User-Guided Clustering in Heterogeneous Information Networks via Motif-Based Comprehensive Transcription

Yu Shi*, Xinwei He*, Naijing Zhang*, Carl Yang, and Jiawei Han

University of Illinois at Urbana-Champaign, Urbana, IL USA
{yushi2, xhe17, nzhang31, jiyang3, hanj}@illinois.edu

Abstract. Heterogeneous information networks (HINs) with rich semantics are ubiquitous in real-world applications. For a given HIN, many reasonable clustering results with distinct semantic meaning can simultaneously exist. User-guided clustering is hence of great practical value for HINs where users provide labels to a small portion of nodes. To cater to a broad spectrum of user guidance evidenced by different expected clustering results, carefully exploiting the signals residing in the data is potentially useful. Meanwhile, as one type of complex networks, HINs often encapsulate higher-order interactions that reflect the interlocked nature among nodes and edges. Network motifs, sometimes referred to as meta-graphs, have been used as tools to capture such higher-order interactions and reveal the many different semantics. We therefore approach the problem of user-guided clustering in HINs with network motifs. In this process, we identify the utility and importance of directly modeling higher-order interactions without collapsing them to pairwise interactions. To achieve this, we comprehensively transcribe the higher-order interaction signals to a series of tensors via motifs and propose the MoCHIN model based on joint non-negative tensor factorization. This approach applies to arbitrarily many, arbitrary forms of HIN motifs. An inference algorithm with speed-up methods is also proposed to tackle the challenge that tensor size grows exponentially as the number of nodes in a motif increases. We validate the effectiveness of the proposed method on two real-world datasets and three tasks, and MoCHIN outperforms all baselines in three evaluation tasks under three different metrics. Additional experiments demonstrated the utility of motifs and the benefit of directly modeling higher-order information especially when user guidance is limited. [1]

Keywords: heterogeneous information networks · user-guided clustering · higher-order interactions · network motifs · non-negative tensor factorization

[*] These authors contributed equally to this work.

[1] The code and the data are available at https://github.com/NoSegfault/MoCHIN.

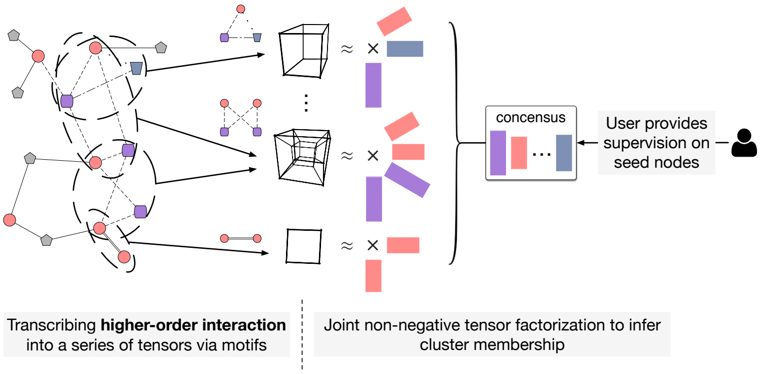

concensus
User provides supervision on seed nodes
Transcribing **higher-order interaction** into a series of tensors via motifs
Joint non-negative tensor factorization to infer cluster membership

Fig. 1: Overview of the proposed method MoCHIN that directly models all nodes in higher-order interactions where each type of nodes in the HIN corresponds to a color and a shape in the figure.

1 Introduction

Heterogeneous information network (HIN) has been shown to be a powerful approach to model linked objects with informative type information [21,23,24,28]. Meanwhile, the formation of complex networks is often partially attributed to the higher-order interactions among objects in real-world scenarios [2,18,35], where the "players" in the interactions are nodes in the network. To reveal such higher-order interactions, researchers have since been using network motifs. Leveraging motifs is shown to be useful in tasks such as clustering [2,36], ranking [37] and representation learning [20]. [2]

Clustering is a traditional and fundamental task in network mining [7]. In the context of an HIN with rich semantics, reasonable clustering results with distinct semantic meaning can simultaneously exist. In this case, personalized clustering with user guidance can be of great practical value [6,10,17,21,29]. Carefully exploiting the fine-grained semantics in HINs via modeling higher-order interaction is a promising direction for such user-guided clustering since it could potentially generate a richer pool of subtle signals to better fit different users' guidance, especially when users cannot provide too much guidance and the supervision is hence weak.

However, it is non-trivial to develop a principled HIN clustering method that exploits signals revealed by motifs as comprehensively as possible. This is because most network clustering algorithms are based on signals derived from the relatedness between each pair of nodes [7]. While a body of research has shown that

[2] Higher-order interaction is sometimes used interchangeably with high-order interaction in the literature, and clustering using signals from higher-order interactions is referred to as higher-order clustering [2,36]. Motifs in the context of HINs are sometimes called the meta-graphs, and we opt for motifs primarily because meta-graphs have been used under a different definition in the study of clustering [27].

it is beneficial for clustering methods to derive features for each node pair using motifs [5,8,10,16,38], this approach essentially collapses a higher-order interaction into pairwise interactions, which is an irreversible process. Such irreversible process is not always desirable as it could cause information loss. For example, consider a motif instance involving three nodes – A, B, and C. After collapsing the higher-order interaction among A, B, and C into pairwise interactions, we are still able to sense the tie between A and C, but such a tie would no longer depend on B – a potentially critical semantic facet of the relationship between A and C. Such subtle information could be critical to distinguishing different user guidance. We will further discuss this point by real-world example in Section 4 and experiments in Section 7. Furthermore, although it is relatively easy to find semantically meaningful HIN motifs [5,8], motifs in HINs can have more complex topology compared to motifs in homogeneous networks do [2,36]. In order to fully unleash the power of HIN motifs and exploit the signals extracted by them, we are motivated to propose a method that applies to arbitrary forms of HIN motifs.

To avoid such information loss with a method applicable to arbitrary forms of motifs, we propose to directly model the higher-order interactions by comprehensively transcribing them into a series of tensors. As such, the complete information of higher-order interactions is preserved. Based on this intuition, we propose the MoCHIN model, short for **Mo**tif-based **C**lustering in **HIN**s, with an overview in Figure 1. MoCHIN first transcribes information revealed by motifs into a series of tensors and then performs clustering by joint non-negative tensor decomposition with an additional mechanism to reflect user guidance.

In this direction, an additional challenge arises from inducing tensor via corresponding motif – the size of the tensor grows exponentially as the number of nodes involved in the motif increases. Fortunately, motif instances are often sparse in real-world networks just as the number of edges is usually significantly smaller than the number of node pairs in a large real-world network. This fact is to be corroborated in Section 3 of the supplementary file. We hence develop an inference algorithm taking advantage of the sparsity of the tensors and the structure of the proposed MoCHIN model.

Lastly, we summarize our contributions as follows: (i) we identify the utility of modeling higher-order interaction without collapsing it into pairwise interactions to avoid losing the rich and subtle information captured by motifs; (ii) we propose the MoCHIN model that captures higher-order interaction via motif-based comprehensive transcription; (iii) we develop an inference algorithm and speed-up methods for MoCHIN; (iv) experiments on two real-world HINs and three tasks demonstrated the effectiveness of the proposed method as well as the utility of the tensor-based modeling approach in user-guided HIN clustering.

2 Related Work

Network motifs and motifs in HINs. Network motifs, or graphlets, are usually used to identify higher-order interactions [2,18,35,36]. One popular research direction on network motifs has centered on efficiently counting motif instances such as triangles and more complex motifs [1, 26]. Applications of motifs have also been found in tasks such as network partition and clustering [2, 12, 36, 39] as well as ranking [37].

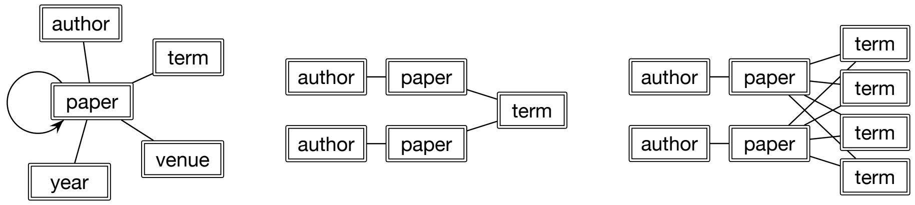
In the context of HINs, network motifs are sometimes referred to as meta-graphs or meta-structures and have been studied recently [5,8,10,15,16,20,33,38]. Many of these works study pairwise relationship such as relevance or similarity [5, 8, 15, 16, 38], and some other address the problem of representation learning [20, 33] and graph classification [34]. Some of these prior works define meta-graphs or meta-structures to be directed acyclic graphs [8, 38], whereas we do not enforce this restriction on the definition of HIN motifs.

Clustering in heterogeneous information networks. As a fundamental data mining problem, clustering has been studied for HINs [13, 21, 22, 28–30]. One line of HIN clustering study leverages the synergetic effect of simultaneously tackling ranking and clustering [22,30]. Clustering on specific types of HINs such as those with additional attributes has also been studied [13]. Wu et al. [32] resort to tensor for HIN clustering. Their solution employs one tensor for one HIN and does not model different semantics implied by different structural patterns.

User guidance brings significantly more potentials to HIN clustering by providing a small portion of seeds [21, 29], which enables users to inject intention of clustering. To reveal the different semantics in an HIN, pioneering works exploit the meta-path, a special case of the motif, and reflect user-guidance by using the corresponding meta-paths [17, 29].

To the best of our knowledge, a recent preprint [3] is the only paper that tackles HIN clustering and applies to arbitrary forms of HIN motifs, which is not specifically designed for the scenario with user guidance. Given an HIN and a motif (*i.e.*, typed-graphlet), this method filter the original adjacent matrix to derive the typed-graphlet adjacency matrix and then perform spectral clustering on the latter matrix. While being able to filter out information irrelevant to the given motif, this method essentially exploits the edge-level pairwise information in the adjacent matrix rather than directly modeling each occurrence of higher-order interaction. Other related works include a meta-graph–guided random walk algorithm [10], which is shown to outperform using only meta-paths. Note that this method cannot distinguish motif AP4TPA from meta-path APTPA, which are to be introduced in Section 4. Sankar et al. [20] propose a convolutional neural network method based on motifs which can potentially be used for user-guided HIN clustering. This approach restricts the motifs of interest to those with a target node, a context node, and auxiliary nodes. Gujral et al. [6] propose a method based on tensor constructed from stacking a set of adjacency matrices, which can successfully reflect user guidance and different semantic aspects. This method essentially leverages features derived for node pairs.

