# GraphLDA: Latent Dirichlet Allocation-based Visual Exploration of Dynamic Graphs

Category: Research

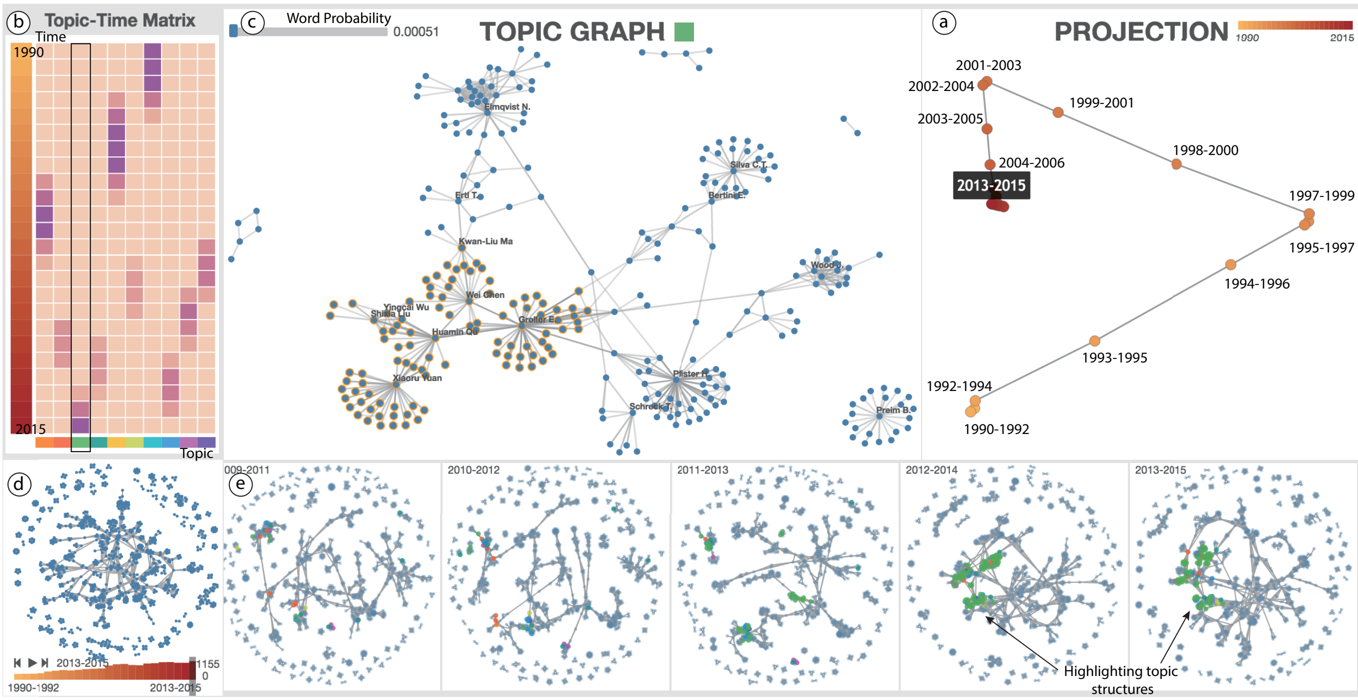


Figure 1: The interface of our system: (a) Projection View, visualizing the overview of the similarity relation between graphs of different time steps; (b) Topic-Time Matrix View, showing probability distributions of extracted topics (structures) in graphs of different time steps; (c) Topic View, showing the detailed information of extracted topics (structures); (d) Animation View, playing the animation to show the evolution of the dynamic graph; (e) Small Multiples View, showing the snapshots of raw graphs in the context of extracted topics (structures) in different time steps.


## Abstract

In dynamic graph visualization and analysis, it is very challenging to visualize both the overall evolution of trends and the detailed changes of structures simultaneously. In this work, we propose a latent Dirichlet allocation (LDA) -based visual exploration method for dynamic graphs. With the LDA-based analysis, we can reveal important structures in the dynamic graph based on the extracted semantic topics. To gain a deeper understanding of the derived structures and their evolution, we propose a visual analytics pipeline enabling users to interpret and explore the dynamic graph. To experiment with the proposed method, we provide a visual analytics system to test with real-world data. Our cases on the datasets of dynamic collaboration network and social communication network have demonstrated the effectiveness of the proposed method.


**Keywords:** Dynamic Graph, Latent Dirichlet Allocation, Graph Structure, Evolution

## 1 Introduction

Graphs are ubiquitous in our world, which are commonly employed to describe relationships between entities. Graphs are featured relationships in a wide range of scenarios, such as the social network in Twitter, financial transactions, biochemistry, software engineering, etc. As the world is evolving, dynamic graphs depict relationships in our world more vividly and reliably. The evolution of the graph is reflected by changes of nodes, links, and structures along the time. By studying such evolution of dynamic graph, we can get a better understanding of how people relationships change over time, how certain information spreads within social networks, etc. However, analyzing the evolution in dynamic graphs is a big challenge due to the complexity of topological and temporal information.

Researchers have contributed great efforts in visualization and analysis of dynamic graphs. However, this problem is still far from fully solved. Animation and small multiples are two basic approaches directly visualizing the evolution of graph structures [4]. Both of them are difficult for users to relate and compare graphs of different time steps when the number of time steps becomes large. More sophisticated analysis methods are proposed for dynamic graphs in recent years, such as dimension reduction techniques and clustering approaches [23, 36, 38]. Although they already have a good emphasis on the overview of the evolution, they can not directly tell how these evolutions happen. These methods require users to manually check the original data to find reasons of the similarity, which is infeasible when the graphs are large or within a large time range. Therefore, dynamic graph analysis urgently requires the capability to reveal the patterns of evolution, which could be used to reflect details about the overall trends.

In this work, we proposed a latent Dirichlet allocation (LDA)-

based visual exploration method, to reveal both the overall patterns, such as stable patterns, recurring patterns and anomalies, and the evolution of main structures of dynamic graphs. Our method regards each link as one *word* and the graph in each time step as a *document*. Main structures could be derived as LDA topics, which involve the temporal patterns in the dynamic graph. To further analyzing the derived *topics*, we propose a visual analytics system to enable users to explore the evolutions of the main structures of the dynamic graph. Our method can not only tell what the overall patterns are, but also give reasons why such patterns happen based on the identified main structures. Specifically, our contribution is listed as followed.

- Proposing a novel technique to adapt LDA model to detect the main structures in the dynamic graph.
- Analyzing both the evolution of the dynamic graph and the changes of its main structures simultaneously, reasoning the patterns and similarity of graphs in different time steps.
- Providing a visual analytics system to facilitate the analysis of the results of our method, enabling the iterative exploration of the dynamic patterns.

This paper is structured as follows. In Section 2, we review existing works in related topics. In Section 3, we present the technique details of the LDA model as a background. We detail our motivation, design rationales, LDA settings and analysis pipeline for dynamic graph analysis in Section 4. Visual analytics workflow and technique details are presented in Section 5. We demonstrate the usage of our method in Section 6. Finally, we discuss the limitations of our method and future work.

## 2 Related Works

We review existing works that are most related to our method, including techniques developed for (1) dynamic graph visualization, (2) dynamic graph visual analysis and (3) topic model and LDA-based analysis.

### 2.1 Dynamic Graph Visualization

To directly visualize the evolution of dynamic graphs, there are mainly two kinds of approaches, i.e. animation and small multiples. Beck et al. [4] propose a detailed classification and survey on this topic. The animation technique shows a sequence of graphs orderly like a movie. There are continuous research works using animation to explore dynamic graphs [14, 2]. One important challenge in designing animation is how to preserve users' mental map. To solve the problem, Diehl et al. [12] compute a global layout which makes the positions of nodes in each time step more stable. GraphAEL [17] also aggregates a sequence of graphs to calculate a consistent layout. Different from the previous works, equivalent nodes in different time steps are linked by virtual edges to preserve the stable layout. To better preserve the stability, Gorochowski et al. [21] introduce an age-directed approach to adaptively changing the position of nodes. When updating the graph layout, they not only consider the local changes of positions of nodes, but also involve their positions in the last layout. Che et al. [9] use a Laplacian constrained distance embedding method to maintain overall structures of a sequence of graphs.

