Community Detection through Representation learning in Evolving Heterogenous Networks

A Master's Thesis proposal

EGOR DMITRIEV, Utrecht University, The Netherlands

Recent developments in big data and graph representation learning have allowed researchers to make breakthroughs in social network analysis and the identification of communities. While opening a lot of research opportunities, such approaches are highly limited to snapshots of rapidly evolving social networks. This, in fact, is a great simplification of the real-world situation which is always evolving and expanding by the user and/or machine interactions.

Relying on novel research of dynamic graph representation learning, the goal of my thesis project is to build a framework for community detection and representation in evolving heterogeneous networks. To verify the merit of the proposed framework, it will be evaluated against baselines on static heterogeneous graphs, and analyzed against gathered twitter dataset on covid measures.

Contents

Author's address: Egor Dmitriev, e.dmitriev@students.uu.nl, Utrecht University, The Netherlands.

1 INTRODUCTION AND BACKGROUND

Social Network Analysis (SNA) is a huge part of the Network Science field and is concerned with the process of investigating social structures that occur in the real-world using Network and Graph Theory. These social structures usually include social media networks, economic transaction networks, knowledge networks, and disease transmission networks.

One main issue to address while studying this type of real-world events lies in the identification of meaningful substructures hidden within the overall complex system. The SNA is therefore applied to extract patterns from the data usually in form of information flow, identification of high throughput nodes and paths, and discovery of communities and clusters. In this thesis, we are going to focus on the problem of community discovery.

This thesis proposal is structured as follows: in this section, we are going to introduce basic concepts and challenges of Dynamic Community Detection. In Section 2 a brief literature survey is conducted on identifying the current state of the art and approaches to Dynamic Community Detection. In Section 3 we will describe the problem we are trying to solve as well as formulate the research questions. In Section 4 we will elaborate on our proposed methodology for solving the posed problem and answering the research questions. Finally, in **??** the concrete planning for the research project is laid out.

1.1 Community Detection

The problem of partitioning a complex network into *communities* which represent groups of individuals with high interaction density, while individuals from different communities have comparatively low interaction density is known as Community Discovery (CD). CD is a task of fundamental importance within SNA as it discloses deeper properties of networks. It provides insight into networks' internal structure and its organizational principles.

Many useful applications of CD have been studied by researchers including identification of criminal groups [48], social bot detection [23], targeted marketing [34], and public health/disease control [47].

With the explosion of human- and machine-generated data, often collected by social platforms, more datasets are emerging having rich temporal information that can be studied. CD operates only on static networks. Meaning that their temporal dimension is often omitted, which often does not yield a good representation of the real world, where networks constantly evolve. Such networks are often referred to as dynamic networks as their components such as nodes and edges may appear and fade from existence. Accordingly community detection on such dynamic networks is called Dynamic Community Detection (DCD).

DCD algorithms incorporate additional temporal data are often able to both outperform their counterpart CD algorithms [10, 13, 30, 43], as well as provide additional information about communities for analysis [39]. This additional information comes in form of community events such as (birth, growth, split, merging, and death) or in form of the ability to track the membership of certain individuals over time.

1.2 Challenges in Community Detection

DCD is seen as the hardest problem within Social Network Analysis. The reason for this is mainly because DCD, unlike CD, also involves tracking the found communities over time. This tracking relies on the consistency of the detected communities, as usually slight changes to the network may

cause a different community membership assignment. Not properly accounting for this uncertainty may cause community and result drift [7].

Additionally, the increasing richness of the data is not only limited to temporal data. The real-world data often connects entities of different modalities. This multi-modality occurs through the fact that the entities and relations themselves may be of different types (meta topology-based features). For example users, topics, and documents in a social network (or vehicles and landmarks in a traffic network). More complex networks may include asymmetric relationships, and temporal networks may include appearing, disappearing, or streaming edges/nodes.

Another example of multi-modality in networks comes in form of node and relation features (content-based features). These features may come in form of structured (numerical, categorical, or vector data) or unstructured data such as images and text. It is of high importance to explore this multi-modal data as it may not always be possible to explain the formation of communities using network structural information alone.

Finally, a more systematic issue is that there is no common definition for a community structure. Within networks, it is usually described in terms of membership assignment, while in more content-based settings communities are described in terms of modeled topics (that usually represent interest areas) or distributions over latent similarity space. Both definitions have their shortcomings as they often fail to account for more complex community structures (such as overlapping and hierarchical communities) and non-linearity of structures often found in the real world.

Task of community detection is often compared to clustering and graph clustering, which not always may be a fair comparison as a main focus point in many CD algorithms is the fact that the amount of communities is unknown a priori. Communities are not are never planted in the real world and the algorithms should detect them in an unsupervised manner.

2 LITERATURE REVIEW

The problem of dynamic community detection was noticed quite early on within the SNA community and a considerable amount of research has been made in order to provide a comprehensive analysis of the network. While the said research was mostly focused on the discovery of communities using topologically-based features and node connectivity, the covered methods did research the limitations and challenges posed by a temporal context.

In recent years, significant developments have been made in the space of deep learning. Mainly in the development of new deep learning methods capable of learning graph-structured data [4, 18, 25] which is fundamental for SNA. Because of this, various problems within the field have been revisited, including community detection problems. The approaches have been expanded by incorporation of more complex features, solving the problems concerning multi-modality, and the introduction of unsupervised learning.

Despite this resurgence, the DCD problem has received little attention. Though a few efforts have been made to incorporate the deep learning methods by introducing content-based similarity dynamic, the definition of unified constraints for end-to-end learning, and usage of graph representation-based CD algorithms within a temporal context, the current state of the art leaves a lot to be desired.

We structure the literature as follows: first, we describe the various interpretations of the Community Structure in Section 2.1. Next, we explore various approaches and techniques related to Graph Representation Learning in Section 2.2. Then, we provide an overview of the current state-of-the-art

approaches for Community Detection and Dynamic Community Detection tasks in Section 2.3 and Section 2.4. Finally, we discuss the ways to evaluate the said algorithms in Section 2.5 and the datasets available in Section 2.6.

2.1 Community Structures

The goal of this section is to introduce fundamental structures for the Dynamic Community Detection task. We do this by combining various definitions used in the relevant literature as well as establishing the purpose for these structures, before proceeding into approaches for detecting communities in the following sections.

2.1.1 Communities

Communities in real-world networks can be of different kinds: disjoint (students belonging to different educational institutions), overlapping (person having membership in different social groups) and hierarchical (components of a car). One of the main reasons behind the complexity of CD is that there is not one unique definition what a community actually is.