We additionally review the related work on matrix and tensor factorization for clustering in the supplementary file for this paper. These studies are relevant but cannot be directly applied to the scenario of higher-order HIN clustering.



(a) The schema [25].

(b) The APTPA motif, which is also a meta-path.

(c) The AP4TPA motif.

Fig. 2: Examples of schema and motif in the DBLP network.

3 Preliminaries

In this section, we define related concepts and notations.

Definition 1 (Heterogeneous information network and schema [28]). *An information network is a directed graph $G = (\mathcal{V}, \mathcal{E})$ with a node type mapping $\varphi : \mathcal{V} \to \mathcal{T}$ and an edge type mapping $\psi : \mathcal{E} \to \mathcal{R}$. When $|\mathcal{T}| > 1$ or $|\mathcal{R}| > 1$, the network is referred to as a* ***heterogeneous information network (HIN)****. The* ***schema*** *of an HIN is an abstraction of the meta-information of the node types and edge types of the given HIN.*

As an example, Figure 2a illustrates the schema of the DBLP network to be used in Section 7. We denote all nodes with the same type $t \in \mathcal{T}$ by $\mathcal{V}_t$.

Definition 2 (HIN motif and HIN motif instance). *In an HIN $G = (\mathcal{V}, \mathcal{E})$, an* ***HIN motif*** *is a structural pattern defined by a graph on the type level with its node being a node type of the original HIN and an edge being an edge type of the given HIN. Additional constraints can be optionally added such as two nodes in the motif cannot be simultaneously matched to the same node instance in the given HIN. Further given an HIN motif, an* ***HIN motif instance*** *under this motif is a subnetwork of the HIN that matches this pattern.*

Figure 2c gives an example of a motif in the DBLP network with four distinct terms referred to as $AP4TPA$. If a motif is a path graph, it is also called a meta-path [29]. The motif, $APTPA$, in Figure 2b is one such example.

Definition 3 (Tensor, k-mode product, mode-k matricization [19]). *A* ***tensor*** *is a multidimensional array. For an N-th–order tensor $\mathcal{X} \in \mathbb{R}^{d_1 \times \dots \times d_N}$, we denote its $(j_1, \dots, j_N)$ entry by $\mathcal{X}_{j_1,\dots,j_N}$. The* ***$k$-mode product*** *of $\mathcal{X}$ and a matrix $\mathbf{A} \in \mathbb{R}^{d_k \times d}$ is denoted by $\mathcal{Y} = \mathcal{X} \times_k \mathbf{A}$, where $\mathcal{Y} \in \mathbb{R}^{d_1 \times \dots \times d_{k-1} \times d \times d_{k+1} \times \dots \times d_N}$, and $\mathcal{Y}_{\dots,j_{k-1},j,j_{k+1},\dots} = \sum_{s=1}^{d_k} \mathcal{X}_{\dots,j_{k-1},s,j_{k+1},\dots} \mathbf{A}_{s,j}$. We denote matrix $\mathcal{X}_{(k)} \in \mathbb{R}^{(d_1 \cdot \ldots \cdot d_{k-1} \cdot d_{k+1} \cdot \ldots \cdot d_N) \times d_k}$ the* ***mode-k matricization****, i.e., mode-k unfolding, of the tensor $\mathcal{X}$, where the i-th column of $\mathcal{X}_{(k)}$ is obtained by vectorizing the $(n-1)$-th order tensor $\mathcal{X}_{\dots,:,j,:,\dots}$ with j on the k-th index.*

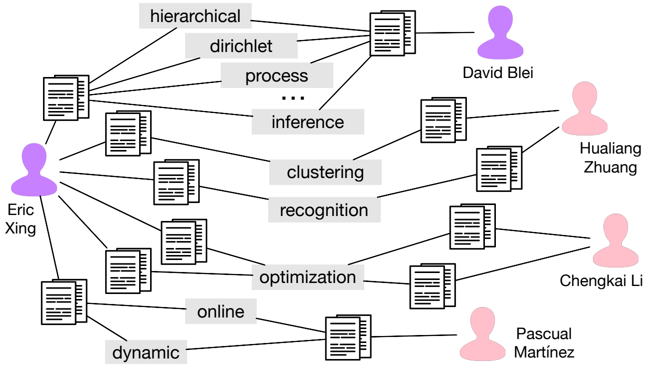


Fig. 3: A subnetwork of DBLP. According to the ground truth data, *Eric Xing* and *David Blei* were graduated from the same research group.

For simplicity, we denote $\mathcal{X} \times_{i=1}^{N} \mathbf{A}_i \coloneqq \mathcal{X} \times_1 \mathbf{A}_1 \times_2 \ldots \times_N \mathbf{A}_N$. Additionally, we define $[\otimes_{i=1}^{N \setminus k} \mathbf{A}_i] \coloneqq \mathbf{A}_1 \otimes \ldots \otimes \mathbf{A}_{k-1} \otimes \mathbf{A}_{k+1} \otimes \ldots \otimes \mathbf{A}_N$, where $\otimes$ is the Kronecker product [19].

Lastly, we introduce a useful lemma that converts the norm of the difference between two tensors to that between two matrices.

Lemma 4 ([4]). *For all $k \in \{1, 2, \ldots, N\}$,*

$$\left\| \mathcal{X} - \mathcal{Y} \times_{i=1}^{N} \mathbf{A}_i \right\|_F = \left\| \mathcal{X}_{(k)} - \mathbf{A}_k \mathcal{Y}_{(k)} [\otimes_{i=1}^{N \setminus k} \mathbf{A}_i]^\top \right\|_F,$$

where $\|\cdot\|_F$ is the Frobenius norm.

4 Higher-Order Interaction in Real-World Dataset.

In this section, we use a real-world example to motivate the design of our method that aims to comprehensively model higher-order interactions revealed by motifs.

DBLP is a bibliographical network in the computer science domain [31] that contains nodes with type author, paper, term, *etc.* In Figure 3, we plot out a subnetwork involving five authors: *Eric Xing*, *David Blei*, *Hualiang Zhuang*, *Chengkai Li*, and *Pascual Martinez*. According to the ground truth labels, *Xing* and *Blei* graduated from the same research group, while the other three authors graduated from other groups. Under meta-path $APTPA$, one would be able to find many path instances from *Xing* to authors from different groups. However, if we use motif $AP4TPA$, motif instances can only be found over *Xing* and *Blei*, but not between *Xing* and authors from other groups. This implies motifs can provide more subtle information than meta-paths do, and if a user wishes to cluster authors by research groups, motif $AP4TPA$ can be informative.

More importantly, if we look into the $AP4TPA$ motif instances that are matched to *Xing* and *Blei*, the involved terms such as *dirichlet* are very specific to their group's research interest. In other words, *dirichlet* represents an important semantic facet of the relationship between *Xing* and *Blei.* Modeling the higher-order interaction among *dirichlet* and other nodes can therefore kick in more information. If one only used motifs to generate pairwise or edge-level signals, such information would be lost. In Section 7, we will further quantitatively validate the utility of comprehensively modeling higher-order interactions.

5 The MoCHIN Model

In this section, we describe the proposed model with an emphasis on its intention to comprehensively model higher-order interaction while availing user guidance.

Revisit on clustering by non-negative matrix factorization. Non-negative matrix factorization (NMF) has been a popular method clustering [11,14]. Usually with additional constraints or regularization, the basic NMF-based algorithm solves the following optimization problem for given adjacency matrix M

$$\min_{\mathbf{V}_1,\mathbf{V}_2\geq 0} \left\|\mathbf{M} - \mathbf{V}_1^\top \mathbf{V}_2\right\|_F^2, \tag{1}$$

where $\|\cdot\|_F$ is the Frobenius norm, $\mathbf{A} \geq 0$ denotes matrix $\mathbf{A}$ is non-negative, and $\mathbf{V}_1$, $\mathbf{V}_2$ are two $C \times |\mathcal{V}|$ matrices with C being the number of clusters. In this model, the j-th column of $\mathbf{V}_1$ or that of $\mathbf{V}_2$ gives the inferred cluster membership of the j-th node in the network.

Single-motif–based clustering in HINs. Recall that an edge essentially characterizes the pairwise interaction between two nodes. To model higher-order interaction without collapsing it into pairwise interactions, a natural solution to clustering is using the inferred cluster membership of all involved nodes to reconstruct the existence of each motif instance. This solution can be formulated by non-negative tensor factorization (NTF), and studies on NTF per se and clustering via factorizing a single tensor can be found in the literature [19].

Specifically, given a single motif m with N nodes having node type $t_1, \ldots, t_N$ of the HIN, we transcribe the higher-order interaction revealed by this motif to a N-th–order tensor $\mathcal{X}$ with dimension $|\mathcal{V}_{t_1}| \times \ldots \times |\mathcal{V}_{t_N}|$. We set the $(j_1, \ldots, j_N)$ entry of $\mathcal{X}$ to 1 if a motif instance exists over the following n nodes: j_1-th of $\mathcal{V}_{t_1}$, $\ldots$, j_N-th of $\mathcal{V}_{t_N}$; and set it to 0 otherwise. By extending Eq. (1), whose objective is equivalent to $\left\|\mathbf{M} - \mathbf{V}_1^\top \mathbf{I} \mathbf{V}_2\right\|_F^2$ with $\mathbf{I}$ being the identity matrix, we can approach the clustering problem by solving

$$\min_{\mathbf{V}_1,\mathbf{V}_2\geq 0} \left\|\mathcal{X} - \mathcal{I} \times_{i=1}^{N} \mathbf{V}_i\right\|_F^2 + \lambda \sum_{i=1}^{N} \|\mathbf{V}_i\|_1, \tag{2}$$

where $\mathcal{I}$ is the N-th order identity tensor with dimension $C \times \ldots \times C$, $\|\cdot\|_1$ is the entry-wise l-1 norm introduced as regularization to avoid trivial solution, and λ is the regularization coefficient. We also note that this formulation is essentially the CP decomposition [19] with additional l-1 regularization and non-negative constraints. We write this formula in a way different from its most common form for notation convenience in the inference section (Section 6) considering the presence of regularization and constraints.

