Boost Dynamic Community Detection via Exploiting Member Transition Information

Zhongyu Pan[1], Junwei Cheng[1], Weixiong Liu[1], Chaobo He[1(✉)], Quanlong Guan[2], and Xuequan Lin[1]

[1] School of Computer Science, South China Normal University, Guangzhou, China
{zhongyupan,jung,weixiongliu,hechaobo,xuequanlin}@m.scnu.edu.cn
[2] College of Information Science and Technology, Jinan University, Guangzhou, China
gql@jnu.edu.cn

Abstract. Dynamic networks, characterized by their continuously changing nodes and edges, play a vital role in various real-world applications. Community detection in dynamic networks, also known as dynamic community detection, is an active research topic. However, most existing methods still face two key problems. Firstly, community membership evolves over time, but few methods effectively exploit this dynamic information. Secondly, they completely rely on the assumption of temporal smoothness, often overlooking the abrupt changes of local topology, referred to as local non-smoothness. In view of this, we propose a contrastive learning based model DCDMT for dynamic community detection. Specifically, for the first problem, we introduce the detection mechanism of community member transition, which can distinguish stable and unstable community members between snapshots. By treating these nodes as the corresponding positive and negative samples, we develop the contrastive learning framework to optimize the dynamic community detection model. For the second problem, DCDMT uses the Hilbert-Schmidt Independence Criterion to improve node independence over time. This strategy helps capture local non-smooth features of nodes across snapshots. Experimental results on real-world datasets demonstrate that DCDMT outperforms state-of-the-art methods.

Keywords: Dynamic networks · Community member transition · Community detection · Contrastive learning.

1 Introduction

Unlike static networks, dynamic networks are characterized by temporal changes of their nodes and edges [2]. For example, with the frequent occurrence of interaction behaviors, the relationships among users in online social networks often evolve over time [10]. Recently, dynamic networks have attracted increasing attention from academia and industry due to their high research value, and community detection in them, also known as dynamic community detection, is a prominent research topic. Essentially, dynamic community detection is a node clustering problem. It aims to identify meaningful node clusters in the dynamic network, meeting the requirement that the nodes within the same cluster have denser links than those in different clusters [6].

In the last decades, benefiting from the powerful representation learning ability to graph structured data, Graph Neural Networks (GNNs) have been widely used in dynamic community detection, and various GNN-based methods have been continuously presented. For instance, EvolveGCN [12], CTGCN [9] and DGCN [3] modify the GNN message-passing mechanism to incorporate temporal information, enabling them to adapt to temporal changes in community structures. By introducing the contrastive learning framework, CGC [13] and CLDG [16] bring positive node samples closer and negative node samples farther away, thereby intuitively facilitating nodes clustering in the dynamic representation learning space. Besides these methods, VGAE [7], as a static network method using a deep autoencoder architecture, also shows great potential in the field of community detection. Subsequent related studies such as VGRNN [5] and VGRGMM [8] have extended it to dynamic networks, which are able to deal with the uncertainty of dynamic community detection via the probability inference framework. Based on them, DyVGRNN [11] and DySAT [15] incorporate attention mechanisms into encoders, thereby improving the ability of feature aggregation. Overall, these existing methods have demonstrated varying levels of effectiveness, but they still face two significant challenges:

- **Exploiting community member transition information.** Although some of the aforementioned works (e.g., CTGCN and EvolveGCN) have tried to use temporal information to improve the performance of community detection, they have ignored the evolution of community members in dynamic networks. As reported in [14] and [1], the evolution of the community is useful to understand network structure. The transition information of stable and unstable community members between snapshots can serve as supervision information to guide the dynamic community detection process.
- **Capturing local non-smoothness between different snapshots.** Generally, most dynamic community detection methods are based on the temporal smoothness assumption that the topology structure does not change abruptly, and evolves smoothly instead. However, real-world dynamic networks often undergo drastic changes in some local areas. For example, in social networks, the following relations of certain users may increase significantly, or some interest groups (i.e., communities) suddenly disappear in a short period. In other words, some nodes in different snapshots exhibit local non-smoothness. Effectively capturing this type of information can help to learn more discriminative node representations, facilitating the boost of the performance of dynamic community detection.