The *link-based* (also referred to as classic) community detection methods intuitively describe communities as groups of nodes within a graph, such that the intra-group connections are denser than the inter-group ones. This definition is primarily based on the *homophily* principle, which refers to the assumption that similar individuals are those that are densely connected together. Therefore, these kind of methods look for sub-graph structures such as cliques and components that identify connectedness within the graph structure to represent the communities.

Unfortunately, in most cases link-based methods fall short to identify communities of similar individuals. This is mainly due to two facts: (i) many similar individuals in a social network are not explicitly connected together, (ii) an explicit connection does not necessarily indicate similarity, but may explained by sociological processes such as conformity, friendship or kinship [8, 10].

A more general definition is introduced in [6] to create an underlying concept generalizing all variants found in the literature (+**?**]). In link-based methods, a direct connection is considered as a particular and very important kind of action, while newer methods also consider content or interest overlap.

Community A community in a complex network is a set of entities that share some closely correlated sets of actions with the other entities of the community.

2.1.2 Dynamic Communities

Similar to how communities can be found in static networks, dynamic communities extend this definition by utilizing the temporal dimension to define their life cycle/evolution over a dynamic network. A dynamic community is characterized by a collection of communities and a set of transformations on these communities over time.

This persistence of communities across time subjected to progressive changes is an important problem to tackle. Though, as noted by [44] the problem can be compared to the famous "the ship of Theseus" paradox. Because (verbatim), *deciding if an element composed of several entities at a given instant is the same or not as another one composed of some—or even none—of such entities at a later point in time is necessarily arbitrary and cannot be answered unambiguously.*

Most of the works agree on two atomic transformations on the communities, including node/edge appearance and vanishing. While some such as [1, 5, 39] define a more extensive set of transformations (also referred to as events) which may be more interesting for analytical purposes:

- Birth, when a new community emerges at a given time.
- Death, when a community disappears. All nodes belonging to this community lose their membership.
- Growth, when a community acquires some new members (nodes).
- Contraction, when a community loses some of its members.
- Merging, when several communities merge to form a new community.
- Splitting, when a community is divided into several new ones.
- Resurgence, when a community disappears for a period and reappears.

These events/transformations are often not explicitly used during the definition and/or representation of dynamic communities. Nevertheless, most of the methods covered in the following sections do define a way in their algorithm to extract such events from the resulting data.

Finally, it is important to note that dynamic networks can differ in representation. They can be represented as either a time series of static networks (also referred to as snapshots) or as a real-time stream of edges (referred to as temporal networks). Within the global context of dynamic community detection, they can be seen as equivalent as the conversion between the two representations can be done in a lossless way. The latter, temporal networks are often used to handle incremental changes to the graph and are most commonly applied within real-time community detection settings.

2.2 Graph Representation Learning

The representation-based approaches stem from the field of computational linguistics which relies heavily on the notion of *distributional semantics* stating that words occurring in similar contexts are semantically similar. Therefore the word representations are learned as dense low-dimensional representation vectors (embeddings) of a word in a latent similarity space by predicting words based on their context or vice versa [33, 41]. Using the learned representations similarity, clustering and other analytical metrics can be computed.

The success of these representation learning approaches has spread much farther than just linguistics as similar ideas are also applied to other fields including graph representation learning. Methods such as deepwalk [42], LINE [50], and node2vec [15] use random walks to sample the neighborhood/context in a graph (analogous to sentences in linguistic methods) and output vector representations (embeddings) that maximize the likelihood of preserving the topological structure of the nodes within the graph.

Whereas previously the structural information features of graph entities had to be hand-engineered, these new approaches are data-driven, save a lot of time labeling the data, and yield superior feature/representation vectors. The methods can be trained to optimize for *homophily* on label prediction or in an unsupervised manner on link prediction tasks.

Newer approaches introduce the possibility for the fusion of different data types. GraphSAGE [17] and Author2Vec [54] introduce a methodology to use node and edge features during the representation learning process. Other approaches explore ways to leverage heterogeneous information present within the network by using *metapath* based random walks (path defined by a series of node/link types) [9] or by representing and learning relations as translations within the embedding space [3]. In Nguyen et al. [38] the authors introduce a way to encode temporal information by adding chronological order constraints to various random walk algorithms. Other relevant advancements within the field include Graph Convolutional Networks (GCN) [26] and (Variational) Graph Auto-Encoders (GAE) [24] which present more effective ways to summarize and represent larger topological neighborhoods or whole networks.

2.3 Link-based Approaches

Link-based approaches to (Dynamic) Community Detection rely on connection strength to find communities within the network. The main criteria for communities are the assumed property that intra-group connections are denser than inter-group ones. The networks are partitioned in such a way, that optimizes for a defined measure characterizing this property.

We start this section by covering the fundamentals of link-based community detection by introducing commonly used community quality measures and algorithms for optimizing them. Next, we introduce the link-based DCD problem and the unique challenges that arise as opposed to CD. Then we proceed to cover the current state of the art by describing the related works, their solutions to the said challenges, and possible extensions to the problem.

2.3.1 Community Detection

Different metrics exist quantifying the characteristic of *homophily* over edge strength. The most common metric is Modularity which measures the strength of the division of a network into modules (communities). Its popularity stems from the fact that it is bounded and cheap to compute, though it has other problems such as resolution limit (making detecting smaller communities difficult). Other metrics that can be found in the literature include but are not limited to:

- Conductance: the percentage of edges that cross the cluster border
- Expansion: the number of edges that cross the community border
- Internal Density: the ratio of edges within the cluster with respect to all possible edges
- Cut Ratio and Normalized Cut: the fraction of all possible edges leaving the cluster
- Maximum/Average ODF: the maximum/average fraction of nodes' edges crossing the cluster border

Modularity. Modularity directly measures the density of links inside a graph and is therefore computed on communities (sets of nodes) individually by weighing edges based on community similarity (or exact matching). Calculation of modularity is done by aggregating for each pair of nodes the difference between the expected connectivity (amount of edges between the nodes) and the actual connectivity (existence of an edge) given their degrees (Eq. 1). The final result represents the delta difference by how much the given graph exceeds a random graph as expected connectivity is determined by a random rewiring graph. Because intra-community pairs are weighted lower than inter-community pairs the score can vary.

$$Q = \frac{1}{2m} \sum_{vw} \sum_{r} \left[\overbrace{A_{vw}}^{\text{Connectivity}} - \underbrace{\frac{k_v k_w}{2m}}_{\text{Expected Connectivity}} \right] \overbrace{S_{vr} S_{wr}}^{\text{Community Similarity}} \tag{1}$$

Louvain Method. Finding an optimal partition of a graph into communities is an NP-hard problem. This is because, while calculating the modularity score can be done in a timely manner, still all possible node to community assignments have to be considered. Therefore heuristic-based methods such as the Louvain method are usually used.