Symbol	Definition	Symbol	Definition
$\mathcal{V}$, $\mathcal{E}$	The set of nodes and the set of edges	$\mathcal{X}^{(m)}$	The tensor constructed from motif m
φ, ψ	The node and the edge type mapping	$\mathbf{M}^{(t)}$	The seed mask matrix for node type t
$\mathcal{T}$, $\mathcal{R}$, $\mathcal{M}$	The set of node types, edge types, and candidate motifs	$\mathbf{V}_i^{(m)}$	The cluster membership matrix for the i-th node in motif m
$\mathcal{V}_t$	The set of all nodes with type t	$\mathbf{V}_t^*$	The consensus matrix for node type t
$o(m)$	The number of nodes in motif $m \in \mathcal{M}$	$\boldsymbol{\mu}$	The vector $(\mu_1, \ldots, \mu_{\lvert\mathcal{M}\rvert})$ of motif weights
C	The number of clusters	$\times_k$	The mode-k product of a tensor and a matrix
λ, θ, ρ	The hyperparameters	$\otimes$	The Kronecker product of two matrices

Table 1: Summary of symbols

Proposed model for motif-based clustering in HINs. Real-world HINs often contain rich and diverse semantic facets due to its heterogeneity [25,28,29]. To reflect the different semantic facets of an HIN, a set $\mathcal{M}$ of more than one candidate motifs are usually necessary for the task of user-guided clustering. With additional clustering seeds provided by users, the MoCHIN model selects the motifs that are both meaningful and pertinent to the seeds.

To this end, we assign motif-specific weights $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_{\lvert\mathcal{M}\rvert})$, such that $\sum_{m\in\mathcal{M}} \mu_m = 1$ and $\mu_m \geq 0$ for all $m \in \mathcal{M}$. Denote $\mathcal{X}^{(m)}$ the tensor for motif m, $\mathbf{V}_i^{(m)}$ the cluster membership matrix for the i-th node in motif m, $o(m)$ the number of nodes in motif m, and $\varphi(m,i)$ the node type of the i-th node in motif m. For each node type $t \in \mathcal{T}$, we put together cluster membership matrices concerning this type and motif weights to construct the consensus matrix $\mathbf{V}_t^* := \sum_{\varphi(m,i)=t} \frac{\mu_m \mathbf{V}_i^{(m)}}{\sum_{i'=1}^{o(m)} \mathbb{1}_{[\varphi(m,i')=\varphi(m,i)]}}$, where $\mathbb{1}_{[P]}$ equals to 1 if P is true and 0 otherwise. With this notation, $\sum_{i'=1}^{o(m)} \mathbb{1}_{[\varphi(m,i')=\varphi(m,i)]}$ is simply the number of nodes in motif m that are of type t.

Furthermore, we intend to let (i) each cluster membership $\mathbf{V}_i^{(m)}$ be close to its corresponding consensus matrix $\mathbf{V}_{\varphi(m,i)}^*$ and (ii) the consensus matrices not assign seed nodes to the wrong cluster. We hence propose the following objective with the third and the fourth term modeling the aforementioned two intentions

$$\begin{aligned}\mathcal{O} = \sum_{m\in\mathcal{M}} \left\| \mathcal{X}^{(m)} - \mathcal{I}^{(m)} \times_{i=1}^{o(m)} \mathbf{V}_i^{(m)} \right\|_F^2 + \lambda \sum_{m\in\mathcal{M}} \sum_{i=1}^{o(m)} \left\| \mathbf{V}_i^{(m)} \right\|_1 \\ + \theta \sum_{m\in\mathcal{M}} \sum_{i=1}^{o(m)} \left\| \mathbf{V}_i^{(m)} - \mathbf{V}_{\varphi(m,i)}^* \right\|_F^2 + \rho \sum_{t\in\mathcal{T}} \left\| \mathbf{M}^{(t)} \circ \mathbf{V}_t^* \right\|_F^2,\end{aligned} \tag{3}$$

where $\circ$ is the Hadamard product and $\mathbf{M}^{(t)}$ is the seed mask matrix for node type t. Its (i,c) entry $\mathbf{M}_{i,c}^{(t)} = 1$ if the i-th node of type t is a seed node and it should not be assigned to cluster c, and $\mathbf{M}_{i,c}^{(t)} = 0$ otherwise.

Finally, solving the problem of HIN clustering by modeling higher-order interaction and automatically selecting motifs is converted to solving the following optimization problem with Δ being the standard simplex

$$\min_{\{\mathbf{V}_i^{(m)} \geq 0\}, \boldsymbol{\mu}\in\Delta} \quad \mathcal{O}. \tag{4}$$

6 The Inference Algorithm

In this section, we first describe the algorithm for solving the optimization problem as in Eq. (4). Then, a series of speed-up tricks are introduced to circumvent the curse of dimensionality, so that the complexity is governed no longer by the dimension of the tensors but by the number of motif instances in the network.

Update $\mathbf{V}_k^{(l)}$ and μ. Each clustering membership matrix $\mathbf{V}_k^{(l)}$ with non-negative constraints is involved in all terms of the objective function (Eq. (3)), where $l \in \mathcal{M}$ and $k \in \{1, \dots, o(l)\}$. We hence develop multiplicative update rules for $\mathbf{V}_k^{(l)}$ that guarantees monotonic decrease at each step, accompanied by projected gradient descent (PGD) to find global optimal of $\boldsymbol{\mu} = [\mu_1, \dots, \mu_{|\mathcal{M}|}]^\top$. Overall, we solve the optimization problem by alternating between $\{\mathbf{V}_k^{(l)}\}$ and $\boldsymbol{\mu}$.

To update $\mathbf{V}_k^{(l)}$ when $\{\mathbf{V}_i^{(m)}\}_{(m,i)\neq(l,k)}$ and $\boldsymbol{\mu}$ are fixed under non-negative constraints, we derive the following theorem. For notation convenience, we further denote $\mathbf{V}_t^* = \sum_{\varphi(m,i)=t} \eta_i^m \mathbf{V}_i^{(m)}$, where $\eta_i^m := \frac{\mu_m}{\sum_{i'=1}^{o(m)} \mathbb{1}_{[\varphi(m,i')=\varphi(m,i)]}}$.

Theorem 5. *The following update rule for $\mathbf{V}_k^{(l)}$ monotonically decreases the objective function.*

$$\mathbf{V}_k^{(l)} \leftarrow \mathbf{V}_k^{(l)} \circ \left[\frac{\mathcal{X}_{(k)}^{(l)}[\otimes_{i=1}^{o(l)\setminus k}\mathbf{V}_i^{(l)}]\mathcal{I}_{(k)}^{(l)\top} + \theta(1-\eta_k^l)(\mathbf{V}_{\varphi(l,k)}^* - \eta_k^l\mathbf{V}_k^{(l)}) + \theta\eta_k^l \sum_{\varphi(m,i)=\varphi(l,k)}^{(m,i)\neq(l,k)} [\mathbf{V}_i^{(m)} - \mathbf{V}_{\varphi(l,k)}^* + \eta_k^l \mathbf{V}_k^{(l)}]^+}{\mathbf{V}_k^{(l)}\mathcal{I}_{(k)}^{(l)}[\otimes_{i=1}^{o(l)\setminus k}\mathbf{V}_i^{(l)}]^\top[\otimes_{i=1}^{o(l)\setminus k}\mathbf{V}_i^{(l)}]\mathcal{I}_{(k)}^{(l)\top} + \rho\eta_k^l\mathbf{M}^{\varphi(l,k)} \circ \mathbf{V}_{\varphi(l,k)}^* + \theta\eta_k^l \sum_{\varphi(m,i)=\varphi(l,k)}^{(m,i)\neq(l,k)} ([\mathbf{V}_i^{(m)} - \mathbf{V}_{\varphi(l,k)}^* + \eta_k^l\mathbf{V}_k^{(l)}]^- + \eta_k^l\mathbf{V}_k^{(l)}) + \theta(1-\eta_k^l)^2\mathbf{V}_k^{(l)} + \lambda} \right]^{\frac{1}{2}}, \tag{5}$$

where for any matrix $\mathbf{A}$, $[\mathbf{A}]^+ := \frac{|\mathbf{A}|+\mathbf{A}}{2}$, $[\mathbf{A}]^- := \frac{|\mathbf{A}|-\mathbf{A}}{2}$.

We defer the proof of this theorem to Section 1 of the supplementary file. For fixed $\{\mathbf{V}_i^{(m)}\}$, the objective function Eq. (3) is convex with respect to $\boldsymbol{\mu}$. We therefore use PGD to update $\boldsymbol{\mu}$, where the gradient can be analytically derived with straightforward calculation.

Computational Speed-Up. Unlike the scenario where researchers solve the NTF problem with tensors of order independent of the applied dataset, our problem is specifically challenging because the tensor size grows exponentially with the tensor order. For instance, the AP4TPA motif discussed in Section 4 is one real-world example involving 8 nodes, which leads to an 8-th order tensor.