Besides maintaining the consistent layout, researchers also investigate the transition techniques between consecutive steps of the dynamic graph [2, 20]. For example, Bach et al. [2] use staged transitions and specifically highlight changes in the graph between time steps.

Small multiples technique shows the complete sequence of graphs using static snapshots with juxtaposition [8], superimposition [13, 15], or the integration of these two methods [35, 37]. Small multiples provide an overview of the dynamic graph and suitable for analytical tasks. However, when the sequence is long, it is hard to show all graphs because the screen space is limited. There are research works focusing on addressing the challenge of displaying long-sequence graphs [22, 37].

To take the advantage of both animation and small multiples, Beck et al. [5] combine the two approaches to show long sequences of graphs based on Parallel Edge Splatting. To better help users keep the mental map, DiffAni [34] integrates animation and difference maps to generate semantic-meaningful small multiples.

Although existing works have employed small multiples and animations for better context map preservations in the dynamic graph, none of them are developed for visualizing the evolution of the dynamic graph and changes of detected main structures simultaneously with animation and small multiples techniques. Our paper is focusing on this issue.

### 2.2 Dynamic Graph Visual Analysis

Visual analysis methods for dynamic graphs can be divided into two main categories, i.e. similarity exploration and community analysis. In similarity exploration, researchers focus on the topological information [42] as well as the multi-variate information [23] of the dynamic graph. Von Landesberger et al. [42] employ Multidimensional Scaling (MDS) algorithm when analyzing the spreading patterns of contagion. Besides the topological similarity exploration, Hadlak et al. [23] group nodes and links based on their associated attributes which are changing over time. Based on the similarities, Steiger et al. [36] propose visual comparison methods to find anomaly situation in the dynamic graph. Existing works are successful techniques in illustrating the similarity in the dynamic graph. However, none of them can detect multiple representative structures across multiple time steps, and compare similarities of graphs based on these structures. More recently, van den Elzen et al. [38] propose methods of reducing graphs in time steps to points. They treat the graph in each time step as a high-dimensional vector, and then uses dimension reduction method to effectively show the similarity of graphs in the two-dimensional space for comparison. However, their method only shows an overview of the evolution of the dynamic graph, but can not give reasons why they are similar or different, which is the challenge we want to tackle in this work.

Community detection is one important direction in analyzing dynamic graphs. A community is defined as a group of nodes with strong inner-connections and weak inter-connections. Falkowski et al. [16] detect and show communities over time without analyzing the evolution of the dynamic graph. Small multiples are frequently used in the community detection in dynamic network [39, 40]. Vehlow et al. [40] detect the communities and show the evolution with Sankey diagram. Further to investigate the hierarchical structures of communities, they propose a series of connected matrices to visualize the split and merge of the hierarchical community structures [39]. Different from focusing on the community structure, our LDA-based method does not require a pre-definition of structures of interest, which is flexible and suitable for extraction of more general structures from dynamic graphs.

### 2.3 Topic Model and LDA-based Analysis

The topic model is widely used in text analysis to mine hidden semantic structures in a corpus of documents. It also provides a more reasonable way to measure the similarities between documents at the high level of semantics instead of the level of words. Landauer et al. [29] first propose the concept of Latent Semantic Analysis (LSA) in 1988. In LSA, a latent semantic layer is added between documents and words. Latent semantics are extracted from the relationship among words to construct semantic space, where documents are then projected to obtain a sparse representation for further analysis. Later, statistic analysis and generative model are introduced in pLSI/pLSA [25, 26] to solve synonyms and polysemy problem. Blei et al. [6] propose the concept of topic model and related LDA model. LDA is a multi-layer Bayesian model, including three layers, e.g. words, topics, and documents. Each topic is a mixture of words, while every document is a mixture of topics. By introducing the Dirichlet distribution, LDA model is able to avoid over-fitting which pLSA suffers. Afterwards, there are more studies on the variations of LDA [31].

In text visualization and visual analytics, LDA model is also widely used. To visualize the evolution of topics along the time, TIARA [45] encodes the hotness of topics using the width of rivers in ThemeRiver. TextFlow [11] further uses the metaphor of rivers to indicate the emerging, vanishing, merging and splitting events in topic models. LeadLine [43] is also a river-like visualization, but more emphasized on the bursting to topics hotness. iVisClustering [30] not only provides various visualization techniques for LDA model, but also enables users steering of LDA process. Besides applications in text analysis field, LDA model has been adopted in computer graphics and computer vision field [44] for various purposes, such as segmentation, classification, pattern recognition, etc. In the visualization of traffic data, Chu et al. [10] use LDA model to discover hidden themes from trajectories data. In flow fields analysis, Hong et al. [27] develop an LDA-based model to measure the similarities between fields lines in order to identify the meaningful flow structures.

In graph mining and analysis, there are several works applying LDA to analyze static graphs. Zhang et al. [46] propose to use LDA to find communities in large-scale social networks. Henderson and Eliassi-Rad [24] use LDA model to discover latent groups in large directed graphs. Different from these previous works, we focus on the dynamic graph analysis. In a later work from Henderson et al., they apply a dynamic version of LDA model to discover the group in dynamic graphs [32]. In this work, they treat each source node at each time step corresponds to a document at related time step and links from this node as words in the document. Different from their approach, we use a novel definition in the dynamic graph which is able to identify the evolution of main structures, which can not be achieved by their methods. Moreover, we integrate a visual analytics approach to facilitate the evolutionary patterns exploration. Besides, these works focus on how to employ the LDA model in graph analysis but ignore how to support users to explore results in details. From the visual analytics research perspective, to the best of our knowledge, there are no previous works employing the LDA technique to analyzing the evolution of main structures in dynamic graphs.

## 3 Background

To better gain an overview of the problem and our proposed method, we first give a brief introduction of the LDA model.

Latent Dirichlet allocation (LDA) was first proposed by Blei et al. [6] to explain why documents are similar from the latent topic level instead of the word level. The symbols used in this paper are listed in Table 1. Any document $d_j$ is modeled as a mixture of $K$ topics, while any topic $k$ is characterized by a multinomial distribution $\phi_k$ over vocabulary $\mathscr{V}$. Among all variables, only $w_{ij}$ is observable, while others like $z_{ij}$, $\theta_j$, and $\phi_k$ are latent variables. LDA model generates observations of latent variables using the following process:

1. For every document $d_j$ $(j \in \{1,\cdots,D\})$, a topic distribution $\theta_j$ is drawn from a Dirichlet prior with parameter $\alpha$.
2. For every topic $k$ $(k \in \{1,\cdots,K\})$, a word distribution $\phi_k$ is drawn from a Dirichlet prior with parameter $\beta$.
3. For word position $i$ in $j^{\text{th}}$ document $(i \in \{1,\cdots,N_j\}, j \in \{1,\cdots,D\})$, first choose a topic $z_{ij} = k$ from the topic distribution $Multinomial(\theta_j)$, and then choose a word $w_{ij}$ from the chosen topic with word distribution of $Multinomial(\phi_{z_{ij}})$.

| Symbol | LDA | GraphLDA |
|---|---|---|
| $D/T$ | $D$: Number of documents | $T$: Number of time steps |
| $K$ | Number of topics | |
| $\mathscr{V}$ | Vocabulary | The universal set of links |
| $d_j/G_j$ | $d_j$: the $j^{th}$ document | $G_j$: the $j^{th}$ Graph |
| $N_j$ | Number of words in $d_j$ | Number of edges in $G_j$ |
| $w_{ij}/e_{ij}$ | $w_{ij}$: the $i^{th}$ word in $d_j$ | $e_{ij}$: the $j^{th}$ link in $G_i$ |
| $z_{ij}$ | Topic assignment for word $w_{ij}$ / link $e_{ij}$ | |
| $\theta_j$ | Probability of topics in document $d_j$/graph $G_j$ | |
| $\Theta$ | Vector version of $\theta_j$ | |
| $\phi_k$ | Probability of words in topic $k$ | |
| $\Phi$ | Vector version of $\phi_j$ | |
| $\alpha$ | Dirichlet prior for $\theta$ | |
| $\beta$ | Dirichlet prior for $\phi$ | |

Table 1: Symbols used in this paper. Some symbols in GraphLDA follow the tradition in graph visualization community, but are listed with symbols of LDA model for correspondence.