Louvain method [2] is a heuristic-based hierarchical clustering algorithm. It starts by assigning each node in the graph to its own community. Then it merges these communities by checking for each node the change in modularity score produced by assigning it to a neighbor community

(based on the existence of a connection). Once the optimal merges are performed, the resulting communities are grouped into single nodes and the process is repeated.

Since modularity changes can be computed incrementally, the complexity of this method is $O(n \log n)$. Additionally, due to the flexibility of the modularity measure, it allows detecting communities in graphs with weighted edges.

Label Propagation algorithm. Another way to sidestep the complexity issue is using the Label Propagation algorithm as it uses the network structure directly to define partitions and doesn't require any priors (quality metrics). The intuition for the Label Propagation algorithm is as follows: When propagating a label through connections in the network, a single label quickly becomes dominant within a group of densely connected nodes, but these labels usually have trouble crossing sparsely connected regions.

The algorithm starts by assigning each node their own label. After that, for each iteration, each node updates its label to the majority label among its neighbors where ties are broken deterministically. The algorithm stops after a fixed amount of iterations or once it has converged. An important feature of this algorithm is that a preliminary solution can be assigned before each run, therefore only updating existing membership assignments.

2.3.2 Dynamic Community Detection

Dynamic Community Detection can be seen as an extension to community detection by the addition of the Community Tracking task. Tracking relies on the coherency and stability of found communities to define their evolution through time. The said properties are not be taken for granted and introduce new challenges when designing DCD methods. The main issue is in fact that they are competitive with each other causing a trade-off between community coherency/quality and community temporal stability.

Various strategies dealing with this trade-off are categorized by Rossetti and Cazabet [44] and Dakiche et al. [7] where authors reach a consensus over three main groups. In the following sections, we briefly introduce these strategies and describe the current state of the art in similar order.

Independent Community Detection and Matching. Also referred to as the two-stage approach. Works by splitting the DCD task into two stages. The first stage applies CD directly to every snapshot of the network. Followed by the second stage matching the detected communities between the subsequent snapshots.

The advantages of this approach include the fact that it allows for use of mostly unmodified CD algorithms for the first step and that it is highly parallelizable as both detection and matching steps can be applied to each snapshot independently. The main disadvantage is the instability of underlying CD algorithms which may disrupt the community matching process. Many CD methods may give drastically different results in response to slight changes in the network. During the matching, it becomes difficult to distinguish between this algorithm instability and the evolution of the network.

In Wang et al. [52] the authors circumvent this instability issue by looking at the most stable part of the communities, namely core/leader nodes. In their research, they observe that in various datasets most of the nodes change dramatically while only a small portion of the network persists stably. To exploit this feature, the algorithm CommTracker is introduced which first detects the said core nodes, and then defines rules to both extract communities as well as their evolutional events. The community members are assigned based on their connectivity relative to core nodes.

Rossetti [43] proposes a way to detect overlapping communities in dynamic networks. A more robust two-phase community detection method (ANGEL) is proposed to ensure the stability of the found communities. The first phase extracts and aggregates local communities for each node by applying Label Propagation on their Ego-Graph (graph with the said node removed). Because the found communities are biased due to a partial view of the network, they are merged in the second step based on their overlap yielding more stable communities with the possibility of overlap. During the matching for each snapshot, a step forward and backward in time is considered where community splits and merges are detected by reusing the matching criteria of the second phase of the CD step.

Dependent Community Detection. Dependent Community Detection strategy works detecting communities in each snapshot based on the communities found in the previous snapshots. This approach introduces temporal smoothness because the previous solution is reused making the matching process obsolete. Though as part of the described trade-off it can impact the long-term coherence of the dynamic communities. Mainly, because each step introduces a certain amount of error into the results (community drift) which may get amplified within further iterations (error accumulation). Another disadvantage is the fact that the strategy has limited possibilities of parallelization due to its dependent nature.

To lessen the complexity of continuous re-detection of the communities some algorithms process the changes incrementally by limiting the change to a local neighborhood. While this approach has many benefits, it is important to note that these algorithms face a problem where only applying local changes can cause communities to drift toward invalid ones in a global context.

He and Chen [19] introduces an efficient algorithm by modifying the Louvain method algorithm. Based on the observation that between consecutive timesteps only a fraction of connections changes and do not affect communities dramatically, they argue that if all community nodes remain unchanged, the community also remains unchanged. With this in mind, they make a distinction between two types of nodes, ones that change the connection in a snapshot transition and ones that do not. The former have to be recomputed, while the latter maintain their community label. The nodes that maintain their community are merged into community nodes with edges to other community nodes and changed nodes weighted proportionally to their real connectivity (amount of edges when ungrouped). This simplified graph is passed to the Louvain method algorithm for community detection. By reusing the community assignments temporal smoothness is maintained and due to the incremental nature of this algorithm, the overall complexity remains low.

Guo et al. [16] envision the target network as an adaptive dynamical system, where each node interacts with its neighbors. The interaction will change the distances among nodes, while the distances will affect the interactions. The intuition is that nodes sharing the same community move together, and the nodes in different communities keep far away from each other. This is modeled by defining the so-called Attractor algorithm, which consists of three interaction patterns that describe how node connection strength is influenced by neighboring nodes. The edge weights are initialized using Jaccard distance and the propagation is run until convergence. The communities can be extracted by thresholding on edge weight/distance. Thereafter, all changes are treated as network disturbances. The disturbance can be limited to a certain area using a disturbance factor which defines a bound on the possible propagation.

More recently the Yin et al. [58] has proposed an evolutionary algorithm by looking at the DCD from an Evolutionary Clustering Perspective. They detect community structure at the current time under the guidance of one obtained immediately in the past by simultaneously optimizing

for community quality score (modularity) and community similarity between subsequent time steps (NMI). In the methodology, a way is proposed to encode a graph efficiently into a genetic sequence. Additionally, new mutation and crossover operators are proposed which maximize either of the two objectives. By using a local search algorithm, building a diverse initial population, and selecting for dominant candidates the communities maximizing both objectives are obtained.

Simultaneous community detection. The final strategy we consider sidesteps the matching issue by considering all snapshots of the dynamic network at once. This is done by flattening the network in the temporal dimension and coupling edges between the same nodes at different timesteps. These approaches usually don't suffer from instability or community drift. The disadvantages include that the standard principle of a unique partition for each time step can not be applied, limiting the number of possible algorithms. Handling real-time changes to the graph are also usually not considered.