In the proposed inference algorithm, the direct computation of three terms entails complexity subject to the size of the tensor: (i) the first term in the numerator of Eq. (5), $\mathcal{X}_{(k)}^{(l)}[\otimes_{i=1}^{o(l)\setminus k}\mathbf{V}_i^{(l)}]\mathcal{I}_{(k)}^{(l)\top}$, (ii) the first term in the denominator of Eq. (5), $\mathbf{V}_k^{(l)}\mathcal{I}_{(k)}^{(l)}[\otimes_{i=1}^{o(l)\setminus k}\mathbf{V}_i^{(l)}]^\top$ $[\otimes_{i=1}^{o(l)\setminus k}\mathbf{V}_i^{(l)}]\mathcal{I}_{(k)}^{(l)\top}$, and (iii) the first term of the objective function Eq. (3), $\left\|\mathcal{X}^{(m)} - \mathcal{I}^{(m)} \times_{i=1}^{o(m)} \mathbf{V}_i^{(m)}\right\|_F^2$. Fortunately, all

Algorithm 1: The MoCHIN inference algorithm

```
Input  : $\{\mathcal{X}^{(m)}\}$, supervision $\mathbf{M}^{(t)}$, the number of clusters $C$, hyperparameters
         $\theta$, $\rho$, and $\lambda$
Output: the cluster membership matrices $\{\mathbf{V}_t^*\}$
begin
    while not converged do
        for $m \in \mathcal{M}$ do
            while not converged do
                for $i \in \{1, \ldots, o(m)\}$ do
                    Find local optimum of $\mathbf{V}_i^{(m)}$ by Eq. (5).
        Find global optimum of $\boldsymbol{\mu}$ by PGD.
```

these terms can be significantly simplified by exploiting the composition of dense matrix $[\otimes_{i=1}^{o(l)\setminus k}\mathbf{V}_i^{(l)}]\mathcal{I}_{(k)}^{(l)\top}$ and the sparsity of tensor $\mathcal{X}^{(l)}$ ($\mathcal{X}^{(m)}$).

Consider the example that motif $l \in \mathcal{M}$ involves 5 nodes, each node type has $10,000$ node instances, and the nodes are to be clustered into 10 clusters. Then the induced dense matrix $[\otimes_{i=1}^{o(l)\setminus k}\mathbf{V}_i^{(l)}]\mathcal{I}_{(k)}^{(l)\top}$ would have $\prod_{\substack{i=1\\i\neq k}}^{o(l)} |\mathcal{V}_{\varphi(l,i)}| \cdot C^{o(l)-1} = 10^{20}$ entries. As a result, computing term (i), $\mathcal{X}_{(k)}^{(l)}[\otimes_{i=1}^{o(l)\setminus k}\mathbf{V}_i^{(l)}]\mathcal{I}_{(k)}^{(l)\top}$, would involve matrix multiplication of a dense 10^{20} entry matrix. However, given the sparsity of $\mathcal{X}^{(l)}$, one may denote the set of indices of the non-zero entries in tensor $\mathcal{X}$ by $\text{nz}(\mathcal{X}) \coloneqq \{J = (j_1, \ldots, j_N) \mid \mathcal{X}_{j_1,\ldots,j_N} \neq 0\}$ and derive the following equivalency

$$\mathcal{X}_{(k)}^{(l)}[\otimes_{i=1}^{o(l)\setminus k}\mathbf{V}_i^{(l)}]\mathcal{I}_{(k)}^{(l)\top} = \sum_{J\in \text{nz}(\mathcal{X}^{(l)})} \mathcal{X}^{(l)}_{j_1,\ldots,j_{o(l)}}\mathbf{h}(j_k)\prod_{\substack{i=1\\i\neq k}}^{o(l)}(\mathbf{V}_i^{(l)})_{j_i,:},$$

where $\prod$ is Hadamard product of a sequence and $\mathbf{h}(j_k)$ is one-hot column vector of size $|\mathcal{V}_{\varphi(l,k)}|$ that has entry 1 at index j_k. Computing the right-hand side of this equivalency involves the summation over Hadamard product of a small sequence of small vectors, which has a complexity of $O(\text{nnz}(\mathcal{X}^{(l)}) \cdot (o(l) - 1) \cdot C)$ with $\text{nnz}(\cdot)$ being the number of non-zero entries. In other words, if the previous example comes with $1,000,000$ motif instances, the complexity would decrease from manipulating a 10^{20}-entry dense matrix to a magnitude of 4×10^7.

Similarly, by leveraging the sparsity of tensors and composition of dense matrices, one can simplify the computation of term (ii) from multiplication of matrix with 10^{20} entries to that with 10^5 entries; and reduce the calculation of term (iii) from a magnitude of 10^{20} to a magnitude of 10^8. We provide detailed derivation and formulas in the supplementary file.

Finally, we remark that the above computation can be highly parallelized, which has further promoted the efficiency in our implementation. An empirical efficiency study is available in Section 3 of the supplementary file. We summarize the algorithm in Algorithm 1.

7 Experiments

We present the quantitative evaluation results on two real-world datasets through multiple tasks and conduct case studies under various circumstances.

7.1 Datasets and Evaluation Tasks

In this section, we briefly describe (i) the datasets, (ii) the evaluation tasks, and (iii) the metrics used in the experiments. All of their detailed descriptions are provided in Section 4 of the supplementary file.

Datasets. We use two real-world HINs for experiments. **DBLP** is a heterogeneous information network that serves as a bibliography of research in computer science area [31]. The network consists of 5 types of node: author (A), paper (P), key term (T), venue (V) and year (Y). In DBLP, we select two candidate motifs for all applicable methods, including $AP4TPA$ and $APPA$. **YAGO** is a knowledge graph constructed by merging Wikipedia, GeoNames and WordNet. YAGO dataset consists of 7 types of nodes: person (P), organization (O), location (L), prize (R), work (W), position (S) and event (E). In YAGO, the candidate motifs used by all compared methods include $P^6O^{23}L$, $P^7O^{23}L$, $P^8O^{23}L$, $2P2W$, $3PW$.

Evaluation tasks. In order to evaluate models' capability in reflecting different user guidance, we use two sets of labels on authors to conduct two tasks in DBLP similar to previous study [29]. Additionally, we design another task on YAGO with labels on persons. **DBLP-group** – Clustering authors to 5 research groups where they graduated. 5% of the 250 authors with labels are randomly selected as seeds from user guidance. **DBLP-area** – Clustering authors to 14 research areas. 1% of the $7,165$ authors with labels are randomly selected as seeds from user guidance. **YAGO** – Clustering people to 10 popular countries in the YAGO dataset. 1% of the $11,368$ people are randomly selected as seeds from user guidance.

Evaluation metrics. We use three metrics to evaluate the quality of the clustering results generated by each model: Accuracy (Micro-F1), Macro-F1, and NMI. Note that in multi-class classification tasks, accuracy is always identical to Micro-F1. For all these metrics, higher values indicate better performance.

7.2 Baselines and Experiment Setups

Baselines. We use five different baselines to obtain insight on different aspects of the performance of MoCHIN. **KNN** is a classification algorithm that assigns the label of each object in the test set is according to its nearest neighbors. In our scenario, the distance between two nodes is defined as the length of the shortest path between them. **KNN+Motifs** uses signals generated by motifs, but does not directly model all players in higher-order interactions. To extract information from motifs, we construct a motif-based network for each candidate motif,

Table 2: Quantitative evaluation on clustering results in three tasks.

Task	DBLP-group			DBLP-area			YAGO		
Metric	Acc./Micro-F1	Macro-F1	NMI	Acc./Micro-F1	Macro-F1	NMI	Acc./Micro-F1	Macro-F1	NMI
KNN	0.4249	0.2566	0.1254	0.4107	0.4167	0.2537	0.3268	0.0921	0.0810
KNN+Motifs	0.4549	0.2769	0.1527	0.4811	0.4905	0.3296	0.3951	0.1885	0.1660
GNetMine [9]	0.5880	0.6122	0.3325	0.4847	0.4881	0.3469	0.3832	0.2879	0.1772
PathSelClus [29]	0.5622	0.5535	0.3246	0.4361	0.4520	0.3967	0.3856	0.3405	0.2864
TGS [3]	0.6609	0.6513	0.3958	0.4391	0.4365	0.2790	0.6058	0.3564	0.4406
MoCHIN	**0.7382**	**0.7387**	**0.5797**	**0.5318**	**0.5464**	**0.4396**	**0.6134**	**0.5563**	**0.4607**

where an edge is constructed if two nodes are matched to a motif instance in the original HIN. KNN is then applied to each motif-based network. Finally, a linear combination is applied to the outcome probability matrices generated by KNN from the motif-based networks and the original HIN with weights tuned to the best. **GNetMine** [9] is a graph-based regularization framework to address the transductive classification problem in HINs. This method only leverages edge-level information without considering structural patterns such as meta-paths or motifs. **PathSelClus** [29] is a probabilistic graphical model that performs clustering tasks on HINs by integrating meta-path selection with user-guided clustering. For this baseline, we additionally add $APVPA$, $APTPA$, APT, APA, and, $APAPA$ into the set of candidate meta-paths for both DBLP tasks as suggested by the original paper [29] and add $P^{14}O^{14}P$, $P^{15}O^{15}P$, and $P^{16}O^{16}P$ for YAGO task. **TGS** [3] leverages motifs but does not directly model each occurrence of higher-order interaction. It is hence another direct comparison to MoCHIN, besides KNN+Motifs, which is used to analyze the utility of comprehensively transcribing motif instances into tensors. As the authors did not discuss how to inject user guidance into their basic bipartitioning clustering algorithm, we apply multi-class logistic regression on the accompanied typed-graphlet spectral embedding algorithm proposed in the same paper. The typed-graphlet adjacency matrices of multiple motifs are summed together to derive the input for the algorithm as the author suggested in the paper.

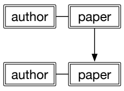
Experiment setups. For MoCHIN, we set hyperparameters $\theta = 1$, $\rho = 100$ and $\lambda = 0.0001$ across all tasks in our experiments. For each model involving motifs, edge-level motifs corresponding to the edge types are included into the set of candidate motifs. For each baseline in each task, we always tune its hyperparameters to achieve the best performance.

7.3 Quantitative Evaluation Result

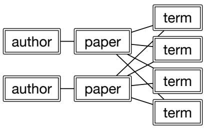
We report the main quantitative results in Table 2. Overall, MoCHIN uniformly outperformed all baselines in all three tasks under all metrics. Note that these three metrics measure different aspects of the model performance. For instance, in the DBLP-area task, PathSelClus outperforms GNetMine under Macro-F1 and NMI, while GNetMine outperforms PathSelClus under Acc./Micro-F1. Achieving superior performance uniformly under all metrics is hence strong evidence

Table 3: Ablation study of the MoCHIN model on the DBLP-group task with the non–edge-level motifs, $APPA$ and $AP4TPA$, optionally removing from the full model.

