resp.



(a) Synthetic ER Network ($\gamma = 50$). (b) Synthetic SF Network ($\gamma = 20$). (c) sub-BIO Network ($\gamma = 5$).

Fig. 3. Performance of detecting the injected interrelated dense subgraphs in the ER, SF, and sub-BIO networks. Here shows the averaged *F score* of all three within layers. The method marked with the superscript '^{a}' means the result with the aggregated graph as input; 'INDUEN/Exp.' denotes the variant of INDUEN by removing the EXPANDER step.

In all experiments, we set $K = 10$, $\lambda = 10^{-10}$, and set βs as (6) for INDUEN. By default, we adopt the joint density in (1) with all $\gamma_{i,j}$ being the same value as γ, and choose GREEDYSOLVER as DSDSOLVER with Aggregation-Detection strategy; $\gamma = 0$ is used when adopting Layer-By-Layer Detection strategy. For other baselines, the form of (2) is used if taking the aggregated graph as their input.

C. Effectiveness for IDSD

1) Injection Detection: We test the performance of different methods for detecting the injected interrelated densest subgraphs with different injection densities in all multilayer networks.

Networks: We use ER network by setting $p = 0.05$ to generate links of all within layers, and SF network with the parameters m as $(40, 60, 80)$. We also sample a small sub-BIO over real BIO network for easy-control, which consists of randomly selected 3,000 nodes from each layer while keeping all links associated with them, and randomly select 300 nodes from each layer as the target to inject an interrelated dense subgraph where the cross-layer links of those nodes are generated and reconnected with probability 0.05 and the density (link probability) of these targets is p_{gt}.

We inject interrelated densest subgraph with different p_{gt} to control the density of the ground truth and randomly link the target nodes, then apply detection methods to these synthetic cases. $\gamma = 50, 20, 5$ for ER, SF, and sub-BIO networks, respectively.

Fig. 3 illustrates the average detection performance for all layers. We can see that INDUEN consistently achieves the best performance with $F = 1.0$, and it has a lower detection density of the interrelated densest subgraphs, which means it can accurately find all the nodes forming the dense subgraphs even if their densities are not so significant, e.g., no more than 0.2. As a comparison, the results of INDUEN/Exp. verify that EXPANDER can do help to improve the detection results, especially when facing low injection densities. The performance of most baselines becomes better with the increase in injection density.

For the *ER network* in Fig. 3(a), DESTINE can only detect the target with a density higher than 0.25; MF-DSD achieves little better results than FRAUDAR; OQC's performance is independent of the injection density; the results of FRAUDAR and MF-DSD for the aggregation graph are worse than adopting Layer-By-Layer detection strategy.

For the *SF network* in Fig. 3(b), INDUEN achieves complete detection even with ≤ 0.1 injection density; INDUEN/Exp. has imperfect detection (nearly similar $F \approx 0.6$) for each layer, which justifies that EXPANDER boosts the performance by improving the recall. The behaviour of FRAUDAR ^{a} is interesting in that its performance drops when the injection density is larger than 0.6, and it also outperforms FRAUDAR when the injection density falls in $(0.4, 0.6]$; Greedy has better performance only when the density exceeds 0.4; the detection result of DESTINE for each layer hardly changes with the increase of the injection density, leading to a stable F score.

For the *sub-BIO network* in Fig. 3(c), FRAUDAR ^{a} achieves good performance and can detect subgraphs with a lower density than FRAUDAR; hence, FRAUDAR is sensitive to the type of base networks. MF-DSD ^{a} is also better than MF-DSD but slightly worse than FRAUDAR ^{a}; the opposite is true for OQC, which is superior to OQC ^{a}; their results are better than those in ER network. Overall, DESTINE does not have a significant advantage over others in both cases.

In conclusion, INDUEN can deal with and be insensitive to different types of networks, while other baselines have inferior performance and are influenced by the underlying graphs. Besides, we also compared the performance for detecting the *injected cliques* in the ER and SF networks following the same setting as DESTINE, and it also turns out that INDUEN significantly outperforms all other baselines. See the details and more results in the supplementary.

2) IDSD for Real Networks: For the detected multilayer subgraph G, we denote the volume density as $\rho_i = \frac{||\mathbf{A}_i||_{1,1}}{n_i \cdot (n_i - 1)}$ for the ith within layer, and as $\rho_{i,j} = \frac{||\mathbf{C}_{i,j}||_{1,1}}{n_i \cdot n_j}$ for the $\mathbf{C}_{i,j}$ cross-layer dependencies. Without ground truth, we define different density metrics to measure the detected subgraph G concerning the properties of interrelated densest subgraphs.

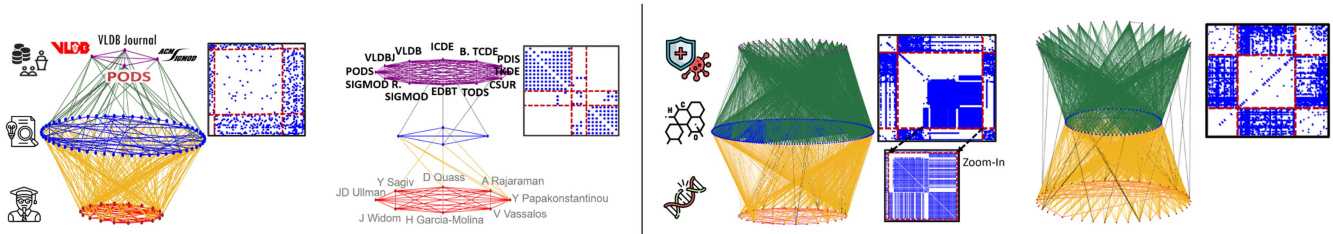
$$d_{\min}(G) = \min\left\{\{\rho_i\} \cup \{\rho_{i,j}\} \mid \forall i, j \in [g]\right\},$$

$$d_{geo}(G) = \left(\prod_{i=1}^{g} \rho_i \cdot \prod_{j>i}^{g} \rho_{i,j}\right)^{1/N},$$