Mucha et al. [36] adopt a simple yet powerful solution to this problem by connecting identical nodes between different time steps within the unified network. On this network, they apply a modified Louvain method algorithm to extract the communities whose members can be split over different timesteps.

Ghasemian et al. [12] apply stochastic block model-based approach. They make a distinction between two edge types: (i) spatial edges (edges between neighbors) and (ii) temporal edges (edges connection nodes in different timesteps). Using this distinction, they define a Belief Propagation equations to learn marginal probabilities of node labels ($\mu_s^i(t)$) over time. Additionally in their research, they introduce a way to derive a limit to the detectability of communities. This is, because some communities may not be detectable as their probability nears that of chance.

2.4 Representation-based Approaches

The main difference between Representation-based approaches and link-based approaches is the fact that they usually don't directly model the network based on connections. Instead, they learn an intermediate representation of the graph or its components to which CD detection can be applied to. While also relying on the idea of *homophily*, most methods define additional objectives to improve community quality.

The main reasoning for this is the fact that real-world networks are non-linear, meaning that there may be no connections when they make sense and vice versa [53]. By using deep neural networks to learn these embeddings one can address such non-linearity as they are in general very robust against noise. Other notable benefits to using representation-based approaches include the fact that they compress the data efficiently as real-world networks are very sparse. They can also represent multi-modal features, network (meta) structure, and temporal dimension by defining them all in a compatible similarity space or learning mappings to this space. Finally, representations are naturally easy to compute similarity on.

In this section, we describe representation-based approaches by covering both CD and DCD approaches. To provide a more cohesive overview of the methods, we group them by their innovations instead.

Affiliation Graph Models. While arguably not being representational by itself, Affiliation Graph Network (AGM) model introduced in Yang and Leskovec [56] is very influential within the deep learning/representation learning CD field.

The AGM models a network as a bipartite graph with communities as first-class citizens and is represented by the following equation $B(V, C, M, \{p_c\})$, where V represents nodes, C set of communities, M node-community memberships and $\{p_c\}$ model parameters (a single probability p_c per community). It can model non-overlapping, overlapping, and hierarchical communities by defining rules on membership sets in M. AGM can be used in both generative and discriminative settings.

The generative scenario goes as follows: Given an AGM F generates links between each pair of nodes exceeding a baseline probability p. The can be done by considering that according to AGM, each pair of nodes in community A is connected with a probability p_A. Therefore, the probability of two nodes having a connection is proportional to the number of communities they share (which is defined in the model).

The discriminative scenario is defined as: Given a graph G and, find a model F that may have generated it. By assuming that the graph was generated using an AGM, the parameters M, number of communities $|C|$ and $\{p_c\}$ have to be found. Process for finding such a model to the graph usually max likelihood fitting. AGM is relaxed to have membership strengths F_{uC}, which helps to define the probability of nodes u and v connecting through community C ($P_C(u, v)$), and in terms of that by themselves $P(u, v)$. Using this a probability $P(G|F)$ can be constructed quantifying how well the model fits the data. Finally, gradient ascent can be applied to optimize the model parameters.

2.4.1 Graph Reinforcement

The first class of methods we consider are Graph Reinforcement methods. These methods use representation-based learning techniques to enhance the graph by adding valuable edges or reducing the noise by removing noisy connections. This is usually done by training a model on a link-prediction task. A notable benefit of this approach is that other well-known CD methods can be used on the enhanced graph afterward.

Kang et al. [22] present a CD algorithm agnostic pre-processing method for strengthening the community structure of a graph by adding non-existing predicted intra-community edges and deleting existing predicted inter-community edges. Their strategy is to learn topological embedding using a graph representation learning algorithm (node2vec) based on the existing link prediction task. The similarity is computed between different node pairs and put into buckets. Then with the assumption of *homophily* the buckets with a higher value can be considered holding intra-community while buckets with lower inter-community connections. Right buckets are picked from both extremes to create or delete edges. The preemptive CD is done to greedily guide pair-wise similarity computation and avoid a high complexity.

Jia et al. [21] solves the issue of detecting overlapping communities by proposing the CommunityGAN algorithm which jointly optimizes for node and community representations. First, they define a method for efficient motif (in their case clique) sampling from the graph (true/clique, and false/vertex subset). Then, they define a GAN based structure for learning representational vectors where the generator G tries to learn $p_{true}(m|\mathbf{v}_c)$ as preference distribution of motifs to generate most likely vertex subsets most likely to be real motifs. Discriminator D tries to learn the probability of a vertex subset being a real motif, therefore creating a minimax game of progressively optimizing embeddings to be able to encode rich information about network topology.

Both components (G and D) are implemented as a modified relaxed AGM model with a more general definition to be able to handle the motif generation (rather than edge generation). The probability of a set of vertices being a motif is defined in terms of their probability being a motif through a

community, therefore making them community-aware as they now represent the affiliation weight between a vertex and a community. The number of communities is chosen by training and testing part of data on link prediction task.

2.4.2 *Multi-objective optimization*

Another subject where representation-based approaches excel is multi-objective optimization. Usually, a combined objective is defined in terms of a community quality, temporal consistency, or homophily measure. These measures in turn use the proximity between the representation to be able to back-propagate the combined error and optimize the representation(function) directly.

In Rozemberczki et al. [46] authors propose a method that learns cluster centers along with node embeddings. They define objective function as a combination of three terms: normalization term (ensures embeddings are centered at the origin), proximity term (forces nodes with similar neighborhoods to be embedded close), cluster quality term (forces nodes to be close to their nearest cluster). Additionally, a "social network cost" or *homophily* term is added as a regularizer to optimize for proximity between nodes within the same cluster. During training, the clustering coefficient is annealed to ensure convergence and negative sampling is employed to avoid large softmax costs.

Yang et al. [55] propose a similar idea of combining embedding and clustering tasks and solving them in an end-to-end manner. In their work, they employ a Deep Denoise Autoencoder (DAE) to learn topological information of the network by optimizing for reconstruction loss. To learn cluster/community centers they define GRACE cluster module which first computes soft cluster assignment matrix Q by utilizing the embeddings and cluster centers which contains probabilities q_{ik} of node i belonging to cluster k. The clustering loss is defined as KL-divergence between the soft clustering Q and auxiliary target distribution P which is defined by squaring and normalizing the soft assignments to reinforce more confident clustering results while preventing the formation of excessively large clusters. Both embeddings and clustering are optimized alternatively until convergence.