After the generative process is defined, the total probability of the model is defined as:

$$\prod_{i=1}^{K} P(\phi_i;\beta) \prod_{j=1}^{M} P(\theta_j;\alpha) \prod_{t=1}^{N} P(Z_{j,t}|\theta_j) P(W_{j,t}|\phi_{Z_{j,t}}),$$

where $W, Z, \Theta, \Phi$ denotes the vector version of $w_{ij}, z_{ij}, \theta_j$, and $\phi_k$ respectively. We can use $P(W,Z,\Theta,\Phi;\alpha,\beta)$ to represent the likelihood function above. Then, the model estimation process is to maximize it by Bayesian inference with parameters $\alpha$ and $\beta$. The widely used Gibbs sampling implementaion [33] can accomplish one iteration in $O(KN)$ time complexity, where $N$ donates the total number of words in all documents.

The LDA model can be applied as a document clustering method. The $K$ topics could be treated as clusters, and the topic distribution $\theta_j$ for the document $d_j$ denotes the probabilities of membership to every cluster. A more careful approach is to treat $\theta_j$ as a lower-dimensional feature vector for every document, and conduct further analysis, such as dimension reduction. In our method, both usages are employed to provide a better understanding of dynamic graphs from different perspectives.

## 4 GraphLDA

In this section, we first discuss the analysis tasks and rationales for choosing the LDA approach. Afterwards, we detail our adaption of LDA model for the dynamic graph analysis. Finally, we describe our visual exploration pipeline illustrating how to integrate our method with the interactive visual interface.

### 4.1 Analysis Tasks

In dynamic graph analysis, it is critical to derive the evolutionary patterns with the changes of main structures of the graph. It is important to derive the main structure in each time step to tell the reasons why graphs are similar in different time steps. Specifically, our analysis tasks are as follows.

- **T1: Overall trends and similarity in different time steps.** Overall trends indicate the general evolutionary patterns of the dynamic graph. The trends are reflected by the similarities in the graph of different time steps.
- **T2: Evolution of main structures.** Besides the overall trends of dynamic graphs, we need to derive main structures in each

time step. The derived patterns should consider both structural and temporal information.

- **T3: Important entities and connections in main structures.** Important entities and connections play important roles in the evolution of main structures. It is important to identify the backbone of the graphs based on important entities and connections.

To support the analysis tasks, we need a proper method, which is able to reveal the insights of dynamic graph.

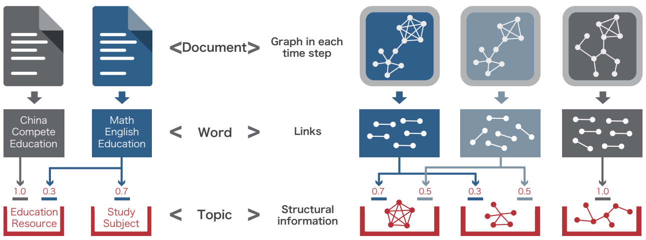


Figure 2: Illustration of our dynamic graph LDA model. In our method, we define a graph in each time step as a *document*, and a link connecting two nodes as a *word*. The derived *topics* could represent the structural information in the dynamic graph.

### 4.2 Rationales

There are several approaches to seek patterns, including classical clustering methods, the community detection methods and the LDA-based methods. First, we state why we do not choose the alternative methods to fulfill the tasks. Afterwards, we tell the reasons for applying the LDA for the dynamic graph analysis.

- **Classical clustering methods** It is a common approach to cluster different nodes in the graph based on their connections and attributes. Classical clustering techniques, such as K-Means, hierarchical clustering, DBSCAN, etc., define similarities from the Euclidean distances by treating nodes or links as high-dimensional points. However, the detected clusters in each time step are not consistent if we apply it in each time step separately. In the other aspect, clustering the super graph, containing the graphs in all the time steps, would lose the temporal information and it's not easy to reveal the evolutionary patterns. In short, it is difficult to summarize high-level features (i.e. evolutionary structures) directly from such low-level similarity definitions (i.e. nodes, links), for they have already been tightly aggregated in the clustering settings.

- **Community detection methods** A community is a set of nodes that have heavy inner connections. There are several state-of-the-art approaches to detect the dynamic communities in dynamic graphs. Fortunato and Hric have summarized related approaches in the work [18]. However, community detection method highly depends on the data only if the nodes forms communities in the graph. Moreover, the community is not the only structure we want to analyze in dynamic graphs. Other structures include backbones, strong connections or other meaningful features. These structures together contribute to the comprehensive descriptions of dynamic graphs.

In GraphLDA model, we not only identify similar graphs in different time steps, but also aggregate nodes and links into meaningful high-level structures.

These meaningful structures are called *topics*. With proper definitions of the document and words, the *topics* give a top-down way, i.e. from the aspects of the overall evolution, to understand the patterns, as well as a bottom-up way, i.e. from the aspects of nodes and links. This dual relationship is an important feature of LDA model, which can not be achieved by other classical clustering methods and community detection methods. In the other aspect, the LDA model can be recognized as a probabilistic clustering technique, which is fit for nature of dynamic graph data. A graph structure could be decomposed into multiple meaningful structures. It is usually hard to obtain clear boundaries between structures. In such situation, the probabilistic assignment could give more accurate information to describe multiple structures in dynamic graphs, compared to those rigid assignments in classical clustering techniques. Considering these points, we choose to employ LDA model to solve the challenging tasks of dynamic graphs. However, how to adapt the LDA model for the dynamic graph analysis is a critical problem. We report our strategies and parameter settings for the GraphLDA in the next section.

### 4.3 Adaption of LDA Model for Dynamic Graph Analysis

In this part, we would like to introduce how to employ LDA model in dynamic graph analysis.

There are three main entities in the LDA model - *documents*, *words*, and *topics*. It is critical to define proper correspondences using the entities of the dynamic graph. Since our main goal is to reveal the evolution of structures in the dynamic graph, we define the graphs in different time steps as *documents*, and links in graphs as the *words*, as shown in Figure 2. Then the *topics* in LDA model correspond certain structures in the dynamic graph, which could span across multiple time steps. A formal description of the correspondence between LDA model and our GraphLDA method is described as follows.

We model a dynamic graph $\Gamma$ as a sequence of graphs:

$$\Gamma = (G_1, G_2, ..., G_T), \tag{1}$$

where $T$ is the total number of time steps, and $G_j$ represents the graph in the $j^{th}$ time step. We directly define $G_j$ to correspond to the document $d_j$ in the LDA model.

For graph $G_j$, we denote the set of nodes and links as $(V_j, E_j)$. The universal set of nodes is $V = \bigcup V_j$. The universal set of links $\mathscr{V}$ can be represented as a Cartesian production $V \times V$, and for each graph $E_j \subseteq \mathscr{V}$. Then we define all links $e_{ij} \in E_j$ in graph $G_j$ as its *words*. We should note if a link connecting two nodes exists at different time steps, their corresponding graphs $G_j$ share a same $e_{ij}$. By this way, we are able to trace the evolution of graph structures along the time dimension.

There are alternatives for the word definitions in our adaption of LDA model. It is possible to define a word as a node, a link, a node with all its connected links, or a connected subgraph. After careful consideration, we decide to define a link in our method to be the equivalent of a word in the LDA model. The main reason is that the links are better in describing the topological information in graphs. If we define a node as a word, then only the density and distribution of nodes in the graph are reflected, but the connection information is lost. If we choose a node with all connected links as a word, there would be much more words but with fewer connections for each word. In the other way, choosing a connected subgraph would lead to a small number of words and lost details of the inner structure of the connected subgraphs. Considering all these facts, we finally made the decision to define links as words.