$$d_{ari}(G) = \frac{\sum_{i=1}^{g} \rho_i + \sum_{j>i}^{g} \rho_{i,j}}{N}.$$

Analysis: Different formulations of the above metrics correspond to different ways of working and contributions of within- and cross-layers. For min-volume density $d_{\min}$, it focuses on *the volume-based density* and reports the minimum by considering the within- and cross-layer connections; the unrelated subgraphs will result in $d_{\min} = 0$ and interrelated dense subgraphs with higher density lead to a larger $d_{\min}$. For geometric mean density d_{geo}, *non-correlation (even higher density) or any empty set results in* $d_{geo} = 0$*; the higher value of* d_{geo}*, the better the result.* For arithmetic mean density d_{ari}, it counts the mean of layer-wise densities (within- and cross-layers contribute equally); a large d_{ari} indicates a large overall density of subgraphs but without a guarantee for the density of each layer. In contrast, a larger value of d_{geo} ensures higher density of subgraphs in each layer.

Table III shows the results of different methods over INFRA-3. It reports the size of the nodeset in each layer (i.e., $|\mathcal{S}_i|$)



(a) Interrelated dense subgraph ($\gamma = 10$). (b) **Near free** dense subgraph. (c) Interrelated dense subgraph ($\gamma = 5$). (d) Corss-layer dependencies ($\gamma = 25$).

Fig. 4. Detected real interrelated densest subgraph patterns in the DBLP (Left) and BIO networks (Right), respectively. (a) shows the detected interrelated dense subgraph from INDUEN. (b) gives an example of a *near-free* dense subgraph detected of DESTINE (i.e., with Layer-By-Layer detection strategy), which could be deemed as camouflaged. (c)-(d) are the results of INDUEN with different γs in joint density metric of (1).

TABLE III
COMPARISON OF THE QUALITY OF DETECTION ON INFRA-3

Method	Size ($\lvert\mathcal{S}_i\rvert$)			Density Metrics		
	AP	AS	POWER	d_{min}	d_{geo}	d_{ari}
OQC	82	13	51	0.00	0.00	0.443
OQC^a	1,204	2,083	6,578	6.82E-4	0.0456	0.502
FRAUDAR	123	13	119	0.00	0.00	0.354
FRAUDAR^a	1,204	2,083	7,000	6.20E-4	0.0448	0.502
MF-DSD	202	73	578	0.00	0.00	0.215
MF-DSD^a	1,204	2,083	6,994	4.13E-4	0.0419	0.502
Greedy	101	16	68	0.00	0.00	0.388
COREAPP	44	12	40	0.00	0.00	0.520
COREAPP^a	1,204	2,083	4,310	**1.25E-3**	0.0510	0.502
KPCORE	82	0	0	0.00	0.00	0.100
DESTINE	30	44	4	0.00	0.00	0.508
INDUEN	73	1,319	2993	5.15E-04	**0.0867**	**0.608**

The best results are shown in bold.

and defined density metrics. We can see that $d_{\min} = d_{geo} = 0$ but $d_{ari} > 0$ for OQC, FRAUDAR, MF-DSD, Greedy, COREAPP, and KPCORE, since they detect at least one *free dense subgraph* for some layers, i.e., $\exists\, C^{\mathcal{S}}_{i,j} = \mathbf{0}$; COREAPP ^{a} attains the best score considering the $d_{\min}$. Even with the same elaborated meta-path as the original work [21], KPCORE can only identify some k-core (cohesive subgraph) in some single layer of the HIN, which is also consistent with the results in the original paper and fails to address our problem. DESTINE and those taking in the Aggregated graph as input get non-zero metric values. Our INDUEN with $\gamma = 0.001$ achieves the best scores (d_{geo} and d_{ari}) for the interrelated densest subgraph; it has the same result as FRAUDAR ^{a} if $\gamma = 1$ (ignored in the table).

Table IV demonstrates the results over AMINER, DBLP, and BIO datasets; the basic conclusion is similar to Table III. INDUEN achieves the best $d_{\min}$ and d_{geo} for three networks, and its d_{ari} is also the best for BIO. Although some baselines achieve the best d_{ari}, the whole detected subgraphs are still uncorrelated (at least one free dense subgraph) making $d_{geo} = 0$. Moreover, $d_{\min} = d_{geo} = 0$ but $d_{ari} > 0$ for OQC, FRAUDAR, Greedy, COREAPP, and KPCORE for all networks; MF-DSD can detect some interrelated subgraphs, but it is not the best. Using the Aggregation graph for OQC and FRAUDAR even deteriorates their performance, resulting from domination by some layers, but it also makes MF-DSD detect larger-size subgraphs for AMINER and DBLP.

Therefore, INDUEN, under the proposed *joint density metric*, achieves the consistent best performance over different real datasets, and d_{ari} cannot well-suit to measure the interrelated dense subgraph settings. Note that γs of those networks have a great difference and depend on the density and structure of the network. The optimal result of INDUEN is robust to γ, as verified in the following section.

D. Real-World Pattern Discovery

In this section, we show that INDUEN finds some interesting patterns corresponding to the interrelated densest subgraphs in real-world networks. The γ is given in the caption.

Academic Networks: As shown in Fig. 4(a), the interrelated densest subgraph denotes 18 closely-cooperated authors published 56 papers on Database field (*VLDB*, *VLDB Journal*, *PODS*, and *SIGMOD*). The within-layer connections denote co-authorship & venues-citation and the cross-layer links are intensively dense, while the citations between papers are relatively sparse due to data limitations. Thus, we can infer with high confidence that the research interests of those authors should also focus on the *Database Field*, which is confirmed by manual verification, i.e., these authors include *A Silberschatz*, *Yuri Breitbart*, and *Arun N. Swami*, etc. Also, we obtain the same results for any $\gamma \in [10, 1e8]$.