Ma et al. [32] proposed a novel approach to constructing community-aware dynamic network embeddings by leveraging multi-objective optimization and extending it into a temporal dimension. They adopt a Graph Autoencoder structure which works by encoding the full graph into a lower-dimensional structure and decoding it again into a graph. Assuming a well-tuned network, this allows authors to encode the network (and its nodes) into more efficient representation vectors which characterize the network well

The objective function they use is defined by three terms: the reconstruction error term minimizing the distance between the ground-truth and the autoencoder output, local structure/homophily preservation term minimizing first- and second-order proximity between connected nodes, and the community evolution preservation term maximizing temporal smoothness of communities at different granularity levels given their representation as an aggregation of their members.

The initial community assignment is generated using the Louvain method for high-level communities and using k-means for fine-grained communities given a max community size parameter w. After that, embeddings at each snapshot are optimized by employing a dependent community detection-like strategy.

Wang et al. [53] employs a similar to DCD detection by utilizing the Graph Autoencoder architecture. As an addition authors add an additional community score term to the objective function also minimizing the distance between nodes in the same community. At last, K-means is run on the

representational vectors to detect communities at different timesteps while reusing the outputs from the previous step.

2.4.3 Multi-modal community detection

Another way to improve community quality is by incorporating multi-modal features. These can come in form of node attributes, content-based or meta-topological data. These representations are incorporated into learned embedding vectors either by direct learning, incorporating them into the objective function, or use of pre-trained models.

In Fani et al. [10] the authors describe their method for identifying user communities through multi-modal feature learning. First user embeddings are learned based on their temporal content similarity by looking at topics of interest. Per-user, a heat map is constructed measuring the user's interest over time and topic axes. By considering users like-minded if their heat maps overlap enough they train low-dimensional content embeddings spanning this user similarity space. Next, they use random walk-based GNN methods to learn topological similarity embeddings for network nodes. Finally, they modify the graph by setting edge weights proportionally to node proximity in this combined embeddings space. After that, the Louvain method is applied to extract these time and content-aware communities.

2.5 Evaluation

In the previous section we have described the different variations of community structure defintions as well as the approaches used for detecting these communities. In this sections we will cover how the found dynamic community structures can be evaluated in a more general setting to allow comparison of different approaches. While dynamic community detection problem can be seen as two tasks (resemblance/detection and matching/tracking) these two should separate evaluation tasks.

In the following sections we cover both of the evaluation tasks by classifying approaches used in the literature in three classes. Namely, annotated, metric-based and task specific.

2.5.1 Annotated

Evaluation of detected (dynamic) communities becomes much easier when the *ground truth communities* are provided. The evaluation is then done by comparing the difference between the produced communities and the effective ones. To perform this comparison, information theory based metric Normalized Mutual Information (NMI) is used which converts community sets to bit-strings and quantifies the "amount of information" can be obtained about one community by observing the other [29].

A possible drawback of this measure is that its complexity is quadratic in terms of identified communities. In [45] alternative measure (NF1) with linear complexity is introduced which similarly to F1 score uses the trade-off between precision and recall (of the average of harmonic means) of the matched communities. In the follow-up work [43] the authors describe a way to apply this measure within the context of DCD by calculating this score for all the snapshots and aggregating the results into one single measure.

Aside from NMI other measures are employed such as Jaccard Coeficient, Accuracy and Rand-Index measuring community overlap [31, 35, 55], Overlapping-NMI [57] and Omega-Index measuring is the accuracy on estimating the number of communities that each pair of nodes shares [56],

In real-world there are usually no ground truth communities. Therefore this approach is usually applied on synthetic datasets where the communities and their dynamicity is sampled from a distribution. Alternative approach some papers take is by defining ground truth communities using the metadata and node attributes present within the datasets. Some datasets may include annotated communities, but this is not common within DCD datasets.

2.5.2 Metric based

Another way to evaluate and compare different CD algorithms without knowing ground truth communities is using a quality function.

Network-based metrics. The first first group of measures we consider operate directly on the network structure. They are most commonly used to evaluate link-based methods as their results are network partitioning sets. Modularity is the most widely used measure [37, 49], since it measures the strength of division of a network into modules. Networks with high modularity have dense connections between the nodes within the modules, and sparse connections between nodes in different modules. Other measures are used as well including:

- **Conductance**: the percentage of edges that cross the cluster border
- **Expansion**: the number of edges that cross the community border
- **Internal Density**: the ratio of edges within the cluster with respect to all possible edges
- **Cut Ratio and Normalized Cut**: the fraction of all possible edges leaving the cluster
- **Maximum/Average ODF**: the maximum/average fraction of nodes' edges crossing the cluster border
- **Triangle Participation Ratio TPR**: measures fraction of triads within the community. A higher TPR indicates a denser community structure

Proximity-based measures. Proximity-based measures are often used to evaluate clustering tasks, but are also often used to evaluate representation-based CD methods since there is a large overlap in their methodology. Additionally, representation based approaches have a benefit of being able to quantify both entities and communities as a d-dimensional vector enabling a more direct comparison of the two [51]. The most common measures include:

- **Silhouette Coefficient**: Is defined as $S(i) = \frac{b(i)-a(i)}{\max\{a(i),b(i)\}}$ where $a(i)$ defines the mean distance from node i to other nodes in the same cluster while $b(i)$ is mean distance to any node not in the same cluster. It measures cohesion of a cluster/community and indicates how well a node is matched to its own cluster.
- **Davies-Bouldin Index**: Is the ratio of the sum of the average distance to the distance between the centers of mass of the two clusters. In other words, it is defined as a ratio of within cluster, to the between cluster separation. The index is defined as an average over all the found clusters, and is therefore also a good measure to deciding how many clusters should be used.
- **Calinski-Harabasz Index**: Is similarly the ratio of the between-cluster to the within-cluster variance. Therefore it measures both cohesion (how well its members fit the cluster) as well as compares it to other clusters (separation).