In addition, we would like to assign the weights of links to the frequencies of words in each document (graph $G_i$). The weight of a link $e_{ij}$ can be derived from its quantitative attributes, as follows:

$$w(e_{ij}) = a \times val(e_{ij}) + (1 - a) \times imp(e_{ij}) \tag{2}$$

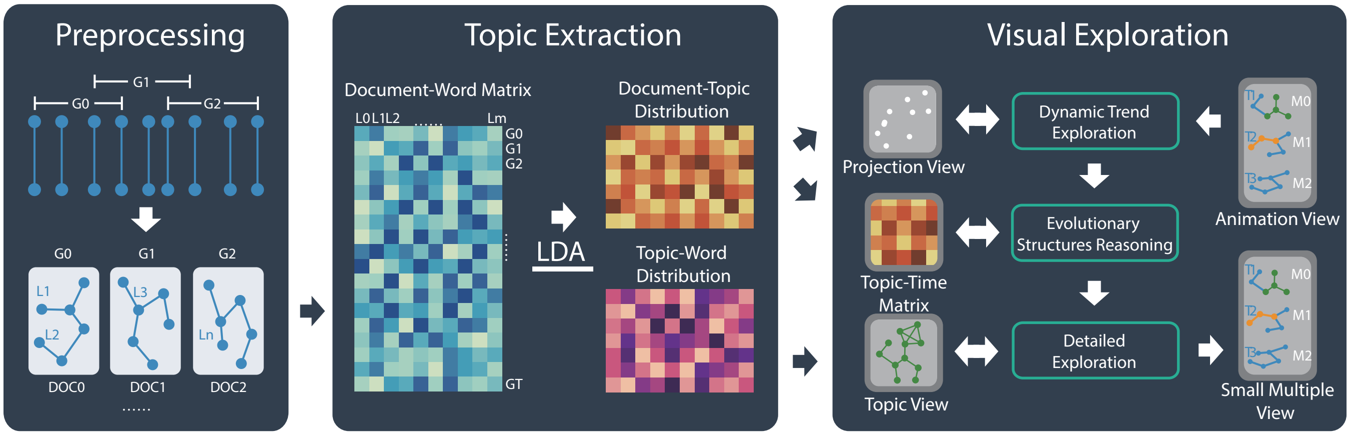


Figure 3: The visual analytics pipeline, which includes three parts - preprocessing, topic extraction and visual exploration. With the input of the document-word matrix, the LDA would output two parts of results - the document-topic distribution and the topic-word distribution. Users can iteratively explore the LDA results from overview to details within a visual analytics loop.

$val(e_{ij})$ can directly use attributes of the link in the original data, while the $imp(e_{ij})$ is derived from the topological information of the link in the graph $G_j$. For example, in the author collaboration network, we can define $val(e_{ij})$ as the times of collaboration between connecting two authors, and $imp(e_{ij})$ as the average of degrees of its connecting two nodes. We use $a$ to balance these two factors. We use 0.5 in our case.

With the correspondences defined above, we are able to represent the graph $G_j$ as a set of links $E_j = \{e_{1,j}, ..., e_{N_j,j}\}$ with frequencies $w(\cdot)$. Then, the whole dynamic graph is transformed to a matrix, where rows and columns represent graphs and links respectively, and the values of cells donate the frequencies. Such relationship is exactly fit the original document-word relationships in text analysis. We feed the graph-link matrix as the input of LDA model. The output contains a set of $K$ topics, probability distributions of topics in each graph $\theta_j$, and probability distributions of words in each topic $\phi_k$. We then donate $\Theta$ and $\Phi$ as the vector versions of $\theta_j$ and $\phi_k$.

The extracted topics are the meaningful structures of the input graph. We can understand the extracted topics from two perspectives. On one hand, graph $\Theta$ in each time step contains a topic distribution. We are able to compare graphs $G_{(\cdot)}$ at different time steps based on their topic distribution $\theta_j$. It provides a channel to explain the similarity of graphs in different time steps. On the other hand, each topic $k$ contains a word distribution $\phi_k$. We can use detailed information of nodes and links to investigate how the structures evolve along the time dimension. In short, the extracted topics acts important roles in our GraphLDA model, bridging the overview and details of the dynamic graph.

Due to the complexity of topics in the context of the dynamic graph, it is not easy to directly understand the extracted results, especially the temporal and topological relationship. To facilitate the understanding of evolution of extracted structures, we present a visual analytics system to integrate the GraphLDA model with the interactive exploration.

## 5 Visual Analytics System

The visual analytics workflow is shown in Figure 3. It contains three parts, including data preprocessing, topic extraction, and visual exploration. The raw dynamic graphs data are preprocessed through sliding window and transformed with our previous definitions. The graph-link relationships are fed into LDA model to extract topics. The output includes topic distributions in graphs $\Theta$ and link distributions in topics $\Phi$. In our visual analytics system, we provide several views to help users explore dynamic graphs with the LDA outputs. Projection View (Figure 1a) and Topic-Time Matrix View (Figure 1b) are used to analyze the topics from overall similarity and temporal attributes. Topic View (Figure 1c) provides the detailed information of nodes and links in the selected topic. To support drilling down to details, we provide Animation View (Figure 1d) and Small Multiples View (Figure 1e) to explore how structures of topics evolve within the original graphs.

### 5.1 Preprocessing and LDA Model

The preprocessing stage is to prepare the input for the LDA model. For the input dynamic graph, we use a time window with a width $\omega$ to get graphs within a specific time range. The sets of nodes and edges in the time range are aggregated. To avoid missing patterns due to hard boundaries, sliding windows are used and neightboring windows are overlapped with $\varepsilon$ time steps. Both the width of sliding window and overlapping can be adjusted by users with respect to input datasets. For the derived graphs using sliding windows, we derive the relationships between graphs and links with our method mentioned previously. Then we adopt the LDA algorithm with our input. Users can control the number of topics, and we also provide estimated values for different datasets. The output of the LDA contains two parts: probability distributions of topics in graphs $\Theta$, and probability distributions of words in topics $\Phi$. In the following sections, we introduce the exploration flow through the proposed views for analyzing these two parts.

### 5.2 Dynamic Trend Exploration

As an entrance of the analysis, we provide a Projection View (Figure 1a) to show the overall evolution of the dynamic graph (**T1**).

In the view, each point represents a graph in a time step, and two points are linked if they are in adjacent time steps. The position is determined by the dimension reduction method. Dimension reduction is used to analyze the similarity of entities in high-dimensional space. Therefore, in Projection View, close points indicate that related graphs are similar. Their effectiveness in the exploration of dynamic graphs and time series data has been demonstrated [38, 3]. Different from previous works, we project the graphs to points into two-dimensional space based on the similarity of topic distribution in each graph $\theta_j$. It emphasizes the similarity of the topological structures, instead of the similarity of raw node-link graphs. We provide users with multiple choices for the dimension reduction, including linear projection method Principal Components Analysis (PCA) and Multi-dimensional Scaling (MDS), and non-linear projection method t-Distributed Stochastic Neighbor Embedding (t-SNE).

In the Projection View, if several graphs are very similar, related points in two-dimensional space would be overlapped which could obstruct users' identification. Therefore, we re-position points with the same position using the Phyllotactic arrangement technique [41], which is also used in the work [38].

In addition, in order to support users to further investigate corresponding graphs of different time steps, we provide the lasso function. Using the lasso, uses can select a group of points to check their topic distributions in Topic-Time Matrix View (Figure 1b), and analyze in which topics they are similar.