As teaching material by a negative example, Fig. 4(b) shows the result of INDUEN with Layer-By-Layer detection (as DESTINE). Those authors were from the Stanford University InfoLab and formed a small clique for cooperation; *these venues had strong mutual references while being more diverse*, though related to Database and Data Mining. There were *very limited cross-layer dependencies*, and we could not figure out the strong cooperation relationship among authors just based on the limited co-authored papers. These dense subgraphs in each within-layer are almost free and independent, which can be deemed as camouflage similar to Section VI-E1.

Bio-Networks Fig. 4(c)–(d) shows the detected results w.r.t. different γs in joint density as (1), which controls the final patterns and depends on user's preference. A small γ tends to find the interrelated densese subgraphs with higher within-layer density, while a large γ will emphasize the importance of the

TABLE IV
COMPARISON OF THE QUALITY OF DETECTION ON REAL-WORLD NETWORKS

Method	AMINER ($\gamma = 5$)						DBLP ($\gamma = 5$)						BIO ($\gamma = 3$)					
	Size ($\lvert\mathcal{S}_i\rvert$)			Density Metrics			Size ($\lvert\mathcal{S}_i\rvert$)			Density Metrics			Size ($\lvert\mathcal{S}_i\rvert$)			Density Metrics		
	Author	Paper	Venue	d_{min}	d_{geo}	d_{ari}	Author	Paper	Venue	d_{min}	d_{geo}	d_{ari}	Chem	Gene	DZ	d_{min}	d_{geo}	d_{ari}
OQC	28	10	48	0	0	0.453	15	10	28	0	0	**0.524**	91	73	119	0	0	0.359
OQC^a	28	0	0	0	0	0.200	15	1	0	0	0	0.200	91	1	2	0	0	0.333
FRAUDAR	28	10	79	0	0	0.410	15	250	47	0	0	0.286	203	1,323	124	0	0	0.193
FRAUDAR^a	28	0	0	0	0	0.200	15	1	0	0	0	0.200	204	2	2	0	0	0.319
MF-DSD	1,686	509	109	0.0003	0.009	0.051	18	253	48	0	0	0.227	147	2,209	159	0	0.004	0.189
MF-DSD^a	2,300	524	184	0.0012	0.007	0.025	145	403	82	0.012	0.030	0.050	160	3	6	0.067	0.267	0.348
Greedy	28	10	68	0	0	0.422	15	221	30	0	0	0.337	91	795	117	0	0	0.296
COREAPP	28	10	40	0	0	0.468	15	79	16	0	0	0.427	91	273	81	0	0	0.351
COREAPP^a	28	0	0	0	0	0.200	15	1	0	0	0	0.200	91	1	2	0	0	0.333
KPCORE	40	0	0	0	0	0.097	15	0	0	0	0	0.200	91	0	0	0	0	0.167
DESTINE	14	5	0	0	0	**0.593**	127	163	5	0.008	0.070	0.164	249	118	142	3E-05	0.011	0.210
INDUEN	122	74	5	**0.0292**	**0.493**	0.285	22	82	3	**0.064**	**0.219**	0.343	138	2	5	**0.100**	**0.424**	**0.549**

The best results are shown in bold and it indicates the optimal/best result occurs for the first time.

cross-layer dependencies more, leading to some dense components very close to multipartite graphs.

The interrelated densest subgraph in Fig. 4(c) has high within-layer densities and cross-layer dependencies. The pattern of chemical interactions is also interesting, since some of them are densely interconnected while others are densely connected to other layers. It reveals that different chemical elements act on different diseases, while genes with high similarity will be influenced by many different chemical elements. Also, the pattern in Fig. 4(d) shows sparse and loose internal connections but dense cross-layer dependencies, which means that some unrelated chemical elements work together to influence dissimilar gene expression and cure some diseases; while there are few connections between these genes and diseases.

E. Apply to Anomaly Detection

Consider real applications in anomaly detection for dense subgraph pattern [11], [68], [69], to evade detection, smart fraudsters will camouflage themselves by trying to 'look normal' via adding links to non-target objects [11]. We apply INDUEN to the challenging task and verify its robustness under camouflage for the multilayer and multipartite networks.

1) Multilayer Networks: Here, we deem those *freely dense subgraphs* in each within layer graph as *camouflages*, i.e., their densities (connection probability) of within-layer may be not less than those of corresponding target interrelated dense subgraph (true anomalies), while the densities of cross-layers are less than the densities of true anomalies. Using the same setting as Section VI-C1 for the ER network, we randomly select the other 10% of nodes (no overlapping with the target injection) in each layer as the nodeset for camouflaging, generate the within-layer random connection with link-probability p_{cam} while keeping their original cross-layers' links as is. Thus, the target injected anomalies correspond to an interrelated densest subgraph which has higher cross-layer dependencies (similar or lower within-layer density) than the camouflages.

In Table V, we show the detection performance with the change of p_{gt} of the injected interrelated dense subgraphs under $p_{cam} = 1.0$. INDUEN with $\gamma = 15$ can perfectly detect the injected anomalies in each layer even for a lower density; and EXPANDER works prominently at higher injection densities.

TABLE V
F SCORES ON *ER NETWORK* WITH DIFFERENT INJECTION DENSITIES UNDER CAMOUFLAGE ($p_{cam} = 1.0$). $\gamma = 15$ FOR INDUEN

Method	k-th layer	Injection density p_{gt}					
		0.10	0.20	0.30	0.40	0.50	0.60
MF-DSD^a	1,2	0.00	0.00	0.00	0.00	0.00	0.00
	3	0.00	0.00	0.00	0.00	0.00	0.12
DESTINE	1,2,3	0.00	0.00	0.00	0.67	0.67	0.67
INDUEN / Exp.	1	**1.00**	**1.00**	0.72	0.67	0.50	0.60
	2	**1.00**	0.94	0.78	0.70	0.91	0.66
	3	**1.00**	0.99	0.77	0.73	0.69	0.72
INDUEN	1,2,3	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**

* Here, OQC,OQC^a, FRAUDAR, FRAUDAR^a, MF-DSD and Greedy are ignored since their detection results are all 0.00.
The best results are shown in bold.