Metric	Acc./Micro-F1	Macro-F1	NMI	Result for Eric Xing
W/o both	0.6567	0.6411	0.5157	✗
W/ APPA	0.7039	0.7062	0.5166	✗
W/ AP4TPA	0.6781	0.6589	0.5502	✓
Full model	**0.7382**	**0.7387**	**0.5797**	✓



The APPA motif.



The AP4TPA motif.

that MoCHIN with higher-order interaction directly modeled is armed with greater modeling capability in the task of user-guided HIN clustering.

MoCHIN prevails in user-guided clustering by exploiting signals from motifs more comprehensively. Recall that KNN+Motifs, TGS, and MoCHIN all exploit signals from motifs. However, the two baselines do not directly model each occurrence of motif instances and only preserve pairwise or edge-level information. In our experiments, even though TGS can generally outperform other baselines, it alongside KNN+Motifs still cannot generate results as good as MoCHIN, which demonstrates the utility of more comprehensively exploiting signals from motifs as MoCHIN does. We interpret this result as when user guidance is limited, a fine-grained understanding of the rich semantics of an HIN is instrumental in dissecting users' intention and generating desirable results.

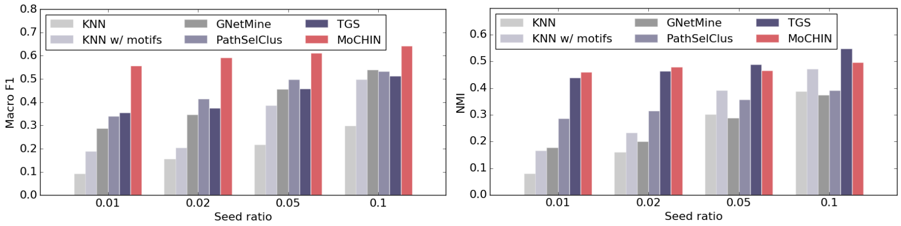
Impact of candidate motif choice. In this section, we study how the choice of candidate motifs impacts MoCHIN and additionally use the concrete example in Figure 3 to understand the model outputs. Particularly, we conducted an ablation study by taking out either or both of the two non–edge-level motifs, $APPA$ and $AP4TPA$, in the DBLP-group task and reported the result in Table 3. The full MoCHIN model outperformed all partial models, demonstrating the utility of these motifs in clustering.

Moreover, we scrutinized the concrete example in Figure 3 and checked how each model assigned cluster membership for Eric Xing. The result is also included in Table 3, which shows only the model variants with $AP4TPA$ made the correct assignment on Eric Xing. In Section 2 of the supplementary file, a visualization of this ablation study is provided to further corroborate our observation.

7.4 Varied Seed Ratio

In addition to using 1% people as seeds for the YAGO task reported in Table 2, we experiment under varied seed ratio 2%, 5%, and 10%. The results are reported in Figure 4. We omit Accuracy (Micro-F1), which has a similar trend with NMI.

For all methods, the performance increased as the seed ratio increased. Notably, MoCHIN outperformed most baselines, especially when seed ratio is small. This suggests MoCHIN is particularly useful when users provide less guidance for being able to better exploit subtle information from limited data.



(a) Macro-F1.

(b) NMI.

Fig. 4: Quantitative evaluation on the YAGO task under varied seed ratio.

Note that higher seed ratio is uncommon in practice since it is demanding for users to provide more than a few seeds.

Lastly, an efficiency study that empirically evaluates the proposed algorithm is provided in Section 3 of the supplementary file.

8 Discussion, Conclusion, and Future Work

One limitation of MoCHIN is that it may not be easily applied to very large datasets even with speed-up methods due to the complexity of the model itself. However, MoCHIN would stand out in the scenario where fine-grained understanding of the network semantics is needed. In the experiment, we have shown that MoCHIN can scale to HINs with tens of thousands of nodes. We note that for user-guided clustering, it is possible the users are mostly interested in the data instances most relevant to their intention, which could be a subset of a larger dataset. For instance, if a data mining researcher wanted to cluster DBLP authors by research group, it is possible they would not care about the nodes not relevant to data mining research. As such the majority of the millions of nodes in DBLP can be filtered out in preprocessing, and this user-guided clustering problem would become not only manageable to MoCHIN but also favorable due to MoCHIN's capability in handling fine-grained semantics. Moreover, in the case where the network is inevitably large, one may trade the performance of MoCHIN for efficiency by using only relatively simple motifs. It is also worth noting that incremental learning is possible for MoCHIN – when new nodes are available, one do not have to retrain the model from scratch.

In conclusion, we studied the problem of user-guided clustering in HINs with the intention to model higher-order interactions. We identified the importance of modeling higher-order interactions without collapsing them into pairwise interactions and proposed the MoCHIN algorithm. Experiments validated the effectiveness of the proposed model and the utility of comprehensively modeling higher-order interactions. Future works include exploring further methodologies to join signals from multiple motifs, which is currently realized by a simple linear combination in the MoCHIN model. Furthermore, as the current model takes user guidance by injecting labels of the seeds, it is also of interest to extend MoCHIN to the scenario where guidance is made available by must-link and cannot-link constraints on node pairs.

Acknowledgments. This work was sponsored in part by U.S. Army Research Lab. under Cooperative Agreement No. W911NF-09-2-0053 (NSCTA), DARPA under Agreement No. W911NF-17-C-0099, National Science Foundation IIS 16-18481, IIS 17-04532, and IIS-17-41317, DTRA HDTRA11810026, and grant 1U54GM114838 awarded by NIGMS through funds provided by the trans-NIH Big Data to Knowledge (BD2K) initiative (www.bd2k.nih.gov). Any opinions, findings, and conclusions or recommendations expressed in this document are those of the author(s) and should not be interpreted as the views of any U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

References

1. Ahmed, N.K., Neville, J., Rossi, R.A., Duffield, N.: Efficient graphlet counting for large networks. In: ICDM (2015)
2. Benson, A.R., Gleich, D.F., Leskovec, J.: Higher-order organization of complex networks. Science **353**(6295), 163–166 (2016)
3. Carranza, A.G., Rossi, R.A., Rao, A., Koh, E.: Higher-order spectral clustering for heterogeneous graphs. arXiv preprint arXiv:1810.02959 (2018)
4. De Lathauwer, L., De Moor, B., Vandewalle, J.: A multilinear singular value decomposition. SIMAX (2000)
5. Fang, Y., Lin, W., Zheng, V.W., Wu, M., Chang, K.C.C., Li, X.L.: Semantic proximity search on graphs with metagraph-based learning. In: ICDE. IEEE (2016)
6. Gujral, E., Papalexakis, E.E.: Smacd: Semi-supervised multi-aspect community detection. In: ICDM (2018)
7. Han, J., Pei, J., Kamber, M.: Data mining: concepts and techniques. Elsevier (2011)
8. Huang, Z., Zheng, Y., Cheng, R., Sun, Y., Mamoulis, N., Li, X.: Meta structure: Computing relevance in large heterogeneous information networks. In: KDD. ACM (2016)
9. Ji, M., Sun, Y., Danilevsky, M., Han, J., Gao, J.: Graph regularized transductive classification on heterogeneous information networks. In: ECMLPKDD. pp. 570–586. Springer (2010)
10. Jiang, H., Song, Y., Wang, C., Zhang, M., Sun, Y.: Semi-supervised learning over heterogeneous information networks by ensemble of meta-graph guided random walks. In: AAAI (2017)
11. Lee, D.D., Seung, H.S.: Algorithms for non-negative matrix factorization. In: Advances in neural information processing systems. pp. 556–562 (2001)
12. Li, P., Milenkovic, O.: Inhomogeneous hypergraph clustering with applications. In: NIPS (2017)
13. Li, X., Wu, Y., Ester, M., Kao, B., Wang, X., Zheng, Y.: Semi-supervised clustering in attributed heterogeneous information networks. In: WWW (2017)
14. Liu, J., Wang, C., Gao, J., Han, J.: Multi-view clustering via joint nonnegative matrix factorization. In: SDM. vol. 13, pp. 252–260. SIAM (2013)
15. Liu, Z., Zheng, V.W., Zhao, Z., Li, Z., Yang, H., Wu, M., Ying, J.: Interactive paths embedding for semantic proximity search on heterogeneous graphs. In: KDD (2018)
16. Liu, Z., Zheng, V.W., Zhao, Z., Zhu, F., Chang, K.C.C., Wu, M., Ying, J.: Distance-aware dag embedding for proximity search on heterogeneous graphs. AAAI (2018)