To address the aforementioned problems, in this paper we propose a novel dynamic community detection model called DCDMT. This model designs the detection mechanism of community member transition, which can identify stable nodes and unstable nodes in the context of community evolutions. To capture local non-smoothness, we employ the Hilbert-Schmidt Independence Criterion

(HSIC) [4] to assess the similarity between two snapshots. This method effectively identifies the independence information of nodes between different snapshots. By further developing the contrastive learning framework incorporating community member transition and local non-smoothness information, DCDMT can learn more beneficial node representations for dynamic community detection. In summary, our main contributions are as follows:

- To effectively capture the dynamics of community structures, we propose a mechanism to detect community member transitions in dynamic networks. This mechanism identifies stable and unstable nodes, enabling the direct incorporation of community member transition information into the community detection process.
- To address the challenges posed by abrupt changes in dynamic networks, we integrate HSIC with a contrastive learning framework. This effectively captures the local non-smooth features, and enables the learning of robust, discriminative node representations for dynamic community detection.
- Experimental results show that DCDMT outperforms the baselines on five real-world datasets, highlighting its superior performance.

2 METHODOLOGY

2.1 Notations and Problem Statement

Without loss of generality, in this paper we focus on the undirected and unweighted dynamic network, denoted as $\mathbf{G}$, and model it as a series of snapshots $\mathbf{G} = (\mathbf{G}^{(1)}, \mathbf{G}^{(2)}, \ldots, \mathbf{G}^{(T)})$, where T is the total number of snapshots and each snapshot at time t is represented as $\mathbf{G}^{(t)} = (\mathbf{V}^{(t)}, \mathbf{E}^{(t)}, \mathbf{X}^{(t)})$. Here, $\mathbf{V}^{(t)}$ and $\mathbf{E}^{(t)}$ denote the sets of nodes and edges of $\mathbf{G}^{(t)}$, respectively. The adjacency matrix $\mathbf{A}^{(t)}$ is employed to represent the topology of $\mathbf{G}^{(t)}$, with $\mathbf{A}^{(t)}_{ij} = 1$ when nodes i and j are connected, and $\mathbf{A}^{(t)}_{ij} = 0$ otherwise. Node features are denoted as matrix $\mathbf{X}^{(t)}$, where the i-th row represents the features of node i at the t-th snapshot. The matrix $\mathbf{Z}^{(t)}$ represents the dynamic network embeddings at time t, where the i-th row corresponds to the embedding of node i. In DCDMT, each node v at time t is assigned a community label $c^{(t)}_v$, indicating the community that it belongs to at this time. In the context of a dynamic network, community structure may evolve over time, leading to the transition of community members. To represent this information, we define the following stable nodes set and unstable nodes set from the perspective of a given node v at time t,

$$\mathrm{StableNodes}(v^{(t)}) = \{u^{(t)} \in \mathbf{V}^{(t)} \mid c^{(t)}_u = c^{(t)}_v \text{ and } c^{(t')}_u = c^{(t')}_v\}, \tag{1}$$

$$\begin{aligned}\mathrm{UnstableNodes}(v^{(t)}) = \{u^{(t)} \in \mathbf{V}^{(t)} \mid &(c^{(t)}_u = c^{(t)}_v \text{ and } c^{(t')}_u \neq c^{(t')}_v) \\ &\text{or } (c^{(t)}_u \neq c^{(t)}_v \text{ and } c^{(t')}_u = c^{(t')}_v)\},\end{aligned} \tag{2}$$

where $t \neq t'$. Based on these, the objective of dynamic community detection here is to partition nodes in every snapshot into K communities $\mathbf{C}^{(t)} = \{C^{(t)}_1, \ldots, C^{(t)}_K\}$, such that the nodes within the same community are close to each other, while those in different communities are far apart.

2.2 DCDMT

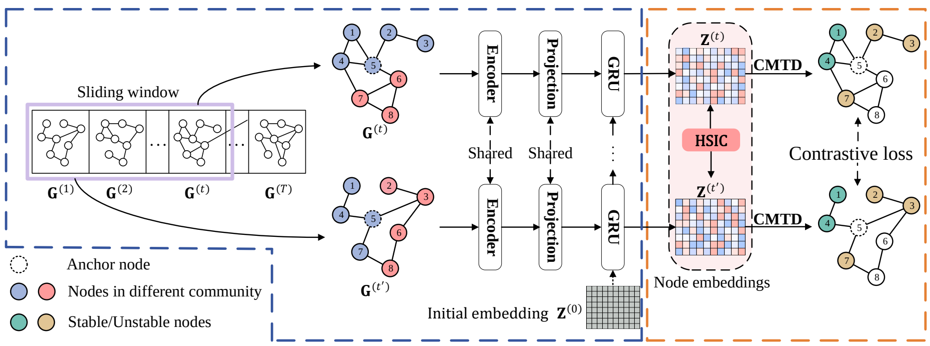


Fig. 1: The framework of DCDMT.