However, DESTINE cannot detect the target until $p_{gt} \geq 0.4$ and fails to distinguish between the camouflages and targets, which results in a low precision = 0.5 and F-score = 0.67. Only MF-DSD ^{a} detects few targets in the 3rd layer when $p_{gt} = 0.6$ and other baselines do not find anything. The reason for the different performances of MF-DSD for different layers is that the larger the within-layer graph, the higher the average degree of the targets and the easier to be detected, when p_{gt}s are the same. Compared to Fig. 3(a), INDUEN works robustly and efficiently and able to resist strong camouflage, while that is not true for DESTINE.

2) Multipartite Graphs: We build a 3-partite graph based on the BIO network, named as BIO-MPG, by extracting the chemical-gene association, gene-gene similarity, and gene-disease association as it is; and non-rigorously use gene-gene similarity as one of the partite. To construct the target dense subgraph as ground truth, we randomly select 300 nodes from each layer of *chemical*, *gene*, and *disease*, and create random connections between them by controlling the injection density. We also randomly select another 500 different nodes from the *gene* set and simply reset the similarity between them as 1.0, which forms a 500×500 clique in the 2nd partite serving as strong camouflage. We report F scores of these three nodesets of each method's detection to measure their performance.

As shown in Fig. 5, INDUEN consistently achieves the best performance for all partites and keeps stable. FRAUDAR can well detect the anomalies in the 1st and 3rd partites, which follows our expectations since it works in the Layer-By-Layer

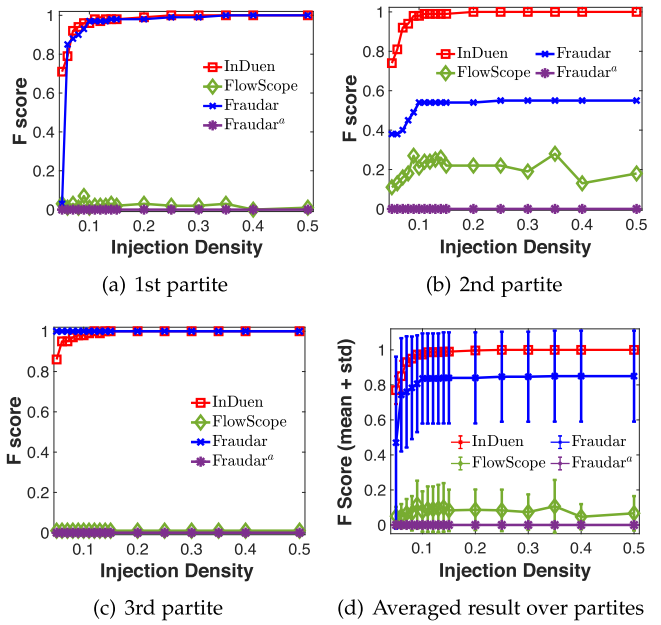


Fig. 5. Performance of detecting the injected anomalies under camouflage settings over a multipartite graph, BIO-MPG. (a)-(c) show the result for each partite, and (d) corresponds to the average result over all partites.

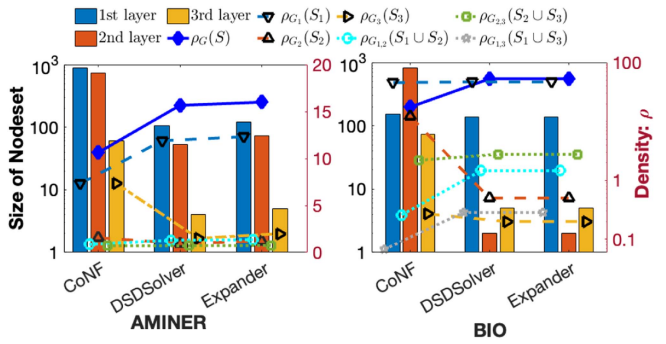


Fig. 6. The size of nodeset and density (joint density $\rho_G(S)$ and average degree density of each layer) after each step (CoNF, DSDSolver, and Expander) in InDuen for AMINER (L) and BIO (R) networks in Table IV.

way and there is no camouflage in these two partites; while its performance drops in the 2nd partite (with recall $= 1.0$ but precision $= \frac{3}{8}$)) due to the influence of camouflage. Meanwhile, Fraudar ^{a} fails to detect any target since the aggregated graph is dominated by camouflage. In addition, although specifically designed for anomaly detection in multipartite graphs, FlowScope, as the state-of-the-art approach to detect densest subgraphs in multipartite graphs, also does not resist the camouflage, it can only find a small portion of inject anomalies, resulting in low F-scores.

Besides, it shows that changing the density of camouflage-dense subgraph has no great influence on the results.

F. Ablation Study

1) Algorithm Steps: Fig. 6 shows the size of nodeset and density of subgraph after CoNF, DSDSolver, and Expander in InDuen, over *AMINER* and *BIO* networks w.r.t. the configuration in Table IV.

TABLE VI
F Scores on *ER Network* With Different Density Hyperparameter γs

Layer	γs							
	0.0	0.6	0.7	0.8	0.9	1	5	10
1	0.0	0.0	0.468	0.468	0.428	0.428	**1.0**	1.0
2	0.0	0.0	0.5	0.5	1.0	1.0	**1.0**	1.0
3	0.0	0.0	1.0	1.0	1.0	1.0	**1.0**	1.0

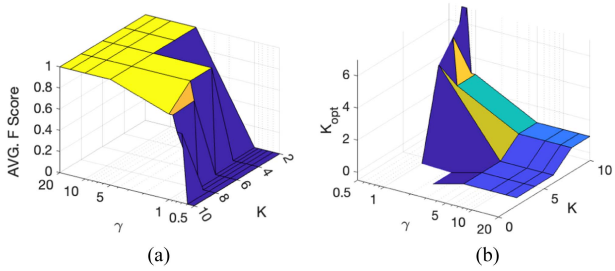


Fig. 7. Performance under vary parameter configurations (γ and K). (a) shows the averaged F score over 3 layers; (b) gives the K_{opt}, i.e., the minimum (optimal) rank K that InDuen figures out the optimal solution.