Stability-based measures. To evaluate temporal stability and consistency of the node and community structures, measures based on temporal smoothness are proposed in Ma et al. [32]. These measures compare the evolution rate of network structures against the evolution rate of the whole network given node embeddings between two consecutive snapshots. The intuition is that rapidly evolving

structures relative to global evolution rate are temporally unstable and thus of low quality. Metrics proposed include:

- **Network Stability**: Is defined as Eq. 2 and evaluates the evolution ratio of the low-dimensional node representations to the network representations between snapshots at a-th time stamp.
- **Community Stability**: Is defined as Eq. 3 and computes stability of communities in dynamic networks on the embedded low-dimensional representations. It is represented as evolution ratio of the low-dimensional community representations to the network representations between snapshots at a-th time stamp.

$$p_s^a = \frac{\left(\left\|\mathrm{H}^{a+1} - \mathrm{H}^a\right\|_2^2\right) / \|\mathrm{H}^a\|_2^2}{\left(\left\|\ \mathrm{A}^{a+1} - \mathrm{A}^a\right\|_2^2\right) / \|\mathrm{A}^a\|_2^2} \tag{2}$$

$$p_c^a = \sum_{k=1}^{q} \left(\frac{\left(\left\|\mathrm{H}_{c_k}^{a+1} - \mathrm{H}_{c_k}^a\right\|_2^2\right) / \left\|\mathrm{H}_{c_k}^a\right\|_2^2}{\left(\left\|\ \mathrm{A}_{c_k}^{a+1} - \mathrm{A}_{c_k}^a\right\|_2^2\right) / \left\|\mathrm{A}_{c_k}^a\right\|_2^2} \right) / q \tag{3}$$

In this case $\left\|\mathrm{H}_{c_k}^{a+1} - \mathrm{H}_{c_k}^a\right\|_2^2$ represents the normalized euclidean distance between the two clusters while $\left\|\ \mathrm{A}_{c_k}^{a+1} - \mathrm{A}_{c_k}^a\right\|_2^2$ represents the normalized euclidean distance between the adjacency matrices of the two clusters. For network stability no such grouping is done and evaluation is performed on global representation and adjacency matrices.

2.5.3 Task specific

In [40] the authors criticize annotation and metric based CD evaluation approaches by proving that they introduce severe theoretical and practical problems. For one, they prove the no free lunch theorem for CD, ie. they prove that algorithmic biases that improve performance on one class of networks must reduce performance on others. Therefore, there can be no algorithm that is optimal for all possible community detection tasks, as quality of communities may differ by the optimized metrics. Additionally, they demonstrate that when a CD algorithm fails, the poor performance is indistinguishable from any of the three alternative possibilities: (i) the metadata is irrelevant to the network structure, (ii) the metadata and communities capture different aspects of network structure, (iii) the network itself lacks structure. Therefore, which community is optimal should depend on it's subsequent use cases and not a single measure.

To address this issue, is common for to evaluate algorithm on both earlier described approaches as auxiliary tasks. The general sentiment behind it is, that communities have better quality if they improve an underlying application. In the following sections we describe a few commonly used auxiliary tasks in literature.

Link-prediction. A common way to evaluate quality of extracted node embeddings within Graph Representation learning is using the link-prediction task. As link-prediction can be done in unsupervised manner, it does not require additional labeling for evaluation. The edge set of the input network is split into a test set on which the model is trained, and a test set on which is used to compare the predicted links against. For node representation learning the predictions are defined by proximity between the nodes and a threshold. To quantify the quality of the results classification metrics are usually employed such as accuracy, precision, recall and f-score.

In context of CD and DCD the link-prediction is usually modified to measure the predicting capability of the community embeddings. Fani et al. [10] define a user prediction task to predict which users posted a certain news article at a certain time. Their methodology is, given a news at time t, find the closest community to the article in representational similarity space. All members of the given community are seen as predicted users over which the classification metrics are calculated. Similarly Ma et al. [32] modify the task by predicting whether the edges will still exist within the next time stamp to also quantify temporal prediction capability of the trained embeddings.

Recommendation Tasks. Another way quality of node representations can be evaluated, is using recommendation tasks. Here the idea is, instead of predicting a single item like in link-prediction, to rank the items based on their recommendation confidence. Using the ranked list, standard information retrieval metrics such as precision at rank, mean reciprocal rank and success at rank can be computed. This approach is applied to CD [10, 20, 46] by ranking recommendations for per community instead of on individual basis. Communities with higher scores therefore would indicate a high similarity between their members.

2.6 Datasets

2.6.1 Synthetic Datasets

Paper	Description
Lancichinetti et al. [28]	Static networks (widely used)
Greene et al. [14]	Generate Graphs based on Modularity measure
Granell et al. [13]	
Hamilton et al. [18]	Generate Time dependent Heterogeneous graphs using modularity optimization and multi-dependency sampling
SYN - Ghalebi et al. [11]	
SBM - Lancichinetti and Fortunato [27]	extracted from the dynamic Stochastic Block Model

2.6.2 Real World Datasets

Dataset	Description
Enron	Includes: Persons, Email Categories, Sentiment, Email Content
KIT (dead)	
Weibo	Includes: Persons, Tweets, Followers; **Excludes: Tweet Content**
Digg	Includes: Persons, Stores, Followers, Votes; **Excludes: Content**
Slashdot	Includes: Persons, Votes; **Excludes: Content**
IMDB	Actor movie network; Content is implicitly defined
WIKI-RFA	Wikipedia Adminitrator Election; Network of Voters and Votees. Links are votes and vote comments
FB-wosn	User friendship links and User posts on users walls; **Excludes: Content**

Dataset	Description
TweetUM (dead)	Twitter Tweets, User Profiles and Followers; Includes: Content
Reddit Pushift	User Submissions and Posts on Subreddits; With timestamps
Bitcoin Trust Network	Network Nodes and peer Ratings; With timestamps
LastFM1k	User - Song Listen histories; With timestamps
MovieLens25M	Users and Movie Ratings; With timestamps
Memetracker	
Rumor Detection	Rumor Detection over Varying Time Windows; Twitter data; With timestamps

3 RESEARCH QUESTIONS

4 APPROACH

PLANNING

[1] Sitaram Asur, Srinivasan Parthasarathy, and Duygu Ucar. An event-based framework for characterizing the evolutionary behavior of interaction graphs. *ACM Transactions on Knowledge Discovery from Data*, 3(4):16:1–16:36, December 2009. ISSN 1556-4681. doi: 10.1145/1631162.1631164.

[2] Vincent D. Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, October 2008. ISSN 1742-5468. doi: 10.1088/1742-5468/2008/10/P10008.

[3] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating Embeddings for Modeling Multi-relational Data. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013.

[4] Michael M. Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: Going beyond Euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, July 2017. ISSN 1053-5888, 1558-0792. doi: 10.1109/MSP.2017.2693418.

[5] Rémy Cazabet, Hideaki Takeda, Masahiro Hamasaki, and F. Amblard. Using dynamic community detection to identify trends in user-generated content. *Social Network Analysis and Mining*, 2012. doi: 10.1007/s13278-012-0074-8.

[6] Michele Coscia, Fosca Giannotti, and Dino Pedreschi. A Classification for Community Discovery Methods in Complex Networks. *Statistical Analysis and Data Mining*, 4(5):512–546, October 2011. ISSN 19321864. doi: 10.1002/sam.10133.