17. Luo, C., Pang, W., Wang, Z.: Semi-supervised clustering on heterogeneous information networks. In: PAKDD (2014)
18. Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Chklovskii, D., Alon, U.: Network motifs: simple building blocks of complex networks. Science **298**(5594), 824–827 (2002)
19. Papalexakis, E.E., Faloutsos, C., Sidiropoulos, N.D.: Tensors for data mining and data fusion: Models, applications, and scalable algorithms. TIST **8**(2), 16 (2017)
20. Sankar, A., Zhang, X., Chang, K.C.C.: Motif-based convolutional neural network on graphs. arXiv preprint arXiv:1711.05697 (2017)
21. Shi, C., Li, Y., Zhang, J., Sun, Y., Philip, S.Y.: A survey of heterogeneous information network analysis. TKDE **29**(1), 17–37 (2017)
22. Shi, C., Wang, R., Li, Y., Yu, P.S., Wu, B.: Ranking-based clustering on general heterogeneous information networks by network projection. In: CIKM (2014)
23. Shi, Y., Chan, P.W., Zhuang, H., Gui, H., Han, J.: Prep: Path-based relevance from a probabilistic perspective in heterogeneous information networks. In: KDD (2017)
24. Shi, Y., Gui, H., Zhu, Q., Kaplan, L., Han, J.: Aspem: Embedding learning by aspects in heterogeneous information networks. In: SDM (2018)
25. Shi, Y., Zhu, Q., Guo, F., Zhang, C., Han, J.: Easing embedding learning by comprehensive transcription of heterogeneous information networks. In: KDD (2018)
26. Stefani, L.D., Epasto, A., Riondato, M., Upfal, E.: Triest: Counting local and global triangles in fully dynamic streams with fixed memory size. TKDD **11**(4), 43 (2017)
27. Strehl, A., Ghosh, J.: Cluster ensembles—a knowledge reuse framework for combining multiple partitions. JMLR **3**(Dec), 583–617 (2002)
28. Sun, Y., Han, J.: Mining heterogeneous information networks: a structural analysis approach. SIGKDD Explorations **14**(2), 20–28 (2013)
29. Sun, Y., Norick, B., Han, J., Yan, X., Yu, P.S., Yu, X.: Integrating meta-path selection with user-guided object clustering in heterogeneous information networks. In: KDD (2012)
30. Sun, Y., Yu, Y., Han, J.: Ranking-based clustering of heterogeneous information networks with star network schema. In: KDD. pp. 797–806. ACM (2009)
31. Tang, J., Zhang, J., Yao, L., Li, J., Zhang, L., Su, Z.: Arnetminer: extraction and mining of academic social networks. In: KDD (2008)
32. Wu, J., Wang, Z., Wu, Y., Liu, L., Deng, S., Huang, H.: A tensor cp decomposition method for clustering heterogeneous information networks via stochastic gradient descent algorithms. Scientific Programming **2017** (2017)
33. Yang, C., Feng, Y., Li, P., Shi, Y., Han, J.: Meta-graph based hin spectral embedding: Methods, analyses, and insights. In: ICDM (2018)
34. Yang, C., Liu, M., Zheng, V.W., Han, J.: Node, motif and subgraph: Leveraging network functional blocks through structural convolution. In: ASONAM (2018)
35. Yaveroğlu, Ö.N., Malod-Dognin, N., Davis, D., Levnajic, Z., Janjic, V., Karapandza, R., Stojmirovic, A., Pržulj, N.: Revealing the hidden language of complex networks. Scientific reports **4**, 4547 (2014)
36. Yin, H., Benson, A.R., Leskovec, J., Gleich, D.F.: Local higher-order graph clustering. In: KDD (2017)
37. Zhao, H., Xu, X., Song, Y., Lee, D.L., Chen, Z., Gao, H.: Ranking users in social networks with higher-order structures. In: AAAI (2018)
38. Zhao, H., Yao, Q., Li, J., Song, Y., Lee, D.L.: Meta-graph based recommendation fusion over heterogeneous information networks. In: KDD (2017)
39. Zhou, D., Zhang, S., Yildirim, M.Y., Alcorn, S., Tong, H., Davulcu, H., He, J.: A local algorithm for structure-preserving graph cut. In: KDD (2017)

Supplementary File for "User-Guided Clustering in Heterogeneous Information Networks via Motif-Based Comprehensive Transcription"

Yu Shi*, Xinwei He*, Naijing Zhang*, Carl Yang, and Jiawei Han

University of Illinois at Urbana-Champaign, Urbana, IL USA
{yushi2, xhe17, nzhang31, jiyang3, hanj}@illinois.edu

1 Additional Proof and Formulas

1.1 The Proof for Theorem 1 in the Main File

Inspired by prior art on non-negative matrix factorization [12], we provide the proof for Theorem 1 in the main file on tensor factorization as follows.

Proof. With the equivalency given by Lemma 1 in the main file

$$\left\|\mathcal{X}^{(m)} - \mathcal{I}^{(m)} \times_{i=1}^{o(m)} \mathbf{V}_i^{(m)}\right\|_F = \left\|\mathcal{X}_{(k)}^{(m)} - \mathbf{V}_k^{(m)} \mathcal{I}_{(k)}^{(m)} [\otimes_{i=1}^{o(m)\setminus k} \mathbf{V}_k^{(m)}]^\top\right\|_F,$$

we construct the auxiliary function

$$\begin{aligned}
\mathcal{Z}(\mathbf{V}_k^{(l)}, \widetilde{\mathbf{V}}) = \sum_{s,t} \{ & (\mathbf{V}_k^{(l)} \mathcal{I}_{(k)}^{(l)} [\otimes_{i=1}^{N\setminus k} \mathbf{V}_i^{(l)}]^\top [\otimes_{i=1}^{N\setminus k} \mathbf{V}_i^{(l)}] \mathcal{I}_{(k)}^{(l)\top})_{s,t} (\widetilde{\mathbf{V}})_{s,t}^2 / (\mathbf{V}_k^{(l)})_{s,t} \\
& -2(\mathcal{X}_{(k)}^{(l)} [\otimes_{i=1}^{N\setminus k} \mathbf{V}_i^{(l)}] \mathcal{I}_{(k)}^{(l)\top})_{s,t} (\mathbf{V}_k^{(l)})_{s,t} (1 + \log \frac{(\widetilde{\mathbf{V}})_{s,t}}{(\mathbf{V}_k^{(l)})_{s,t}}) \\
& +\theta(1-\eta_k^l)^2 (\widetilde{\mathbf{V}})_{s,t}^2 - 2\theta(1-\eta_k^l)(\mathbf{V}_{\varphi(l,k)}^* - \eta_k^l \mathbf{V}_k^{(l)}) \\
& \cdot(\mathbf{V}_k^{(l)})_{s,t} (1 + \log \frac{(\widetilde{\mathbf{V}})_{s,t}}{(\mathbf{V}_k^{(l)})_{s,t}}) + \theta \sum_{\varphi(m,i)=\varphi(l,k)}^{(m,i)\neq(l,k)} \eta_k^{l2} (\widetilde{\mathbf{V}})_{s,t}^2 \\
& -2([\theta \sum_{\varphi(m,i)=\varphi(l,k)}^{(m,i)\neq(l,k)} \eta_k^l (\mathbf{V}_i^{(m)} - \mathbf{V}_{\varphi(l,k)}^* + \eta_k^l \mathbf{V}_k^{(l)})]^+)_{s,t} \\
& \cdot(\mathbf{V}_k^{(l)})_{s,t} (1 + \log \frac{(\widetilde{\mathbf{V}})_{s,t}}{(\mathbf{V}_k^{(l)})_{s,t}}) + ([\theta \sum_{\varphi(m,i)=\varphi(l,k)}^{(m,i)\neq(l,k)} \eta_k^l (\mathbf{V}_i^{(m)} \\
& -\mathbf{V}_{\varphi(l,k)}^* + \eta_k^l \mathbf{V}_k^{(l)})]^-)_{s,t} ((\mathbf{V}_k^{(l)})_{s,t}^2 + (\widetilde{\mathbf{V}})_{s,t}^2) / (\mathbf{V}_k^{(l)})_{s,t} \\
& +\rho(\mathbf{M}^{\varphi(l,k)} \circ (\mathbf{V}_{\varphi(l,k)}^* - \eta_k^l \mathbf{V}_k^{(l)} + \eta_k^l \widetilde{\mathbf{V}}))_{s,t}^2 \\
& +\lambda((\mathbf{V}_k^{(l)})_{s,t}^2 + (\widetilde{\mathbf{V}})_{s,t}^2) / (2(\mathbf{V}_k^{(l)})_{s,t}) \}.
\end{aligned}$$

Straightforward derivation can show the following three relations hold:

* These authors contributed equally to this work.

1. $\mathcal{Z}(\mathbf{V}_k^{(l)}, \mathbf{V}_k^{(l)}) = \mathcal{O}(\mathbf{V}_k^{(l)})$,
2. $\mathcal{Z}(\mathbf{V}_k^{(l)}, \widetilde{\mathbf{V}}) \geq \mathcal{O}(\widetilde{\mathbf{V}})$, and
3. $\mathcal{Z}(\mathbf{V}_k^{(l)}, \widetilde{\mathbf{V}})$ is convex with respect to $\widetilde{\mathbf{V}}$.

Therefore, by setting $\frac{\partial}{\partial \widetilde{\mathbf{V}}} \mathcal{Z}(\mathbf{V}_k^{(l)}, \widetilde{\mathbf{V}}) = 0$, one can find $\mathcal{Z}(\mathbf{V}_k^{(l)}, \widetilde{\mathbf{V}})$ is minimized at $\widetilde{\mathbf{V}} = \widetilde{\mathbf{V}}_{\text{opt}}$, where $\widetilde{\mathbf{V}}_{\text{opt}}$ is the righthand side of Eq. (5) in the main file, and $\mathcal{O}(\mathbf{V}_k^{(l)}) = \mathcal{Z}(\mathbf{V}_k^{(l)}, \mathbf{V}_k^{(l)}) \geq \mathcal{Z}(\mathbf{V}_k^{(l)}, \widetilde{\mathbf{V}}_{\text{opt}}) \geq \mathcal{O}(\widetilde{\mathbf{V}}_{\text{opt}})$. It follows that setting $\mathbf{V}_k^{(l)}$ to $\widetilde{\mathbf{V}}_{\text{opt}}$ monotonically decreases the objective function $\mathcal{O}$ which is exactly the update rule in Theorem 1. □

1.2 Omitted Formulas for Inference Speed-Up Methods

The first term in the denominator of Eq. (5) in the main file, $\mathbf{V}_k^{(l)} \mathcal{I}_{(k)}^{(l)} [\otimes_{i=1}^{o(l)\backslash k} \mathbf{V}_i^{(l)}]^\top [\otimes_{i=1}^{o(l)\backslash k} \mathbf{V}_i^{(l)}] \mathcal{I}_{(k)}^{(l)\top}$, again involves matrix multiplication of the huge dense matrix $[\otimes_{i=1}^{o(l)\backslash k} \mathbf{V}_i^{(l)}] \mathcal{I}_{(k)}^{(l)\top}$. Leveraging the composition of $[\otimes_{i=1}^{o(l)\backslash k} \mathbf{V}_i^{(l)}] \mathcal{I}_{(k)}^{(l)\top}$, one can show that

$$\mathbf{V}_k^{(l)} \mathcal{I}_{(k)}^{(l)} [\otimes_{i=1}^{o(l)\backslash k} \mathbf{V}_i^{(l)}]^\top [\otimes_{i=1}^{o(l)\backslash k} \mathbf{V}_i^{(l)}] \mathcal{I}_{(k)}^{(l)\top} = \mathbf{V}_k^{(l)} \prod_{\substack{i=1 \\ i \neq k}}^{o(l)} \left(\mathbf{V}_i^{(l)\top} \mathbf{V}_i^{(l)} \right).$$

where $\prod$ is Hadamard product of a sequence. As such, instead of multiplying a huge dense matrix, one may only compute Hadamard product and matrix multiplication over a few relatively small matrices. Note that in the example provided in Section 6 of the main file, $[\otimes_{i=1}^{o(l)\backslash k} \mathbf{V}_i^{(l)}] \mathcal{I}_{(k)}^{(l)\top}$ has 10^{20} entries, while $\mathbf{V}_i^{(l)}$ has only $10000 \times 10 = 10^5$ entries and $o(l) = 5$.