### 5.3 Reasoning about Evolution of Structures

In this stage, users are supported to explore why graphs in different time steps are similar or dissimilar. To reason about such similarity and dissimilarity, we provide users two views - Topic-Time Matrix View (Figure 1b) and Topic View (Figure 1c). Topic-Time Matrix View presents the probability distribution of topics in graphs $\Theta$. In the view, each column represents a topic $k$ and each row is the graph $G_j$ in $j_{th}$ time step. In order to show the evolution along the time dimension, rows are arranged by time order. The opacity of cells encodes the probability of the $i_{th}$ topic in the graph $G_j$, i.e. $\theta_{ij}$. Darker cells indicate higher probabilities, and vice versa. With the Topic-Time Matrix View, we can find dominated topics in each time step (Figure 1b) as well as multiple topics contribute at the same time (Figure 7d - Day 4). Therefore, with this view, we are able to analyze the evolution trends of main structures in graphs along time, showing more details of the similarities in the Projection View.

By selecting one topic, users can investigate its detailed structural information, i.e. nodes and links contained, in the Topic View (Figure 1c). The structure to be visualized is derived from the probability distribution of links and connecting nodes in the selected topic. Node-link diagram is directly used to render the structures, and the probabilities of links are encoded using opacity. The opacity of links encodes the probability of the $i_{th}$ word in the topic $k$, i.e. $\phi_{ki}$. The links with high probabilities indicate important and frequent connections in this topic. We also provide the detailed information, i.e. node name, classes, etc on demand. Users are allowed to filter out links with low probabilities and related nodes with low probability by setting a threshold. Thus, users can easily perceive the main structure in each topic. Besides, users are supported to lasso a group of nodes in Topic View and check their temporal evolution using Small Multiples View (Figure 1e).

The exploration step fits the analytics tasks we proposed previously (**T2, T3**). We derive the topics, i.e. graph structures, to show the global trend and its changes along time. We are able to know in which time steps graphs are similar, and also to tell what exact structures they share and how they change with time.

### 5.4 Detailed Exploration

In the detailed exploration, we support users to drill down to investigate how exact structures evolves along time. We aim to show the appearance and/or the removal of nodes and links. We provide the Animation View (Figure 1d) and the Small Multiples View (Figure 1e) for this purpose (**T3**).

In the Animation View, we take two strategies to preserve users' mental map. On the one hand, we initialize the positions of nodes at each time step by their layout in the last one using the following equation:

$$p'_{t+1}(x,y) = b \times p'_t(x,y) + (1-b) \times p_{t+1}(x,y), b \subseteq [0,1], \quad (3)$$

where $b$ is a parameter indicating the importance of the previous layout. On the other hand, we employ five stages to transform the layout of the current time step to the next one, including removing links, removing nodes, updating positions of nodes, adding new nodes, and adding new links (Figure 4). The effectiveness of such staged transition has been demonstrated in the work of Marey [19, 20] and GraphDiaries [2].

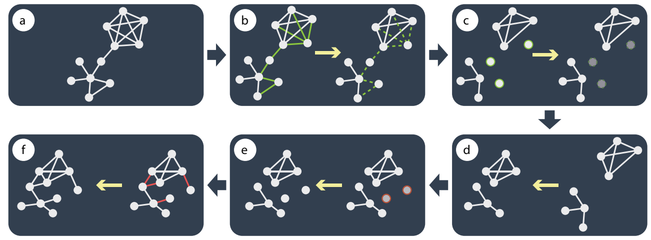


Figure 4: Five stages in animation transition from the layout of the current time step $t$ to the next one $t+1$. (a) The graph layout at the time step $t$; (b) Highlight and fade out links to be removed; (c) Highlight and fade out nodes to be removed; (d) Update nodes and links; (e) Highlight and fade in nodes to added; (f) Highlight and fade in links to be added, and finally get the layout at the time step $t+1$.

The animation requires users to memorize graphs in time steps, which raises cognitive burden for graph comparison. To compensate with this, we provide the Small Multiples View to compare the structures of topics in multiple graphs of different time steps (Figure 5). We use different colors to represent different topics. Users can perceive certain evolutionary patterns, such as emergence, disappearance, morphing, etc., of structures in different time periods. Those recurring structures can also be observed, as shown in our case study (Figure 7).

When users find patterns and specific topics of interest in the detail view, they can highlight them in the overview and explore the related graphs sharing the patterns. They can also select the topics in other time periods for continuous exploration. Through such iterative process, users can finally understand the structural evolution of the dynamic graph.

In summary, users can explore the results of our LDA-based method from the overview to details. Overall trends and similarity of graphs in different time steps are visualized in the Projection View based on dimension reduction (**T1**). For the evolution of main structures, it is showed in the Topic-Time Matrix View (**T2**). At the same, Topic View is provided for users to explore detail structure in each topic (**T2, T3**). With the filter interaction, users are allowed to identify important nodes and connections in Small Multiples View and Animation View (**T3**).

### 5.5 System Implementation

Our system is built upon three components: a data processing component, a server component, and the web interface. The data processing component is implemented for data handling and transmission. We build up a Python framework, supporting the sliding window, aggregation, and LDA calculation. The server component is also built with Python, using a Tornado Web Framework, supporting the connection of data processing and web data fetching. The front-end is built up with d3.js [7] within HTML5, using the BackBone and Require.js framework.

## 6 Case Studies

In this section, we provide two case studies with real-world dynamic graphs. One case is using the coauthor graph, while the other one is the social communication graph.

### 6.1 Case 1: Dynamic Collaboration Network

The dynamic collaboration network we used here contains papers published in IEEE VIS (previously named VisWeek) conferences

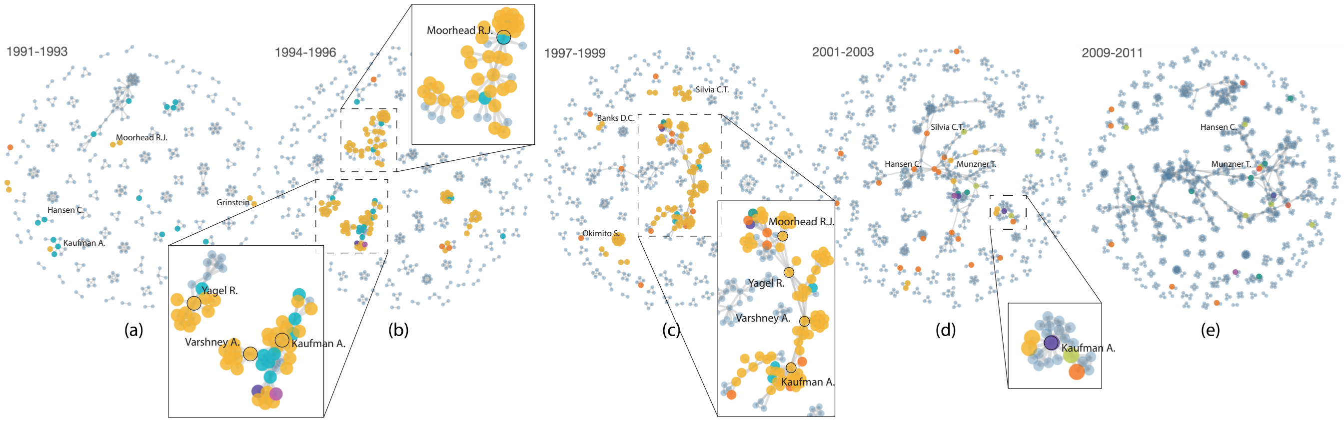


Figure 5: Small Multiple View, showing the evolutions of the raw data. We can highlight the selected nodes of one topic for different periods. Three groups formed a large group throughout different years. These dynamic scenarios can be explored from the specific topic.

from 1990 to 2015 [28]. The nodes in the graph represent authors. A link is added between two authors when they collaborated to publish a VIS paper. There are in total 4,813 nodes and 14,033 links. (Figure 1).