[7] Narimene Dakiche, Fatima Benbouzid-Si Tayeb, Yahya Slimani, and Karima Benatchba. Tracking community evolution in social networks: A survey. *Information Processing & Management*, 56(3):1084–1102, May 2019. ISSN 0306-4573. doi: 10.1016/j.ipm.2018.03.005.

[8] Chris Diehl, Galileo Namata, and Lise Getoor. Relationship Identification for Social Network Discovery. pages 546–552, January 2007.

[9] Yuxiao Dong, Nitesh V. Chawla, and Ananthram Swami. Metapath2vec: Scalable Representation Learning for Heterogeneous Networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, pages 135–144, New York, NY, USA, August 2017. Association for Computing Machinery. ISBN 978-1-4503-4887-4. doi: 10.1145/3097983.3098036.

[10] Hossein Fani, Eric Jiang, Ebrahim Bagheri, Feras Al-Obeidat, Weichang Du, and Mehdi Kargar. User community detection via embedding of social network structure and temporal content. *Information Processing & Management*, 57 (2):102056, March 2020. ISSN 03064573. doi: 10.1016/j.ipm.2019.102056.

[11] Elahe Ghalebi, Baharan Mirzasoleiman, Radu Grosu, and Jure Leskovec. Dynamic Network Model from Partial Observations. *arXiv:1805.10616 [cs, stat]*, February 2019.

[12] Amir Ghasemian, Pan Zhang, Aaron Clauset, Cristopher Moore, and Leto Peel. Detectability thresholds and optimal algorithms for community structure in dynamic networks. *Physical Review X*, 6(3):031005, July 2016. ISSN 2160-3308. doi: 10.1103/PhysRevX.6.031005.

[13] Clara Granell, Richard K. Darst, Alex Arenas, Santo Fortunato, and Sergio Gómez. Benchmark model to assess community structure in evolving networks. *Physical Review E*, 92(1):012805, July 2015. ISSN 1539-3755, 1550-2376. doi: 10.1103/PhysRevE.92.012805.

[14] Derek Greene, Dónal Doyle, and Pádraig Cunningham. Tracking the Evolution of Communities in Dynamic Social Networks. In *2010 International Conference on Advances in Social Networks Analysis and Mining*, pages 176–183, August 2010. doi: 10.1109/ASONAM.2010.17.

[15] Aditya Grover and Jure Leskovec. Node2vec: Scalable Feature Learning for Networks. *arXiv:1607.00653 [cs, stat]*, July 2016.

[16] Qian Guo, Lei Zhang, Bin Wu, and Xuelin Zeng. Dynamic community detection based on distance dynamics. In *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 329–336, August 2016. doi: 10.1109/ASONAM.2016.7752254.

[17] William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive Representation Learning on Large Graphs. *arXiv:1706.02216 [cs, stat]*, September 2018.

[18] William L. Hamilton, Rex Ying, and Jure Leskovec. Representation Learning on Graphs: Methods and Applications. *arXiv:1709.05584 [cs]*, April 2018.

[19] Jialin He and Duanbing Chen. A fast algorithm for community detection in temporal network. *Physica A: Statistical Mechanics and its Applications*, 429, July 2015. doi: 10.1016/j.physa.2015.02.069.

[20] Mingqing Huang, Qingshan Jiang, Qiang Qu, Lifei Chen, and Hui Chen. Information fusion oriented heterogeneous social network for friend recommendation via community detection. *Applied Soft Computing*, 114:108103, January 2022. ISSN 1568-4946. doi: 10.1016/j.asoc.2021.108103.

[21] Yuting Jia, Qinqin Zhang, Weinan Zhang, and Xinbing Wang. CommunityGAN: Community Detection with Generative Adversarial Nets. *arXiv:1901.06631 [cs]*, December 2019.

[22] Yoonsuk Kang, Jun-Seok Lee, Won-Yong Shin, and Sang-Wook Kim. Community reinforcement: An effective and efficient preprocessing method for accurate community detection. *Knowledge-Based Systems*, page 107741, November 2021. ISSN 0950-7051. doi: 10.1016/j.knosys.2021.107741.

[23] Arzum Karataş and Serap Şahin. A Review on Social Bot Detection Techniques and Research Directions. October 2017.

[24] Thomas N. Kipf and Max Welling. Variational Graph Auto-Encoders. *arXiv:1611.07308 [cs, stat]*, November 2016.

[25] Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks. *arXiv:1609.02907 [cs, stat]*, February 2017.

[26] Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks. *arXiv:1609.02907 [cs, stat]*, February 2017.

[27] Andrea Lancichinetti and Santo Fortunato. Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. *Physical Review E*, 80(1):016118, July 2009. ISSN 1539-3755, 1550-2376. doi: 10.1103/PhysRevE.80.016118.

[28] Andrea Lancichinetti, Santo Fortunato, and Filippo Radicchi. Benchmark graphs for testing community detection algorithms. *Physical Review E*, 78(4):046110, October 2008. ISSN 1539-3755, 1550-2376. doi: 10.1103/PhysRevE.78.046110.

[29] Andrea Lancichinetti, Santo Fortunato, and Janos Kertesz. Detecting the overlapping and hierarchical community structure of complex networks. *New Journal of Physics*, 11(3):033015, March 2009. ISSN 1367-2630. doi: 10.1088/1367-2630/11/3/033015.

[30] Xin Liu, Weichu Liu, Tsuyoshi Murata, and Ken Wakita. Community Detection in Multi-Partite Multi-Relational Networks Based on Information Compression. *New Generation Computing*, 34(1):153–176, March 2016. ISSN 1882-7055. doi: 10.1007/s00354-016-0206-1.

[31] Linhao Luo, Yixiang Fang, Xin Cao, Xiaofeng Zhang, and Wenjie Zhang. Detecting Communities from Heterogeneous Graphs: A Context Path-based Graph Neural Network Model. *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 1170–1180, October 2021. doi: 10.1145/3459637.3482250.

[32] Lijia Ma, Yutao Zhang, Jianqiang Li, Qiuzhen Lin, Qing Bao, Shanfeng Wang, and Maoguo Gong. Community-aware dynamic network embedding by using deep autoencoder. *Information Sciences*, 519:22–42, May 2020. ISSN 0020-0255. doi: 10.1016/j.ins.2020.01.027.

[33] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. *arXiv:1301.3781 [cs]*, September 2013.

[34] Mohammad Mosadegh and Mehdi Behboudi. Using Social Network Paradigm for Developing a Conceptual Framework in CRM. *Australian Journal of Business and Management Research*, 1:63–71, August 2011. doi: 10.52283/NSWRCA.AJBMR.20110104A06.