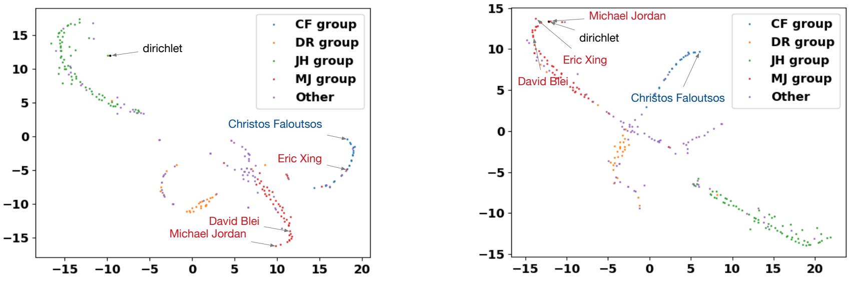
Lastly, evaluating the loss function Eq. (3) in the main file for determining convergence involves the computation of the Frobenius norm of its first term, *i.e.*, $\mathcal{X}^{(m)} - \mathcal{I}^{(m)} \times_{i=1}^{o(m)} \mathbf{V}_i^{(m)}$, which is a huge, dense tensor. Again by exploiting the desirable sparsity property of $\mathcal{X}^{(m)}$, we can calculate the Frobenius norm of $\mathcal{X}^{(m)} - \mathcal{I}^{(m)} \times_{i=1}^{o(m)} \mathbf{V}_i^{(m)}$ as follows

$$\begin{aligned}
&\left\| \mathcal{X}^{(m)} - \mathcal{I}^{(m)} \times_{i=1}^{o(m)} \mathbf{V}_i^{(m)} \right\|_F^2 \\
=& \left\| \mathcal{X}^{(m)} \right\|_F^2 - 2 \left\| \mathcal{X}^{(m)} \circ \mathcal{I}^{(m)} \times_{i=1}^{o(m)} \mathbf{V}_i^{(m)} \right\|_1 + \left\| \mathcal{I}^{(m)} \times_{i=1}^{o(m)} \mathbf{V}_i^{(m)} \right\|_F^2 \\
=& \left\| \mathcal{X}^{(m)} \right\|_F^2 - 2 \sum_{j_1,\ldots,j_o(m)} (\mathcal{X}^{(m)})_{j_1,\ldots,j_{o(m)}} \sum_{c=1}^{C} \prod_{i=1}^{o(m)} (\mathbf{V}_i^{(m)})_{j_i,c} \\
&+ \sum_{c_1=1}^{C} \sum_{c_2=1}^{C} \prod_{i=1}^{o(m)} (\mathbf{V}_i^{(m)})_{:,c_1}^\top (\mathbf{V}_i^{(m)})_{:,c_2}.
\end{aligned} \tag{1}$$

This equivalency transforms the computation of a dense and potentially high-order tensor to that of a sparse tensor accompanied by a couple of matrix manipulation. The complexity of the first and the second term in the above formula are $O(\mathrm{nnz}(\mathcal{X}^{(m)}))$ and $O(C \cdot o(m) \cdot \mathrm{nnz}(\mathcal{X}^{(m)}))$, respectively, thanks to the sparsity of $\mathcal{X}^{(m)}$. With the complexity of the third term being $O(C^2 \cdot \sum_{i=1}^{o(m)} |\mathcal{V}_{\varphi(m,i)}|)$, the overall complexity is reduced from $O(\prod_{i=1}^{o(m)} |\mathcal{V}_{\varphi(m,i)}|)$ to $O(C \cdot o(m) \cdot \mathrm{nnz}(\mathcal{X}^{(m)}) + C^2 \cdot \sum_{i=1}^{o(m)} |\mathcal{V}_{\varphi(m,i)}|)$. That is, considering the previous example, the complexity of evaluating this Frobenius norm would decrease from a magnitude of 10^{20} to a magnitude of 10^8.

It is worth noting that the trick introduced in the last equivalency, Eq. (1), has already been proposed in the study of Matricized Tensor Times Khatri-Rao Product (MTTKRP) [1, 6, 18]. MTTKRP and our model share a similarity in this trick because, unlike update rule Eq. (5) in the main file, evaluating the loss function Eq. (3) in the main file does not involve the non-negative constraints.

2 Visualization of the Ablation Study

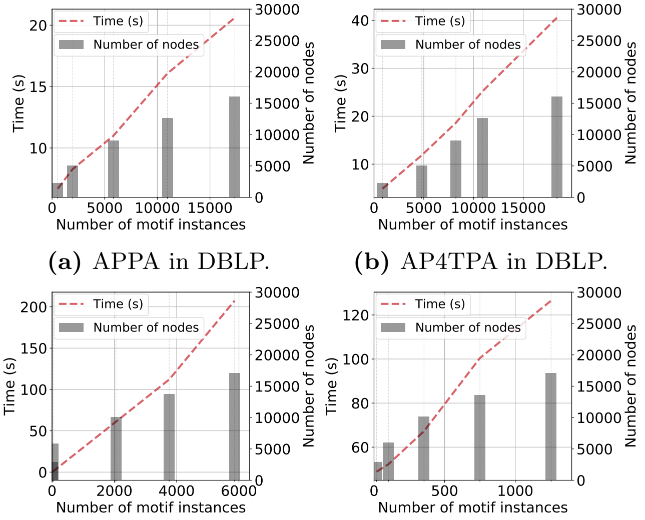


(a) W/o both.

(b) Full model

Fig. 1: Visualization of the ablation study where the author nodes are color-coded according to the truth label.

To better understand the impact of candidate motif choice discussed in Section 7.3of the main file, we further visualized the inferred membership of each node in Figure 1 by projecting its corresponding column in the consensus matrix $\mathbf{V}_t^*$ using t-Distributed Stochastic Neighbor Embedding (t-SNE). As discussed in Section 4, *dirichlet* reflects a distinctive facet of the relationship between *Xing* and *Blei* pertaining to their graduating group. The full model containing $AP4TPA$ inferred all of them to be close under the user guidance concerning research group. In contrast, the partial model with only edge-level motifs not only mistakenly assigned *Xing* to *Faloutsos*'s group but also learned *dirichlet* to be far away from either Xing or Blei. This observation echos the intuition discussed in Section 4that modeling higher-order interaction can introduce a richer pool



(a) APPA in DBLP. **(b)** AP4TPA in DBLP.

(c) 2P2W in YAGO. **(d)** 3PW in YAGO.

Fig. 2: Wall-clock runtime for inferring all parameters of one motif and the number of nodes against the number of motif instances in a series of downsampled HINs. The proposed algorithm empirically achieves near-linear efficiency, and motif instances are indeed sparse in HINs.

of signals, and such modeling should be comprehensive and fine-grained in the task of user-guided clustering.

3 Efficiency Study

In this section, we empirically evaluate the efficiency of the proposed algorithm with a focus on the speed-up tricks described in Section 6 of the main file. Specifically, we estimate the runtime for inferring all parameters involved in one motif while all other parameters are fixed, or equivalently, reaching convergence of the while-loop from line 4 to line 6 in Algorithm 1 in the main file.

This study was conducted on both the DBLP dataset and the YAGO dataset for each of their respective non–edge-level motifs: APPA and AP4TPA in DBLP; 2P2W and 3PW in YAGO. The non–edge-level motifs are studied because (i) they are more complex in nature and (ii) the tensors induced by edge-level motifs are essentially matrices, the study of which degenerates to the well-studied case of non-negative matrix factorization. To downsample the HINs, we randomly knock out a portion of papers in DBLP or persons in YAGO. The involved edges and the nodes that become dangling after the knock-out are also removed from the network. The reason node type paper and person are used is that they are associated with the most diverse edge types in DBLP and YAGO, respectively.

In the end, we obtain a series of HINs with 10%, 25%, 50%. 75%, 100% of papers or persons left.

To more accurately evaluate the efficiency of the proposed algorithm in this study, we turn off the parallelization in our implementation and use only one thread. We record the wall-clock runtime for inferring all parameters of each concerned motif, $\{\mathbf{V}_k^{(l)}\}_{k=1}^{o(l)}$, while fixing the motif weights $\boldsymbol{\mu}$ and parameters of other motifs, $\{\mathbf{V}_i^{(m)}\}_{m\neq l}$. The experiment is executed on a machine with Intel(R) Xeon(R) CPU E5-2680 v2 @ 2.80GHz. The result is reported in Figure 2.

The proposed algorithm empirically achieves near-linear efficiency in inferring parameters of each given motif. As presented in Figure 2, the runtime for all motifs on both datasets are approximately linear to the number of involved motif instances. This result is in line with the analysis provided in Section 6 of the main file and justifies the effectiveness of the speed-up tricks.

Moreover, we also reported the number of motif instances against the number of nodes regardless of type in each downsampled network. For all four studied motifs, we do observe motif instances are sparse and do not explode quickly as the size of the network increases.

4 Detailed Description of Datasets, Evaluation Tasks, and Evaluation Metrics

In this section, we provide the detailed description of the datasets, evaluation tasks, and metrics used in the experiments. **Datasets.** We use two real-world HINs for experiments.