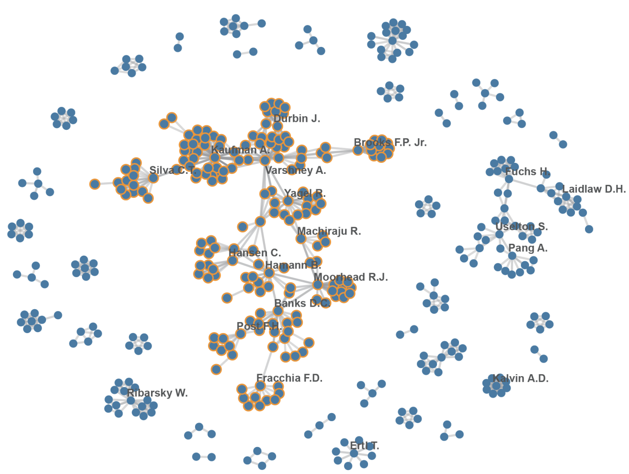


Figure 6: The structure of the dominating topic in 1995-1999 with active researchers and their collaborations highlighted. There are three connected sub graphs with different behaviors in different time periods. The detailed could be interactively explored in Figure 5

.

In the preprocessing step, the length of sliding windows $\omega$ is set to three years, and their overlap $\varepsilon$ is set to two years. Then we obtain 24 graphs. We set the number of topics $K$ to 10, and run the LDA algorithm with 2,000 iterations. The results are visualized in Figure 1. In the projection view, we could observe three clusters of projected points, which represent different stable periods in the growth the VIS conference. In the early times of the conference, the coauthor relations were stable, since the researchers are still in a small number. Then graph structures changed a lot in the middle years from 1996 to 2001, which is indicated by the path in the projection view. We speculate that it was caused by the rise of IEEE InfoVis, which contributed a lot to the diverse features in those years. There is also a visible change in the graphs structures around 2005 and 2006, which could be brought by the start of the VAST symposium. In recent years, the structures are gradually becoming stable, since more researchers are contributing to the conference consistently.

We further investigate the trend and its change in different periods in the Structure-Time View (Figure 1b). In the first 10 years, two main topics dominated. While in recent years, the dominating topics changed faster, which indicated that main structures of collaboration groups are changing faster than before. Further, we could explore the details of extracted topics. These topics successfully extracted the main structures of the collaborations of coauthor graphs in different years. We provide two topics as illustrations. First, we are interested the dominating topic in 2013-2015 and checked the details of the topic (Figure 1c). We find the active researchers, e.g. Hanspeter Pfister, Eduard Gröller, Huamin Qu, Thomas Ertl, etc, and their collaborations as the backbone of this topic. We select a subset of nodes (Figure 1c - larger nodes) and highlight them in the graphs of recent years using small multiples (Figure 1e). We can find that these authors were becoming more active and had more collaborations from 2009 to 2015.

Further, we can explore other topics and find evolutionary connection patterns. We investigate the topic that was dominating periods in 1995-1999. In the Topic View (Figure 6), the links with high probabilities connect nodes corresponding to authors including Arie Kaufman, Claudio Silvia, Hansen Charles, Amitabh Varshney, and Roni Yagel, etc. We select them and their collaborated people for further exploration. We could observe they are active in different periods of time (Figure 5). Before 1994, only a small portion of them published papers, including Arie, Hansen, etc. (Figure 5a). In 1994-1999, we could find the stronger collaborations among them. For example, Amitabh had more publications with Arie Caufman. We checked the fact and found that Amitabh joined Stony Brook to collaborate with Arie Kaufman in 1994. Later, the connections become wider, and more people involved in the connections (Figure 5b, c). Afterwards, these people moved to other collaboration structures in different colors, like Claudio Silva (Figure 5d). Finally in recent years fewer corresponding nodes were highlighted in the snapshot (Figure 5e).

In this case, we can observe the general evolutionary patterns based on the detected topics derived from our GraphLDA method, and similarities-based dimension reduction results. We can also generate hypotheses and explore them into details to find the structural evolution that leads to the change of the overall trends. Using our system, events, like emerging, dismissing, updating, etc., of specific groups can be observed at different time steps.

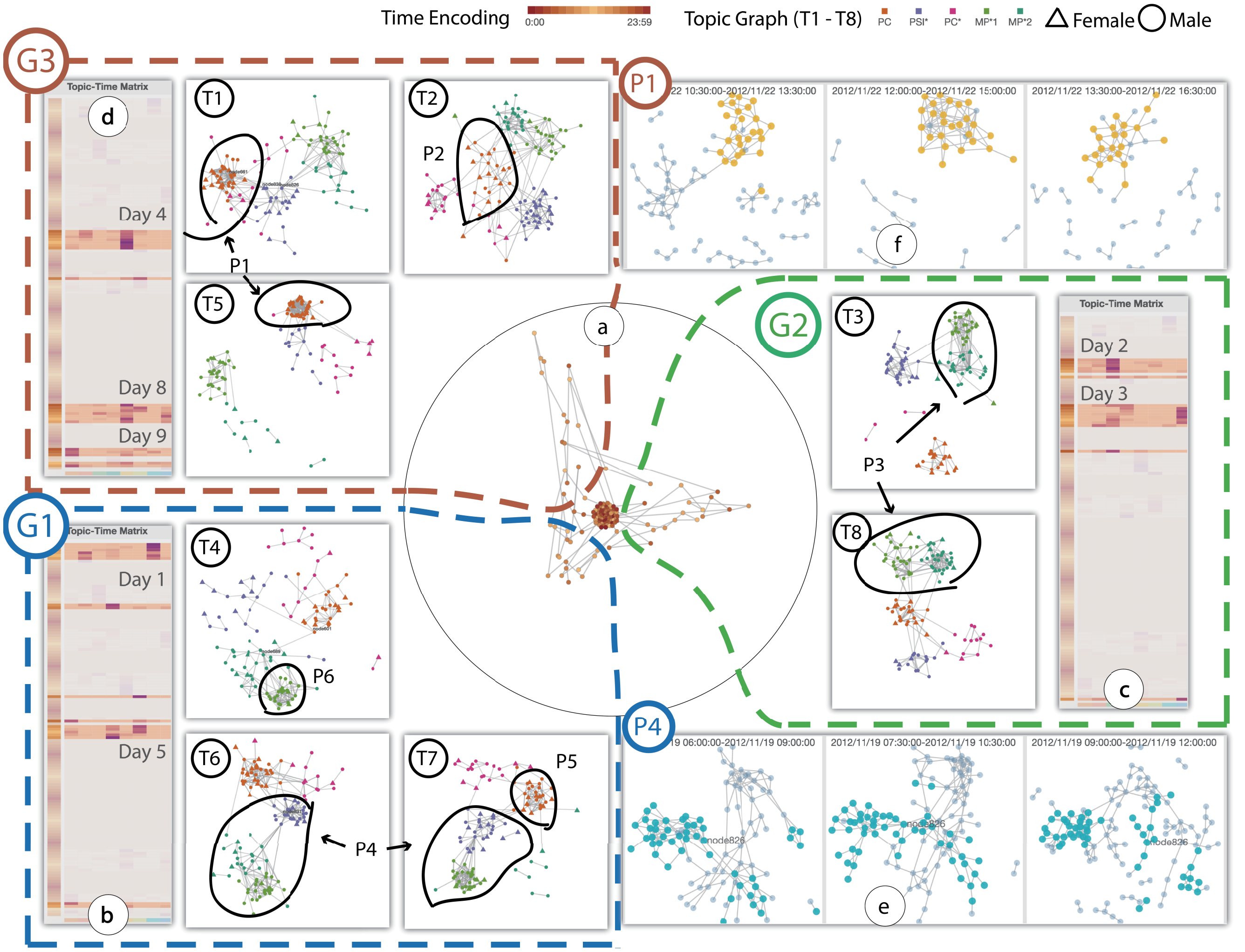


Figure 7: The visual exploration of evolutionary patterns in the social community network. From the projection view (a), we can see a hair-ball cluster and three groups distributed (G1, G2, and G3). We could examine the similarities and relationship among these days by selecting different dominating topics in different time periods (b, c, d). In each group, there are several dominating topics (T1-T8) that represent the main structures at that periods. We could further explore how these patterns distributed in the raw data (e, f). We also identify several subsets of nodes as interesting structures (P1-P6).