Compared with the original network, CoNF significantly reduces the size by filtering out a large portion of nodes (note that the log scale), leading to speedup for DSD; DSDSolver gets a very small graph for each layer in greedy; Expander introduces a few neighbours to expand the nodeset (for ANIMER), which will improve the final performance. In terms of density, by considering the contribution of links of both within and cross layers, all those steps of InDuen consistently optimize the joint density and get better results, but this does not guarantee that the average degree density of each layer is continuously increasing (e.g., 3rd layer of AMINER), due to the difference in the optimization objective.

2) Parameter Sensitivity Analysis: We apply InDuen to detect the injected interrelated densest subgraph over the synthetic *ER network*, whose injection density is set as $p_{gt} = 0.50$. We verify the detection robustness of InDuen w.r.t. density hyperparameter γs, and other parameter analysis, including the rank K, βs, and λ. To simplify the settings, we set $\gamma_{i,j} = \gamma, \beta_{i,j} = \beta$.

Table VI reports the F scores of InDuen for different γs with $K = 10$. It can perfectly detect the targets with $F = 1.0$ for all layers when $\gamma \geq 5.0$; and it might detects all targets in some layer (3rd or 2nd) for a lower γ but not for all layers, e.g., $\gamma \in [0.7, 1.0]$. Therefore, the optimal γ vary for different layers. In addition, the F score of non-perfect detection and value of γ depend on the randomness from the factor initialization in Algorithm 1 and the dense subgraphs injection; a lager γ guarantees a good result, e.g., $\gamma \geq 5$.

Fig. 7 shows the InDuen performance for different K and γ. The conclusion about averaged F score in Fig. 7(a) is consistent with Table VI and $K = 10$ is enough for perfect detection; K_{opt} becomes smaller for larger γs and $K_{opt} = 0/1$ (the first / second rank) even for $\gamma \geq 5$.

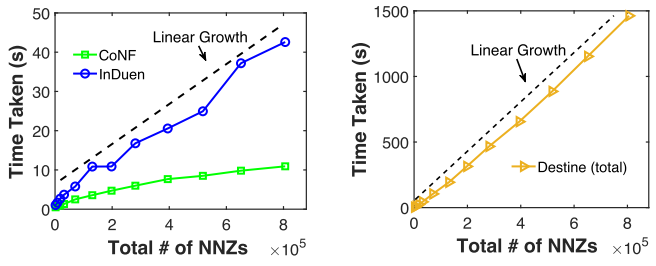


Fig. 8. Linear scalability of INDUEN and CONF to the total number of links in the multilayer network, i.e., $\sum_i \mathbf{A}_i + \sum_{i,j} \mathbf{C}_{i,j}$. In comparison, the total running time of the SOTA method DESTINE is shown in (b).

INDUEN consistently achieves $F = 1.0$ for $\lambda \in [10^{-10}, 10^{-5}]$ and $\beta \in [0, 50]$ (The table is ignored due to the constant results), which is attributed to DSDSOLVER and EXPANDER in Algorithm 3 even different λ and β have some impact on the decomposition factors for adjacency matrices.

G. Scalability

We used the AMINER network here and constructed different-size networks by subsampling the nodes of each within-layer in the proportion of $\{0.1, 0.2, \ldots, 1.0\}$ while keeping the original cross-layer dependencies of the selected nodes.

Fig. 8(a) shows the running time of INDUEN, which is linearly scalable in the total number of links (within- and cross-layer) of the multilayer network. The CONF part takes about $\frac{1}{3}$ of the total. Fig. 8(b) shows the running time of DESTINE with linear scalable. It turns out that our method achieves more than $35\times$ speed up compared with DESTINE.

Moreover, the memory consumption of DESTINE is huge for large networks, and its running time is about 11,000 s for INFRA-3 in Table III while INDUEN only takes about 1,800 s.

VII. CONCLUSION

In this work, we proposed INDUEN algorithm to detect the interrelated densest subgraph in multilayer networks using a flexible joint density measure, This is a configurable framework and employs joint optimization technique by virtue of coupled factorization, greedy search, and local search. Our method can effectively detect interrelated densest subgraphs and consistently outperforms the state-of-the-art baseline baselines, while also revealing various interesting and valuable patterns in real-world networks. Additionally, INDUEN is resilient to the parameters and scales linearly with the size of network, achieving significant speedup.