(1) Overview. Fig. 1 illustrates the framework of DCDMT, which consists of two parts. The left part is used to transform raw graph data into node embeddings, and the right part is responsible for optimizing the model. The embedding learning process involves three key components: the base encoder layer, the projection head layer and the Gated Recurrent Unit (GRU) layer. The raw graph data is first processed by the base encoder and the projection head to generate the basic embeddings. To ensure that each snapshot's embedding captures historical information, we introduce the GRU layer, which integrates temporal dependencies across snapshots. In order to capture the local non-smoothness in dynamic networks, we use HSIC to constrain the similarities between snapshots. In the right part, DCDMT devises the Community Member Transition Detection (CMTD) mechanism to improve the selection of positive and negative sample pairs for contrastive learning, enabling DCDMT to fully exploit member transition information. By using the joint loss raised by contrastive learning and HSIC modules, DCDMT is trained to learn the final node embeddings for dynamic community detection.

(2) Base encoder layer. In DCDMT, each snapshot is initially processed by the base encoder layer. We employ a multi-layer GCN as the base encoder, which can effectively captures the complex relationships among nodes by projecting them into the latent space. The l-th GCN layer is defined as $\mathbf{H}_{l+1} = \sigma(\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{H}_l\mathbf{W}_l)$, where $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ and $\tilde{\mathbf{D}}$ are respectively the adjacency matrix and degree matrix of the input network with self-loops, $\mathbf{W}_l$ is a learnable weight matrix, $\mathbf{H}_l$ is the output embedding matrix at layer l, and σ is the nonlinear activation function (e.g., ReLU used here). We set $\mathbf{H}_0 = \mathbf{X}$, and set $\mathbf{X}$ to the identity matrix $\mathbf{I}$ (i.e., $\mathbf{X} = \mathbf{I}$) when node features are not available. Note that the base encoders for each snapshot share parameters. This mechanism is based on the smoothing principle between snapshots, ensuring that structural information learned from one snapshot remains applicable to subsequent snapshots. For the snapshot at time t, its node embeddings outputted by the corresponding base encoder are denoted as the matrix $\mathbf{H}^{(t)}$.

(3) Projection head layer. To enhance the ability of model to distinguish different communities, we introduce a projection head layer that performs a nonlinear transformation on $\mathbf{H}^{(t)}$, mapping it to a new space. This mapping projects the learned representations into a space where inter-community differences are more distinct, facilitating the separation of community features. The projection head for the snapshot at time t is defined as $\mathbf{P}^{(t)} = f_{\text{proj}}(\mathbf{H}^{(t)})$, where $f_{\text{proj}}(\cdot)$ comprises a Multilayer Perceptron (MLP) with two layers, each followed by the L_2 normalization and LeakyReLU activation function, and $\mathbf{P}^{(t)}$ is the transformed node embeddings matrix at time t.

(4) GRU layer. In dynamic networks, the node embedding of the current snapshot is influenced by the previous snapshot. Specifically, if an unstable node undergoes a community transition in the previous snapshot, its representation in the current snapshot should significantly differ from the representations of other nodes that belonged to the same community in the previous snapshot. Conversely, for stable nodes that remain within the same community, the differences between their embeddings should be minimized. To capture the impact of both stable and unstable nodes across snapshots on the current snapshot, we introduce a GRU with shared parameters to process each snapshot's embedding, respectively. Specifically, the embedding $\mathbf{P}^{(t)}$ at time t is processed through the following steps to obtain its final representation $\mathbf{Z}^{(t)}$:

$$\mathbf{R}^{(t)} = \sigma(\mathbf{W}_\mathbf{r}[\mathbf{Z}^{(t-1)}, \mathbf{P}^{(t)}]), \tag{3}$$

$$\mathbf{Q}^{(t)} = \sigma(\mathbf{W}_\mathbf{u}[\mathbf{Z}^{(t-1)}, \mathbf{P}^{(t)}]), \tag{4}$$

$$\tilde{\mathbf{Z}}^{(t)} = \tanh(\mathbf{W}_\mathbf{h}[\mathbf{R}^{(t)} \odot \mathbf{Z}^{(t-1)}, \mathbf{P}^{(t)}]), \tag{5}$$