### 6.2 Case 2: Social Communication Network

The data used is a real-world face-to-face contacts records [1]. It contains 180 students in the school and the time range is from November 19 to November 27 in 2012. Specifically, nine days, including seven weekdays and two days in the weekends are contained. The students belong to 5 classes in the school. Each student is represented as a node in the graph. Each contact between two students is recorded by a wearable sensor and can be regarded as a link. There are 45,047 contacts, among which 10,104 are unique links. Our goal is to analyze the communication patterns and how these patterns changed along the time dimension.

In the preprocessing and LDA stage, the length of the sliding window $\omega$ is set to 180 minutes, and the overlap $\varepsilon$ is set to 90 minutes. After aggregation, 135 time steps are obtained. We test different settings of the number of topics $K$ from the range 1 - 20. To avoid losing expressiveness when $K$ is set too small, or being interfered with too many details when $K$is set too large, we use 8 as an experienced number. We then run the LDA algorithm with 2,000 iterations. The results are visualized in Figure 7. In the visualization settings, we use the triangle shape to indicate the females while the dot indicates males. We need to note that the color scheme in the Topic View (Figure 7 T1-T8) is used to represent classes instead of topics. Because people intended to communicate with students in the same classes, we regard it as important features to highlight.

In the Projection View (Figure 7a) and Time-Matrix View (Figure 7b, c, d), we use a specific color mapping in order to show the periodic behaviors. The white indicates the daytime and the dark red indicates the night. From the Projection View with a t-SNE projection (Figure 7a), we could see a hair-ball-like cluster in the center, which indicates graphs with very similar patterns. By drilling down to the time information, we find these graphs correspond to the weekday nights and the weekend.

With our approach, we could further tell the evolution of structures in the left seven weekdays. By brushing the similar regions around the center ball (Figure 7a), we could observe three groups of graphs (Figure 7-G1, G2, G3). We report the interactive exploration results here. For each group, we provide a Topic-Time Matrix View with corresponding dominating topics. Generally speaking, we could observe the sequential pattern changes - different topics dominated in different time periods, as well as periodical patterns. For example, in G3, the topic T5 structure has recurred. In day 1 and Day 5 (Figure 7b), the connection structures could be grouped as G1 (Figure 7-G1), topic T6 and T7 share structures that the purple and green class had strong inner connections (Figure 7-P4). Moreover, we found that an active person - id 826 was bridging the communications between these two classes. We could verify our findings in the detailed raw data shown in the selected small multiples (Figure 7e). Similarly, in Day 2 and Day 3 (Figure 7c), we find dominating structures of strong inter-connections of green and cyan classes in group G2 (Figure 7-G2). The nodes within the green and cyan classes are tightly connected (Figure 7-P3). By checking the ground truth, we confirm our findings that these two classes are in the same group, so that students in these two classes have more chances to communicate with each other than the left three classes. By examining the left days - Day 4, 8 and 9 (Figure 7d), we observe patterns forming the group as G3 (Figure 7-G3). We detect the recurring patterns in Day 4 and Day 8 with the same dominating topic T5. We find the two types of communication structures of the orange class. In topic T1 and T5, nodes are inner-connected tightly (Figure 7-P1). The second type is an interesting pattern that the people in the orange classes, acting as hubs of the other four classes (Figure 7-P2). In Day 4, the patterns happened together with other structures, which was not easy to decompose such structures from the original graphs from previous methods.

In this case, we have demonstrated our capability in detecting semantic meaningful structures. We are also able to illustrate the evolutionary patterns of dynamic graphs in the context of these main structures. In the other aspects, we can detect community behaviors, inter-community behaviors and connected backbone of the graphs, which can be applied to the general dynamic graph analysis.

## 7 Discussion

In this work, we present a novel LDA-based method for analyzing the evolution of dynamic graph and main structures simultaneously. We adapt the advanced techniques in topic modeling for visual analysis of dynamic graphs, with a proper design consideration. Different from the existing works that investigate global similarity and patterns in the dynamic graph, our methods can further allow users to explore the evolution of main structures in graphs. Moreover, the extracted topics represent certain semantic structures and temporal information in the dynamic graphs as shown in our case studies. We provide visual analytics tools to trace how such semantic structures evolve, such as the add or removal of nodes and links. The semantic structure is a more general concept compared with the concept of community, which is already widely studied. For example, in our first case study, we could discover how the InfoVis-related authors and collaborations emerged in the IEEE VIS conference. While the community detection methods could hardly identify the whole lifetime, especially the very beginning when InfoVis community is not well formed. As a whole, our LDA-based analysis method shows a great of potential in our case studies.

However, there are some limitations in our method. One issue is the parameter setting in LDA model. Currently, we allow users to manually set the number of topics for each dataset. We would like to further adapt automatic suggestion methods to set the number of topics. In the other aspect, LDA model is based on stochastic process. With the same input and parameter settings, the output would be slightly different. Currently, to support the stable analysis, we record the LDA results thus users can reproduce the previous findings in the same settings. Moreover, our model can leverage new advanced LDA techniques in the future.

It is a first trial to analyze the dynamic graph with the LDA method in visual analytics. By comparing the methods of classic clustering and community detection, we summarized the advantages for using LDA to analyze the dynamic graph. We have learned lessons in the design process. The advantages of LDA are that extracted topics could represent the structure information considering the temporal information and the probability description naturally fits the attributes dynamic graph. We have tested several candidates for the documents and words, and finally chosen the current approach. The key issue is to make the mapping from graph concepts to LDA entities be interpretable. Using graphs in each time step as documents and links as words is reasonable, because the extracted topics can represent structures in the dynamic graph. Other trials fail because they lost structural information or the extracted topics are not interpretable. Thus users can not understand what are the representative structures.

## 8 Conclusion and Future Work

In this work, we present a novel LDA-based visual exploration method for analyzing the dynamic graph. Taking considerations of the structural information and temporal information, we are able to extract the main structures and analyze the evolution of dynamic graphs, such as stable states, recurring states, and outlier. We provide a visual analytics system supporting the LDA-based visual exploration. The system supports users to explore the evolution of dynamic graph with the changes of the main structures. With the two case studies of real-world datasets, we have demonstrated the capability of our method.

Though powerful and inspiring, we can still improve our method from the following perspectives. First, we would like to update the mapping method when employing LDA model in dynamic graph analysis. On one hand, we would consider nodes' attributes when calculating the weight of links, which is the input of LDA model. On the other hand, to gain semantic structures, it is useful to consider more topological information in our method, such as betweenness, centrality and so on. We plan to evaluate our methods with such quantitative attributes in the future. On the one hand, users can analyze the evolution of the dynamic graph based on the probability distribution of topics in graphs. However, it is still a challenge to represent the topic probability distribution of links in the original graphs. Currently, we only assign node color based on the topic with highest probability value it had, which might lose information. We envision for improving that in the future.

## References

[1] Sociopatterns. website. http://www.sociopatterns.org/ datasets/, 2015. [online; accessed 2016-09-01].

[2] B. Bach, E. Pietriga, and J. D. Fekete. Graphdiaries: Animated transitions andtemporal navigation for dynamic networks. *IEEE Transactions on Visualization and Computer Graphics*, 20(5):740–754, May 2014.

[3] B. Bach, C. Shi, N. Heulot, T. Madhyastha, T. Grabowski, and P. Dragicevic. Time curves: Folding time to visualize patterns of temporal evolution in data. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):559–568, Jan 2016.

[4] F. Beck, M. Burch, S. Diehl, and D. Weiskopf. The state of the art in visualizing dynamic graphs. *EuroVis STAR*, 2014.

[5] F. Beck, M. Burch, C. Vehlow, S. Diehl, and D. Weiskopf. Rapid serial visual presentation in dynamic graph visualization. In *2012 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, pages 185–192. IEEE, 2012.

[6] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[7] M. Bostock, V. Ogievetsky, and J. Heer. $D^3$ data-driven documents. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2301–2309, 2011.