REFERENCES

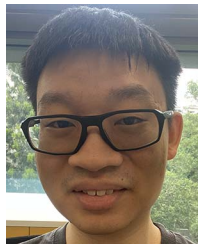
[1] M. Kivelä, A. Arenas, M. Barthelemy, J. P. Gleeson, Y. Moreno, and M. A. Porter, "Multilayer networks," *J. Complex Netw.*, vol. 2, no. 3, pp. 203–271, 2014.
[2] M. De Domenico, "More is different in real-world multilayer networks," *Nature Phys.*, vol. 19, no. 9, pp. 1247–1262, 2023.
[3] Z. Hammoud and F. Kramer, "Multilayer networks: Aspects, implementations, and application in biomedicine," *Big Data Analytics*, vol. 5, no. 1, pp. 1–18, 2020.
[4] A. Halu, M. De Domenico, A. Arenas, and A. Sharma, "The multiplex network of human diseases," *NPJ Syst. Biol. Appl.*, vol. 5, no. 1, pp. 1–12, 2019.
[5] L. K. Gallos, C. Song, S. Havlin, and H. A. Makse, "Scaling theory of transport in complex biological networks," in *Proc. Nat. Acad. Sci. USA*, vol. 104, no. 19, pp. 7746–7751, 2007.
[6] C. Chen, H. Tong, L. Xie, L. Ying, and Q. He, "FASCINATE: Fast cross-layer dependency inference on multi-layered networks," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 765–774.
[7] C. Chen, J. He, N. Bliss, and H. Tong, "Towards optimal connectivity on multi-layered networks," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 10, pp. 2332–2346, Oct. 2017.
[8] J. Xu, T. L. Wickramarathne, and N. V. Chawla, "Representing higher-order dependencies in networks," *Sci. Adv.*, vol. 2, no. 5, 2016, Art. no. e1600028.
[9] X. Li et al., "FlowScope: Spotting money laundering based on graphs," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 4731–4738.
[10] J. Chen and Y. Saad, "Dense subgraph extraction with application to community detection," *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 7, pp. 1216–1230, Jul. 2012.
[11] B. Hooi, H. A. Song, A. Beutel, N. Shah, K. Shin, and C. Faloutsos, "FRAUDAR: Bounding graph fraud in the face of camouflage," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 895–904.
[12] T. Lanciano, A. Miyauchi, A. Fazzone, and F. Bonchi, "A survey on the densest subgraph problem and its variants," 2023, *arXiv:2303.14467*.
[13] M. Charikar, "Greedy approximation algorithms for finding dense components in a graph," in *Proc. Int. Workshop Approximation Algorithms Combinatorial Optim.*, 2000, pp. 84–95.
[14] C. E. Tsourakakis, F. Bonchi, A. Gionis, F. Gullo, and M. A. Tsiarli, "Denser than the densest subgraph: Extracting optimal quasi-cliques with quality guarantees," in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2013, pp. 104–112.
[15] Y. Yang, L. Chu, Y. Zhang, Z. Wang, J. Pei, and E. Chen, "Mining density contrast subgraphs," in *Proc. IEEE 34th Int. Conf. Data Eng.*, 2018, pp. 221–232.
[16] W. Feng, S. Liu, D. Koutra, H. Shen, and X. Cheng, "SpecGreedy: Unified dense subgraph detection," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discov. Databases*, 2020, pp. 181–197.
[17] K. Shin, B. Hooi, and C. Faloutsos, "Fast, accurate, and flexible algorithms for dense subtensor mining," *ACM Trans. Knowl. Discov. Data*, vol. 12, no. 3, pp. 1–30, 2018.
[18] W. Feng, S. Liu, and X. Cheng, "CatchCore: Catching hierarchical dense subtensor," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discov. Databases*, Springer, 2019, pp. 156–172.
[19] D. Barman, S. Bhattacharya, R. Sarkar, and N. Chowdhury, "k-context technique: A method for identifying dense subgraphs in a heterogeneous information network," *IEEE Trans. Computat. Social Syst.*, vol. 6, no. 6, pp. 1190–1205, Dec. 2019.
[20] R. W. Wang, S. X. Wei, and Y. Y. Fred, "Extracting a core structure from heterogeneous information network using h-subnet and meta-path strength," *J. Informetrics*, vol. 15, no. 3, 2021, Art. no. 101173.
[21] Y. Fang, Y. Yang, W. Zhang, X. Lin, and X. Cao, "Effective and efficient community search over large heterogeneous information networks," in *Proc. VLDB Endowment*, vol. 13, no. 6, pp. 854–867, 2020.
[22] Y. Yang, Y. Fang, X. Lin, and W. Zhang, "Effective and efficient truss computation over large heterogeneous information networks," in *Proc. IEEE Int. Conf. Data Eng.*, 2020, pp. 901–912.
[23] E. Galimberti, F. Bonchi, F. Gullo, and T. Lanciano, "Core decomposition in multilayer networks: Theory, algorithms, and applications," *ACM Trans. Knowl. Discov. Data*, vol. 14, no. 1, pp. 1–40, 2020.
[24] R. Zhu, Z. Zou, and J. Li, "Diversified coherent core search on multi-layer graphs," in *Proc. IEEE 34th Int. Conf. Data Eng.*, 2018, pp. 701–712.
[25] C. Chen, J. He, N. Bliss, and H. Tong, "On the connectivity of multi-layered networks: Models, measures and optimal control," in *Proc. IEEE Int. Conf. Data Mining*, 2015, pp. 715–720.
[26] Z. Xu, S. Zhang, Y. Xia, L. Xiong, J. Xu, and H. Tong, "DESTINE: Dense subgraph detection on multi-layered networks," in *Proc. 30th ACM Int. Conf. Inf. Knowl. Manage.*, 2021, pp. 3558–3562.
[27] J. Ni, H. Tong, W. Fan, and X. Zhang, "Flexible and robust multi-network clustering," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2015, pp. 835–844.
[28] X. Liu et al., "Robustness and lethality in multilayer biological molecular networks," *Nature Commun.*, vol. 11, no. 1, pp. 1–12, 2020.