- **DBLP** is a heterogeneous information network that serves as a bibliography of research in computer science area [21]. The network consists of 5 types of node: author (A), paper (P), key term (T), venue (V) and year (Y). The key terms are extracted and released by Chen et al. [5]. The edge types include authorship, term usage, venue published, year published, and the reference relationship. The first four edge types are undirected, and the last one is directed. The schema of the DBLP network is shown in Figure 2a in the main file. In DBLP, we select two candidate motifs for all applicable methods, including $AP4TPA$ and $APPA$, where $APPA$ is also a meta-path representing author writes a paper that refers another paper written by another author and $AP4TPA$ was introduced in Section 3 of the main file.
- **YAGO** is a knowledge graph constructed by merging Wikipedia, GeoNames and WordNet. YAGO dataset consists of 7 types of nodes: person (P), organization (O), location (L), prize (R), work (W), position (S) and event (E). There are 24 types of edges in the network, with 19 undirected edge types and 5 directed edge types as shown by the schema of the YAGO network in Figure 3. In YAGO, the candidate motifs used by all compared methods include $P^6O^{23}L$, $P^7O^{23}L$, $P^8O^{23}L$, $2P2W$, $3PW$, where the first three are

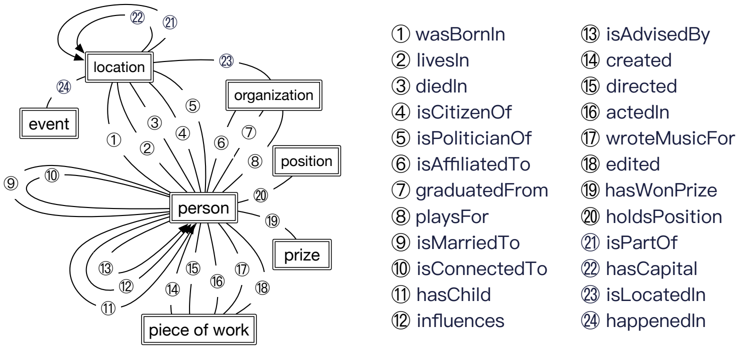


Fig. 3: The schema of YAGO [17].

also meta-paths with the number in superscript being type of edge given in Figure 3. $2P2W$ is the motif that 2 people simultaneously co-created (edge type 14) two pieces of work, and $3PW$ is the motif that 3 people who created (edge type 14), directed (edge type 15) and acted (edge type 16) in a piece of work, respectively.

Evaluation tasks. In order to validate the proposed model's capability in reflecting different guidance given by different users, we use two sets of labels on authors to conduct two tasks in DBLP similar to previous study [19]. Additionally, we design another task on YAGO with labels on persons. We provide datasets and labels used in the experiment along with the submission. **DBLP-group** – Clustering authors to 5 research groups where they graduated, which is an expanded label set from the "four-group dataset" [19]. The "four-group dataset" includes researchers from four renowned research groups led by Christos Faloutsos, Michael I. Jordan, Jiawei Han, and Dan Roth. Additionally, we add another group of researchers, who have collaborated with at least one of the researchers in the "four-group dataset" and label them as the fifth group with the intention to involve more subtle semantics in the original HIN. 5% of the 250 authors with labels are randomly selected as seeds from user guidance. We did not use 1% for seed ratio as in the following two tasks because the number of authors to be clustered in this task is small. The resulted HIN processed as such consists of 19,500 nodes and 108,500 edges. **DBLP-area** – Clustering authors to 14 research areas, which is expanded from the "four-area dataset" [19], where the definition of the 14 areas is derived from the Wikipedia page: List of computer science conferences[1]. 1% of the $7,165$ authors with labels are randomly selected as seeds from user guidance. The HIN processed in this way has 16,100 nodes and 30,239 edges. **YAGO** – Clustering people to 10 popular countries in the YAGO dataset. We knock out all edges with edge type wasBornIn, and if a person had an edge with one of the 10 countries, we assign this country to be the label of this person. Additionally, to avoid making our task trivial, we remove

[1] https://en.wikipedia.org/wiki/List of computer science conferences

all other types of edges between person and location. 1% of the $11,368$ people are randomly selected as seeds from user guidance. There are 17,109 nodes and 70,251 edges in the processed HIN.

Evaluation metrics. We use three metrics to evaluate the quality of the clustering results generated by each model: Accuracy (Micro-F1), Macro-F1, and NMI. **Accuracy** refers to a measure of statistical bias. More precisely it is defined by the division of the number of correctly labeled data by the total size of the dataset. Note that in multi-class classification tasks, accuracy is always identical to Micro-F1. **Macro-F1** refers to the arithmetic mean of the F1 score across all different labels in the dataset, where the F1 score is the harmonic mean of precision and recall for a specific label. **NMI** is the abbreviation for normalized mutual information. Numerically, it is defined as the division of mutual information by the arithmetic mean of the entropy of each label in the data. For all these metrics, higher values indicate better performance.

5 Related Work on Matrix and Tensor Factorization for Clustering.

By factorizing edges that represent pairwise interactions in a network, matrix factorization has been shown to be able to reveal the underlying composition of objects [11]. In this direction, a large body of study has been carried out on clustering networks using non-negative matrix factorization (NMF) [7, 12, 13]. As a natural extension beyond pairwise interaction, tensor has been used to model interaction among multiple objects for decades [9, 22]. A wide range of applications have also been discussed in the field of data mining and machine learning [10, 14].

For the study of clustering and related issues, many algorithms have been developed for homogeneous networks by factorizing a single tensor [2–4, 15, 16]. A line of work transforms a network to a 3-rd order tensor via triangles, which is essentially one specific type of network motif [2, 16]. Researchers have also explored weak supervision in guiding tensor factorization based analysis [3]. A large number of non-negative tensor factorization methods have been proposed for practical problems in computer vision [15]. Besides, tensor-based approximation algorithms for clustering also exist in the literature [4, 20]. One recent work on local network clustering considering higher-order conductance shares our intuition since it operates on tensor transcribed by a motif without decomposing into pairwise interactions [24]. This method is designed for the scenario where one motif is given. Different from the approach proposed in our paper, all the above methods are not designed for heterogeneous information networks, where the use of multiple motifs is usually necessary to reflect the rich semantics in HINs. Finally, we remark that to the best of our knowledge existing tensor-based clustering methods for HINs [8, 23] either do not jointly model multiple motifs or would essentially decompose the higher-order interactions into pairwise interactions.

References

1. Bader, B.W., Kolda, T.G.: Efficient matlab computations with sparse and factored tensors. SIAM Journal on Scientific Computing **30**(1), 205–231 (2007)
2. Benson, A.R., Gleich, D.F., Leskovec, J.: Tensor spectral clustering for partitioning higher-order network structures. In: Proceedings of the 2015 SIAM International Conference on Data Mining. pp. 118–126. SIAM (2015)
3. Cao, B., Lu, C.T., Wei, X., Philip, S.Y., Leow, A.D.: Semi-supervised tensor factorization for brain network analysis. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 17–32. Springer (2016)
4. Cao, X., Wei, X., Han, Y., Lin, D.: Robust face clustering via tensor decomposition. IEEE transactions on cybernetics **45**(11), 2546–2557 (2015)
5. Chen, T., Sun, Y.: Task-guided and path-augmented heterogeneous network embedding for author identification. In: WSDM. ACM (2017)
6. Choi, J.H., Vishwanathan, S.: Dfacto: Distributed factorization of tensors. In: Advances in Neural Information Processing Systems. pp. 1296–1304 (2014)
7. Ding, C., Li, T., Peng, W., Park, H.: Orthogonal nonnegative matrix t-factorizations for clustering. In: KDD (2006)
8. Gujral, E., Papalexakis, E.E.: Smacd: Semi-supervised multi-aspect community detection. In: ICDM (2018)
9. Harshman, R.A.: Foundations of the parafac procedure: Models and conditions for an" explanatory" multimodal factor analysis (1970)
10. Kolda, T.G., Bader, B.W.: Tensor decompositions and applications. SIAM review **51**(3), 455–500 (2009)
11. Lee, D.D., Seung, H.S.: Learning the parts of objects by non-negative matrix factorization. Nature **401**(6755), 788 (1999)
12. Lee, D.D., Seung, H.S.: Algorithms for non-negative matrix factorization. In: Advances in neural information processing systems. pp. 556–562 (2001)
13. Liu, J., Wang, C., Gao, J., Han, J.: Multi-view clustering via joint nonnegative matrix factorization. In: SDM. vol. 13, pp. 252–260. SIAM (2013)
14. Papalexakis, E.E., Faloutsos, C., Sidiropoulos, N.D.: Tensors for data mining and data fusion: Models, applications, and scalable algorithms. TIST **8**(2), 16 (2017)
15. Shashua, A., Hazan, T.: Non-negative tensor factorization with applications to statistics and computer vision. In: ICML (2005)
16. Sheikholeslami, F., Baingana, B., Giannakis, G.B., Sidiropoulos, N.D.: Egonet tensor decomposition for community identification. In: Signal and Information Processing (GlobalSIP), 2016 IEEE Global Conference on. pp. 341–345. IEEE (2016)
17. Shi, Y., Zhu, Q., Guo, F., Zhang, C., Han, J.: Easing embedding learning by comprehensive transcription of heterogeneous information networks. In: KDD (2018)
18. Smith, S., Ravindran, N., Sidiropoulos, N.D., Karypis, G.: Splatt: Efficient and parallel sparse tensor-matrix multiplication. In: Parallel and Distributed Processing Symposium (IPDPS), 2015 IEEE International. pp. 61–70. IEEE (2015)
19. Sun, Y., Norick, B., Han, J., Yan, X., Yu, P.S., Yu, X.: Integrating meta-path selection with user-guided object clustering in heterogeneous information networks. In: KDD (2012)
20. Sutskever, I., Tenenbaum, J.B., Salakhutdinov, R.R.: Modelling relational data using bayesian clustered tensor factorization. In: Advances in neural information processing systems. pp. 1821–1828 (2009)
21. Tang, J., Zhang, J., Yao, L., Li, J., Zhang, L., Su, Z.: Arnetminer: extraction and mining of academic social networks. In: KDD (2008)

22. Tucker, L.R.: Some mathematical notes on three-mode factor analysis. Psychometrika **31**(3), 279–311 (1966)
23. Wu, J., Wang, Z., Wu, Y., Liu, L., Deng, S., Huang, H.: A tensor cp decomposition method for clustering heterogeneous information networks via stochastic gradient descent algorithms. Scientific Programming **2017** (2017)
24. Zhou, D., Zhang, S., Yildirim, M.Y., Alcorn, S., Tong, H., Davulcu, H., He, J.: A local algorithm for structure-preserving graph cut. In: KDD (2017)