[8] M. Burch, M. Fritz, F. Beck, and S. Diehl. Timespidertrees: A novel visual metaphor for dynamic compound graphs. In *2010 IEEE Symposium on Visual Languages and Human-Centric Computing*, pages 168–175. IEEE, 2010.

[9] L. Che, J. Liang, X. Yuan, J. Shen, J. Xu, and Y. Li. Laplacian-based dynamic graph visualization. In *2015 IEEE Pacific Visualization Symposium (PacificVis)*, pages 69–73. IEEE, 2015.

[10] D. Chu, D. Sheets, Y. Zhao, Y. Wu, J. Yang, M. Zheng, and G. Chen. Visualizing hidden themes of taxi movement with semantic transformation. In *Proc. IEEE Pacific Visualization Symposium 2014*, pages 137–146. IEEE, 2014.

[11] W. Cui, S. Liu, L. Tan, C. Shi, Y. Song, Z. Gao, H. Qu, and X. Tong. Textflow: Towards better understanding of evolving topics in text. *IEEE Trans. Vis. Comput. Graph.*, 17(12):2412–2421, 2011.

[12] S. Diehl and C. Grg. Graphs, they are changing - dynamic graph drawing for a sequence of graphs. In *Proc. 10th Int. Symp. Graph Drawing (GD 2002), number 2528 in Lecture Notes in Computer Science, LNCS*, pages 23–31. Springer-Verlag, 2002.

[13] T. Dwyer and P. Eades. Visualising a fund manager flow graph with columns and worms. In *Information Visualisation, 2002. Proceedings. Sixth International Conference on*, pages 147–152. IEEE, 2002.

[14] P. Eades and M. L. Huang. Navigating clustered graphs using force-directed methods. *J. Graph Algorithms Appl.*, 4(3):157–181, 2000.

[15] C. Erten, S. G. Kobourov, V. Le, and A. Navabi. Simultaneous graph drawing: Layout algorithms and visualization schemes. In *International Symposium on Graph Drawing*, pages 437–449. Springer, 2003.

[16] T. Falkowski, J. Bartelheimer, and M. Spiliopoulou. Mining and visualizing the evolution of subgroups in social networks. In *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*, pages 52–58. IEEE Computer Society, 2006.

[17] D. Forrester, S. G. Kobourov, A. Navabi, K. Wampler, and G. V. Yee. Graphael: A system for generalized force-directed layouts. In *International Symposium on Graph Drawing*, pages 454–464. Springer, 2004.

[18] S. Fortunato and D. Hric. Community detection in networks: A user guide. *arXiv preprint arXiv:1608.00163*, 2016.

[19] C. Friedrich and P. Eades. The marey graph animation tool demo. In *International Symposium on Graph Drawing*, pages 396–406. Springer, 2000.

[20] C. Friedrich and P. Eades. Graph drawing in motion. *J. Graph Algorithms Appl.*, 6(3):353–370, 2002.

[21] T. E. Gorochowski, M. di Bernardo, and C. S. Grierson. Using aging to visually uncover evolutionary processes on networks. *IEEE transactions on visualization and computer graphics*, 18(8):1343–1352, 2012.

[22] M. Greilich, M. Burch, and S. Diehl. Visualizing the evolution of compound digraphs with timearctrees. In *Computer Graphics Forum*, volume 28, pages 975–982. Wiley Online Library, 2009.

[23] S. Hadlak, H. Schumann, C. H. Cap, and T. Wollenberg. Supporting the visual analysis of dynamic networks by clustering associated temporal attributes. *IEEE transactions on visualization and computer graphics*, 19(12):2267–2276, 2013.

[24] K. Henderson and T. Eliassi-Rad. Applying latent dirichlet allocation to group discovery in large graphs. In *Proceedings of the 2009 ACM symposium on Applied Computing*, pages 1456–1461. ACM, 2009.

[25] T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57, 1999.

[26] T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 42(1/2):177–196, 2001.

[27] F. Hong, C. Lai, H. Guo, E. Shen, X. Yuan, and S. Li. Flda: Latent dirichlet allocation based unsteady flow analysis. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2545–2554, Dec 2014.

[28] P. Isenberg, F. Heimerl, S. Koch, T. Isenberg, P. Xu, C. Stolper, M. Sedlmair, J. Chen, T. Möller, and J. Stasko. Visualization publication dataset. Dataset: http://vispubdata.org/ , 2015. Published Jun. 2015.

[29] T. K. Landauer, P. W. Foltz, and D. Laham. An introduction to latent semantic analysis. *Discourse Processes*, 25:259–284, 1998.

[30] H. Lee, J. Kihm, J. Choo, J. T. Stasko, and H. Park. ivisclustering: An interactive visual document clustering via topic modeling. *Comput. Graph. Forum*, 31(3):1155–1164, 2012.

[31] W. Medhat, A. Hassan, and H. Korashy. Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, 5(4):1093 – 1113, 2014.

[32] K. T. Miller and T. Eliassi-Rad. Continuous time group discovery in dynamic graphs. In *Notes of the 2009 NIPS Workshop on Analyzing Networks and Learning with Graphs, Whistler, BC, Canada*, 2009.

[33] X.-H. Phan and C.-T. Nguyen. Gibbslda++: A c/c++ implementation of latent dirichlet allocation (lda), 2007.

[34] S. Rufiange and M. J. McGuffin. Diffani: Visualizing dynamic graphs with a hybrid of difference maps and animation. *Visualization and Computer Graphics, IEEE Transactions on*, 19(12):2556–2565, 2013.

[35] L. Shi, C. Wang, and Z. Wen. Dynamic network visualization in 1.5d. In *Visualization Symposium (PacificVis), 2011 IEEE Pacific*, pages 179–186, March 2011.

[36] M. Steiger, J. Bernard, S. Mittelstädt, H. Lücke-Tieke, D. Keim, T. May, and J. Kohlhammer. Visual analysis of time-series similarities for anomaly detection in sensor networks. In *Computer Graphics Forum*, volume 33, pages 401–410. Wiley Online Library, 2014.

[37] S. van den Elzen, D. Holten, J. Blaas, and J. J. van Wijk. Dynamic network visualization withextended massive sequence views. *Visualization and Computer Graphics, IEEE Transactions on*, 20(8):1087–1099, 2014.

[38] S. van den Elzen, D. Holten, J. Blaas, and J. J. van Wijk. Reducing snapshots to points: A visual analytics approach to dynamic network exploration. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):1–10, Jan 2016.

[39] C. Vehlow, F. Beck, P. Auwärter, and D. Weiskopf. Visualizing the evolution of communities in dynamic graphs. In *Computer Graphics Forum*, volume 34, pages 277–288. Wiley Online Library, 2015.

[40] C. Vehlow, F. Beck, and D. Weiskopf. Visualizing dynamic hierarchies in graph sequences. *IEEE transactions on visualization and computer graphics*, 2015.

[41] H. Vogel. A better way to construct the sunflower head. *Mathematical biosciences*, 44(3-4):179–189, 1979.

[42] T. von Landesberger, S. Diel, S. Bremm, and D. W. Fellner. Visual analysis of contagion in networks. *Information Visualization*, 14(2):93–110, 2015.

[43] X. Wang, W. Dou, W. Ribarsky, D. Skau, and M. X. Zhou. Leadline: Interactive visual analysis of text data through event identification and exploration. In *Proceedings of the 2012 IEEE Conference on Visual Analytics Science and Technology (VAST)*, VAST '12, pages 93–102, Washington, DC, USA, 2012. IEEE Computer Society.

[44] X. Wang and E. Grimson. Spatial latent dirichlet allocation. In *Proceedings of Neural Information Processing Systems Conference (NIPS) 2007*, 2007.

[45] F. Wei, S. Liu, Y. Song, S. Pan, M. X. Zhou, W. Qian, L. Shi, L. Tan, and Q. Zhang. Tiara: a visual exploratory text analytic system. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 153–162, 2010.

[46] H. Zhang, B. Qiu, C. L. Giles, H. C. Foley, and J. Yen. An lda-based community structure discovery approach for large-scale social networks. *ISI*, 200, 2007.