[29] L. Mao, "Modeling triple-diffusions of infectious diseases, information, and preventive behaviors through a metropolitan social network–An agent-based simulation," *Appl. Geogr.*, vol. 50, pp. 31–39, 2014.
[30] V. Jethava, A. Martinsson, C. Bhattacharyya, and D. Dubhashi, "Lovász ϑ function, svms and finding dense subgraphs," *J. Mach. Learn. Res.*, vol. 14, pp. 3495–3536, 2013.
[31] V. E. Lee, N. Ruan, R. Jin, and C. Aggarwal, "A survey of algorithms for dense subgraph discovery," in *Managing and Mining Graph Data*, Berlin, Germany: Springer, 2010, pp. 303–336.
[32] A. V. Goldberg, "Finding a maximum density subgraph," Univ. California Berkeley, USA, Tech. Rep. UCB/CSD-84-171, 1984. [Online]. Available: http://www2.eecs.berkeley.edu/Pubs/TechRpts/1984/5956.html
[33] D. Chakrabarti, S. Papadimitriou, D. S. Modha, and C. Faloutsos, "Fully automatic cross-associations," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2004, pp. 79–88.
[34] S. Khuller and B. Saha, "On finding dense subgraphs," in *Proc. Int. Colloq. Automata Lang. Program.*, Springer, 2009, pp. 597–608.
[35] O. D. Balalau, F. Bonchi, T. H. Chan, F. Gullo, and M. Sozio, "Finding subgraphs with maximum total density and limited overlap," in *Proc. 8th ACM Int. Conf. Web Search Data Mining*, 2015, pp. 379–388.
[36] M. Wu, X. Li, C.-K. Kwoh, and S.-K. Ng, "A core-attachment based method to detect protein complexes in PPI networks," *BMC Bioinf.*, vol. 10, no. 1, pp. 1–16, 2009.
[37] L. Akoglu, H. Tong, and D. Koutra, "Graph based anomaly detection and description: A survey," *Data Mining Knowl. Discov.*, vol. 29, no. 3, pp. 626–688, 2015.
[38] A. Beutel, W. Xu, V. Guruswami, C. Palow, and C. Faloutsos, "CopyCatch: Stopping group attacks by spotting lockstep behavior in social networks," in *Proc. 22nd Int. Conf. World Wide Web*, 2013, pp. 119–130.
[39] Y. Ji, Z. Zhang, X. Tang, J. Shen, X. Zhang, and G. Yang, "Detecting cash-out users via dense subgraphs," in *Proc. 28th ACM SIGKDD Conf. Knowl. Discov. Data Mining*, 2022, pp. 687–697.
[40] K. Shin, B. Hooi, and C. Faloutsos, "M-zoom: Fast dense-block detection in tensors with quality guarantees," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discov. Databases*, Springer, 2016, pp. 264–280.
[41] Q.-H. Duong, H. Ramampiaro, and K. Nørvåg, "Multiple dense subtensor estimation with high density guarantee," in *Proc. IEEE Int. Conf. Data Eng.*, 2020, pp. 637–648.
[42] V. Jethava and N. Beerenwinkel, "Finding dense subgraphs in relational graphs," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discov. Databases*, Springer, 2015, pp. 641–654.
[43] Y. Wu, X. Zhu, L. Li, W. Fan, R. Jin, and X. Zhang, "Mining dual networks: Models, algorithms, and applications," *ACM Trans. Knowl. Discov. Data*, vol. 10, no. 4, pp. 1–37, 2016.
[44] E. Galimberti, F. Bonchi, and F. Gullo, "Core decomposition and densest subgraph in multilayer networks," in *Proc. ACM Int. Conf. Inf. Knowl. Manage.*, 2017, pp. 1807–1816.
[45] T. Chen, F. Bonchi, D. Garcia-Soriano, A. Miyauchi, and C. E. Tsourakakis, "Dense and well-connected subgraph detection in dual networks," in *Proc. SIAM Int. Conf. Data Mining*, SIAM, 2022, pp. 361–369.
[46] A. E. Sariyüce, "Motif-driven dense subgraph discovery in directed and labeled networks," in *Proc. Web Conf.*, 2021, pp. 379–390.
[47] A. Paranjape, A. R. Benson, and J. Leskovec, "Motifs in temporal networks," in *Proc. 10th ACM Int. Conf. Web Search Data Mining*, 2017, pp. 601–610.
[48] S.-V. Sanei-Mehri, A. E. Sariyuce, and S. Tirthapura, "Butterfly counting in bipartite networks," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2018, pp. 2150–2159.
[49] A. Zhou, Y. Wang, and L. Chen, "Butterfly counting on uncertain bipartite graphs," in *Proc. VLDB Endowment*, vol. 15, no. 2, pp. 211–223, 2021.
[50] J. Hu, R. Cheng, K. C.-C. Chang, A. Sankar, Y. Fang, and B. Y. Lam, "Discovering maximal motif cliques in large heterogeneous information networks," in *Proc. IEEE 35th Int. Conf. Data Eng.*, 2019, pp. 746–757.
[51] Y. Fang, K. Wang, X. Lin, and W. Zhang, "Cohesive subgraph search over big heterogeneous information networks: Applications, challenges, and solutions," in *Proc. Int. Conf. Manage. Data*, 2021, pp. 2829–2838.
[52] E. Alsentzer, S. Finlayson, M. Li, and M. Zitnik, "Subgraph neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 8017–8029.
[53] L. Zhao, W. Jin, L. Akoglu, and N. Shah, "From stars to subgraphs: Uplifting any GNN with local structure awareness," in *Proc. 10th Int. Conf. Learn. Representations*, Virtual Event, Apr. 2022, pp. 25–29.
[54] T. Zhao, C. Deng, K. Yu, T. Jiang, D. Wang, and M. Jiang, "Error-bounded graph anomaly loss for GNNs," in *Proc. 29th ACM Int. Conf. Inf. Knowl. Manage.*, 2020, pp. 1873–1882.
[55] X. Bi, A. Qu, and X. Shen, "Multilayer tensor factorization with applications to recommender systems," *Ann. Statist.*, vol. 46, no. 6B, pp. 3308–3333, 2018.
[56] F. M. Almutairi, N. D. Sidiropoulos, and G. Karypis, "Context-aware recommendation-based learning analytics using tensor and coupled matrix factorization," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 5, pp. 729–741, Aug. 2017.
[57] D. Koutra et al., "PNP: Fast path ensemble method for movie design," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2017, pp. 1527–1536.
[58] M. R. D. Araujo, P. M. P. Ribeiro, and C. Faloutsos, "TensorCast: Forecasting with context using coupled tensors," in *Proc. IEEE Int. Conf. Data Mining*, 2017, pp. 71–80.
[59] S. Gao, L. Denoyer, and P. Gallinari, "Temporal link prediction by integrating content and structure information," in *Proc. ACM Int. Conf. Inf. Knowl. Manage.*, 2011, pp. 1169–1174.
[60] E. Gujral and E. E. Papalexakis, "SMACD: Semi-supervised multi-aspect community detection," in *Proc. SIAM Int. Conf. Data Mining*, SIAM, 2018, pp. 702–710.
[61] Y. Liu, L. Zhu, P. Szekely, A. Galstyan, and D. Koutra, "Coupled clustering of time-series and networks," in *Proc. SIAM Int. Conf. Data Mining*, SIAM, 2019, pp. 531–539.
[62] E. Acar, T. G. Kolda, and D. M. Dunlavy, "All-at-once optimization for coupled matrix and tensor factorizations," 2011, *arXiv:1105.3422*.
[63] N. Yokoya, T. Yairi, and A. Iwasaki, "Coupled nonnegative matrix factorization unmixing for hyperspectral and multispectral data fusion," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 2, pp. 528–537, Feb. 2012.
[64] C. Eckart and G. Young, "The approximation of one matrix by another of lower rank," *Psychometrika*, vol. 1, no. 3, pp. 211–218, 1936.
[65] D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2000, pp. 535–541.
[66] Y. Asahiro, K. Iwama, H. Tamaki, and T. Tokuyama, "Greedily finding a dense subgraph," *J. Algorithms*, vol. 34, pp. 203–221, 2000.
[67] D. Boob et al., "Flowless: Extracting densest subgraphs without flow computations," 2019, *arXiv: 1910.07087*.
[68] B. A. Prakash, A. Sridharan, M. Seshadri, S. Machiraju, and C. Faloutsos, "EigenSpokes: Surprising patterns and scalable community chipping in large graphs," in *Proc. Pacific-Asia Conf. Knowl. Discov. Data Mining*, Springer, 2010, pp. 435–448.
[69] N. Shah, A. Beutel, B. Gallagher, and C. Faloutsos, "Spotting suspicious link behavior with fbox: An adversarial perspective," in *Proc. IEEE Int. Conf. Data Mining*, 2014, pp. 959–964.
[70] A. Gionis and C. E. Tsourakakis, "Dense subgraph discovery: KDD 2015 tutorial," in *Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2015, pp. 2313–2314.
[71] C. A. Phillips, K. Wang, J. Bubier, E. J. Baker, E. J. Chesler, and M. A. Langston, "Scalable multipartite subgraph enumeration for integrative analysis of heterogeneous experimental functional genomics data," in *Proc. 6th ACM Conf. Bioinf. Comput. Biol. Health Inform.*, 2015, pp. 626–633.
[72] U. Şimşekli, B. Ermiş, A. T. Cemgil, and E. Acar, "Optimal weight learning for coupled tensor factorization with mixed divergences," in *Proc. 21st Eur. Signal Process. Conf.*, 2013, pp. 1–5.
[73] C. Ding, T. Li, W. Peng, and H. Park, "Orthogonal nonnegative matrix t-factorizations for clustering," in *Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2006, pp. 126–135.
[74] M. Araujo, S. Günnemann, G. Mateos, and C. Faloutsos, "Beyond blocks: Hyperbolic community detection," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discov. Databases*, Springer, 2014, pp. 50–65.
[75] M. Q. Pasta, Z. Jan, A. Sallaberry, and F. Zaidi, "Tunable and growing network generation model with community structures," in *Proc. Int. Conf. Cloud Green Comput.*, 2013, pp. 233–240.
[76] J. Leskovec, D. Chakrabarti, J. Kleinberg, and C. Faloutsos, "Realistic, mathematically tractable graph generation and evolution, using kronecker multiplication," in *Proc. Eur. Conf. Princ. Data Mining Knowl. Discov.*, Springer, 2005, pp. 133–145.
[77] M. van Leeuwen, T. De Bie, E. Spyropoulou, and C. Mesnage, "Subjective interestingness of subgraph patterns," *Mach. Learn.*, vol. 105, pp. 41–75, 2016.
[78] X. He and D. Kempe, "Robust influence maximization," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, New York, NY, USA, 2016, pp. 885–894.