[35] Nairouz Mrabah, Mohamed Bouguessa, Mohamed Fawzi Touati, and Riadh Ksantini. Rethinking Graph Auto-Encoder Models for Attributed Graph Clustering. *arXiv:2107.08562 [cs]*, July 2021.

[36] Peter J. Mucha, Thomas Richardson, Kevin Macon, Mason A. Porter, and Jukka-Pekka Onnela. Community Structure in Time-Dependent, Multiscale, and Multiplex Networks. November 2009. doi: 10.1126/science.1184819.

[37] M. E. J. Newman. Fast algorithm for detecting community structure in networks. *Physical Review E*, 69(6):066133, June 2004. ISSN 1539-3755, 1550-2376. doi: 10.1103/PhysRevE.69.066133.

[38] Giang Hoang Nguyen, John Boaz Lee, Ryan A. Rossi, Nesreen K. Ahmed, Eunyee Koh, and Sungchul Kim. Continuous-Time Dynamic Network Embeddings. In *Companion of the The Web Conference 2018 on The Web Conference 2018 - WWW '18*, pages 969–976, Lyon, France, 2018. ACM Press. ISBN 978-1-4503-5640-4. doi: 10.1145/3184558.3191526.

[39] Gergely Palla, Albert-László Barabási, and Tamás Vicsek. Quantifying social group evolution. *Nature*, 446(7136): 664–667, April 2007. ISSN 1476-4687. doi: 10.1038/nature05670.

[40] Leto Peel, Daniel B. Larremore, and Aaron Clauset. The ground truth about metadata and community detection in networks. *Science Advances*, 3(5):e1602548, 2017. doi: 10.1126/sciadv.1602548.

[41] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global Vectors for Word Representation. In *EMNLP*, volume 14, pages 1532–1543, January 2014. doi: 10.3115/v1/D14-1162.

[42] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. DeepWalk: Online Learning of Social Representations. *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710, August 2014. doi: 10.1145/2623330.2623732.

[43] Giulio Rossetti. ANGEL: Efficient, and effective, node-centric community discovery in static and dynamic networks. *Applied Network Science*, 5(1):26, June 2020. ISSN 2364-8228. doi: 10.1007/s41109-020-00270-6.

[44] Giulio Rossetti and Rémy Cazabet. Community Discovery in Dynamic Networks: A Survey. *ACM Computing Surveys*, 51(2):35:1–35:37, February 2018. ISSN 0360-0300. doi: 10.1145/3172867.

[45] Giulio Rossetti, Luca Pappalardo, and Salvatore Rinzivillo. A Novel Approach to Evaluate Community Detection Algorithms on Ground Truth. In Hocine Cherifi, Bruno Gonçalves, Ronaldo Menezes, and Roberta Sinatra, editors, *Complex Networks VII: Proceedings of the 7th Workshop on Complex Networks CompleNet 2016*, Studies in Computational Intelligence, pages 133–144. Springer International Publishing, Cham, 2016. ISBN 978-3-319-30569-1. doi: 10.1007/978-3-319-30569-1_10.

[46] Benedek Rozemberczki, Ryan Davies, Rik Sarkar, and Charles Sutton. GEMSEC: Graph Embedding with Self Clustering. *arXiv:1802.03997 [cs]*, July 2019.

[47] Marcel Salathé and James H. Jones. Dynamics and Control of Diseases in Networks with Community Structure. *PLOS Computational Biology*, 6(4):e1000736, April 2010. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1000736.

[48] Hamed Sarvari, Ehab Abozinadah, Alex Mbaziira, and Damon Mccoy. Constructing and Analyzing Criminal Networks. In *2014 IEEE Security and Privacy Workshops*, pages 84–91, May 2014. doi: 10.1109/SPW.2014.22.

[49] Xing Su, Shan Xue, Fanzhen Liu, Jia Wu, Jian Yang, Chuan Zhou, Wenbin Hu, Cecile Paris, Surya Nepal, Di Jin, Quan Z. Sheng, and Philip S. Yu. A Comprehensive Survey on Community Detection with Deep Learning. *arXiv:2105.12584 [cs]*, May 2021.

[50] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. LINE: Large-scale Information Network Embedding. *Proceedings of the 24th International Conference on World Wide Web*, pages 1067–1077, May 2015. doi: 10.1145/2736277.2741093.

[51] Wei Wang, Feng Xia, Hansong Nie, Zhikui Chen, Zhiguo Gong, Xiangjie Kong, and Wei Wei. Vehicle Trajectory Clustering Based on Dynamic Representation Learning of Internet of Vehicles. *IEEE Transactions on Intelligent Transportation Systems*, PP:1–10, June 2020. doi: 10.1109/TITS.2020.2995856.

[52] Yi Wang, Bin Wu, and Nan Du. Community Evolution of Social Network: Feature, Algorithm and Model. *arXiv:0804.4356 [physics]*, April 2008.

[53] Zhen Wang, Chunyu Wang, Chao Gao, Xuelong Li, and Xianghua Li. An evolutionary autoencoder for dynamic community detection. *Science China Information Sciences*, 63(11):212205, September 2020. ISSN 1869-1919. doi: 10.1007/s11432-020-2827-9.

[54] Xiaodong Wu, Weizhe Lin, Zhilin Wang, and Elena Rastorgueva. Author2Vec: A Framework for Generating User Embedding. *arXiv:2003.11627 [cs, stat]*, March 2020.

[55] Carl Yang, Mengxiong Liu, Zongyi Wang, Liyuan Liu, and Jiawei Han. Graph Clustering with Dynamic Embedding. *arXiv:1712.08249 [physics]*, December 2017.

[56] Jaewon Yang and Jure Leskovec. Community-Affiliation Graph Model for Overlapping Network Community Detection. In *2012 IEEE 12th International Conference on Data Mining*, pages 1170–1175, December 2012. doi: 10.1109/ICDM.2012.139.

[57] Fanghua Ye, Chuan Chen, and Zibin Zheng. Deep Autoencoder-like Nonnegative Matrix Factorization for Community Detection. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, CIKM '18, pages 1393–1402, New York, NY, USA, October 2018. Association for Computing Machinery. ISBN 978-1-4503-6014-2. doi: 10.1145/3269206.3271697.

[58] Ying Yin, Yuhai Zhao, He Li, and Xiangjun Dong. Multi-objective evolutionary clustering for large-scale dynamic community detection. *Information Sciences*, 549:269–287, March 2021. ISSN 0020-0255. doi: 10.1016/j.ins.2020.11.025.