[79] J. Tang, J. Zhang, L. Yao, J.-Z. Li, L. Zhang, and Z. Su, "ArnetMiner: Extraction and mining of academic social networks," in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2008, pp. 990–998.
[80] S. Razick, G. Magklaras, and I. M. Donaldson, "iRefIndex: A consolidated protein interaction database with provenance," *BMC Bioinf.*, vol. 9, no. 1, pp. 1–19, 2008.
[81] P. Erds et al., "On the evolution of random graphs," *Publ. Math. Inst. Hung. Acad. Sci.*, vol. 5, no. 1, pp. 17–60, 1960.
[82] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, 1999.
[83] Y. Fang, K. Yu, R. Cheng, L. V. Lakshmanan, and X. Lin, "Efficient algorithms for densest subgraph discovery," 2019, *arXiv: 1906.00341*.
[84] C. Ding, T. Li, and M. I. Jordan, "Nonnegative matrix factorization for combinatorial optimization: Spectral clustering, graph matching, and clique finding," in *Proc. IEEE Int. Conf. Data Mining*, 2008, pp. 183–192.

Wenjie Feng received the PhD degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences, in 2020. He is a research fellow with the Institute of Data Science, National University of Singapore. His research focuses on large-scale graph mining and learning, trustworthy and safe AI, machine learning and application, and anomaly detection.

Li Wang received the BS degree from Tsinghua University. He is currently working toward the MS degree with the Institute of Computing Technology, Chinese Academy of Sciences. His research interests include multilayered networks, large-scale graph mining and anomaly detection.



Bryan Hooi received the PhD degree in machine learning from Carnegie Mellon University. He an assistant professor with the School of Computing and the Institute of Data Science, National University of Singapore. His research interests include scalable machine learning, deep learning, graph algorithms, anomaly detection, and biomedical applications of AI.

See Kiong Ng received the BS degree in computer science from Carnegie Mellon University (CMU), Pittsburgh, Pennsylvania, the MS degree from the University of Pennsylvania, Philadelphia, Pennsylvania, and the PhD degree in computer science from CMU. He is currently a professor of practice with the School of Computing, National University of Singapore, Singapore, where he is also the director of translational research with the Institute of Data Science. He has authored more than 130 papers on diverse and crossdisciplinary research topics, from bioinformatics to smart cities based on data science and AI.

Shenghua Liu received the PhD degree from the Computer Science and Technology Department, Tsinghua University. He is an associate professor with the Institute of Computing Technology, Chinese Academy of Sciences. He was a visiting scholar with the University of California, Los Angeles and Carnegie Mellon University respectively. His current research interests are designing intelligent and automated algorithms for Big Data mining problems, related to big graphs and series.