$$\mathbf{Z}^{(t)} = (\mathbb{1} - \mathbf{Q}^{(t)}) \odot \mathbf{Z}^{(t-1)} + \mathbf{Q}^{(t)} \odot \tilde{\mathbf{Z}}^{(t)}, \tag{6}$$

where $\mathbb{1}$ is an all-ones matrix. $[,]$ indicates the concatenation operation and $\odot$ represents the Hadamard product. $\mathbf{W}_\mathbf{r}$, $\mathbf{W}_\mathbf{u}$ and $\mathbf{W}_\mathbf{h}$ are the trainable weight matrices. Note that $\mathbf{Z}^{(0)}$ is initialized as a zero matrix.

(5) Community member transition detection (CMTD). In order to identify the sets of stable and unstable nodes across different snapshots, we designed the CMTD mechanism. The motivation behind its design is to analyze the transition patterns of community membership over time. This transition information can serve as an additional supervisory signal in contrastive learning, enhancing the model's ability to capture the dynamic behavior of the network. Specifically, for any two different snapshots at time t and t', CMTD respectively applies the k-means clustering algorithm to $\mathbf{Z}^{(t)}$ and $\mathbf{Z}^{(t')}$ to generate pseudo-labels that indicate each node's community. After performing the one-hot encoding, the community affiliation matrix $\mathbf{F}^{(t)}$ and $\mathbf{F}^{(t')}$ are obtained. Then, multiplying $\mathbf{F}^{(t)}$ ($\mathbf{F}^{(t')}$) by its transpose yields the co-community indicator matrix $\mathbf{O}^{(t)}$ ($\mathbf{O}^{(t')}$). This matrix clearly represents the community relationships of pairwise nodes in the same snapshot, where 1/0 indicates that two corresponding nodes belong to the same/different communities. Finally, the stable nodes matrix $\mathbf{S}^{(t,t')}$ and unstable nodes matrix $\mathbf{U}^{(t,t')}$ generated from two snapshots at time t and t' can be respectively obtained by $\mathbf{S}^{(t,t')} = (\mathbf{O}^{(t)} \odot \mathbf{O}^{(t')}) - \mathbf{I}$ and $\mathbf{U}^{(t,t')} = \mathbf{O}^{(t)} \oplus \mathbf{O}^{(t')}$, where $\oplus$ denotes the element-wise exclusive-or operation for the matrices $\mathbf{O}^{(t)}$ and $\mathbf{O}^{(t')}$.

(6) Capturing local non-smoothness. To capture the local non-smoothness discussed in Section 1, we introduce Hilbert-Schmidt Independence Criterion (HSIC) to measure the dependence between embeddings of two different snapshots:

$$\mathrm{HSIC}(\mathbf{Z}^{(t)}, \mathbf{Z}^{(t')}) = \frac{1}{(N-1)^2}\mathrm{tr}(\mathbf{KMLM}), \tag{7}$$

where $\mathrm{tr}(\cdot)$ is the matrix trace, $\mathbf{K}$ and $\mathbf{L}$ are respectively the gram matrices of embeddings $\mathbf{Z}^{(t)}$ and $\mathbf{Z}^{(t')}$, $\mathbf{M}$ is the centering matrix defined as $\mathbf{I} - \frac{1}{n}\mathbb{1}$, and N is the number of nodes in a snapshot. By minimizing the HSIC, the embedding similarity between different snapshots can be constrained to ensure their respective differences. This will allow the embedding of each snapshot to preserve local non-smoothness information.

(7) Overall loss. We introduce the contrastive learning to self-supervise the process of dynamic community detection. Considering the given dynamic network is composed of multiple snapshots (treated as views), we design two forms of contrastive learning: intra-view and inter-view. Intra-view contrastive learning focuses on the nodes within a single snapshot: treating a node and its immediate neighbors as positive samples and treating non-neighboring nodes as negative samples. This strategy can enhance the cohesiveness of community structures. Inter-view contrastive learning focuses on stable and unstable nodes, which are identified via conducting CMTD across multiple snapshots. Specifically, the given node and its stable nodes are treated as positive samples, and the given node and its unstable nodes are treated as negative samples. Based on these, from the perspective of a given node i, we define the contrastive loss for any two snapshots at time t and t' as:

$$\mathcal{L}_{\mathrm{cont}}(\mathbf{Z}_i^{(t)}, \mathbf{Z}_i^{(t')}) = \log \frac{pos}{pos + neg}, \tag{8}$$

where pos and neg are respectively computed via:

$$pos = \underbrace{\sum_{j \in \mathcal{N}_i^{(t)}} e^{\theta(\mathbf{Z}_i^{(t)}, \mathbf{Z}_j^{(t)})/\tau}}_{\text{intra-view positive pairs}} + \underbrace{e^{\theta(\mathbf{Z}_i^{(t)}, \mathbf{Z}_i^{(t')})/\tau} + \sum_{j \in \mathcal{N}_i^{(t)}} (e^{\theta(\mathbf{Z}_i^{(t)}, \mathbf{Z}_j^{(t')})/\tau} \cdot \mathbf{S}_{ij}^{(t,t')})}_{\text{inter-view positive pairs}}, \tag{9}$$

$$neg = \underbrace{\sum_{j \notin \{\mathcal{N}_i^{(t)} \cup i\}} e^{\theta(\mathbf{Z}_i^{(t)}, \mathbf{Z}_j^{(t)})/\tau}}_{\text{intra-view negative pairs}} + \underbrace{\sum_{j \in \mathcal{N}_i^{(t)}} (e^{\theta(\mathbf{Z}_i^{(t)}, \mathbf{Z}_j^{(t')})/\tau} \cdot \mathbf{U}_{ij}^{(t,t')})}_{\text{inter-view negative pairs}}, \tag{10}$$

where τ is a temperature parameter that controls the concentration of the distribution, $\mathcal{N}_i^{(t)}$ is the set of neighbor nodes of node i in the t-th snapshot. To improve the efficiency, we set the size w of the sliding window to determine the number of snapshots for contrastive learning, and the total contrastive learning objective is defined as:

$$\mathcal{L}_{\text{cont}} = \frac{1}{2N(T-w+1)(w-1)} \sum_{i=1}^{N} \sum_{t=1}^{T-w+1} \sum_{t'=t+1}^{t+w-1} (\mathcal{L}_{\text{cont}}(\mathbf{Z}_i^{(t)}, \mathbf{Z}_i^{(t')}) + \mathcal{L}_{\text{cont}}(\mathbf{Z}_i^{(t')}, \mathbf{Z}_i^{(t)})), \tag{11}$$

In addition to the contrastive loss above, we also leverage the previously mentioned HSIC loss to drive node embeddings to preserve local non-smoothness:

$$\mathcal{L}_{\text{HSIC}} = \sum_{t=1}^{T-w+1} \sum_{t'=t+1}^{t+w-1} \text{HSIC}(\mathbf{Z}^{(t)}, \mathbf{Z}^{(t')}), \tag{12}$$

By combing $\mathcal{L}_{\text{cont}}$ and $\mathcal{L}_{\text{HSIC}}$, the joint loss function for DCDMT is given by $\mathcal{L} = \mathcal{L}_{\text{cont}} + \alpha\mathcal{L}_{\text{HSIC}}$, where α is a hyperparameter used to balance the contribution of $\mathcal{L}_{\text{HSIC}}$.

3 EXPERIMENTS

3.1 Experiment setup

(1) Datasets and baselines. We select five real-world dynamic networks with ground-truth community divisions to validate the effectiveness of DCDMT, including High School, Java, Cellphone Call, Cora and DBLP-T, which are respectively from [3] and [13]. These networks include diverse relationship types, such as social interactions, collaborations and academic citations. Their detailed statistics are provided in Table 1. We select various representative and state-of-the-art methods as baselines, including network representation learning methods and end-to-end dynamic community detection methods. They are VGAE [7], CLDG [16], DGCN [3], DySAT [15], VGRNN [5], DyVGRNN [11] and VGRGMM [8].

Table 1: Statistics of dynamic networks.

Datasets	#Nodes	#Edges	T	K
High School	327	188,508	9	7
Java	376	40,915	20	20
Cellphone Call	400	9,834	10	20
Cora	2,708	5,429	5	7
DBLP-T	6,942	168,124	14	2

(2) Evaluation metrics and parameter settings. We utilize four widely used evaluation metrics to assess the performance of various models for dynamic community detection: Normalized Mutual Information (NMI), F1-score (F1), Accuracy (ACC) and Adjusted Rand Index (ARI). For our proposed method DCDMT, we set the sliding window size w to 5 to balance the performance and the runtime cost. A two-layer GCN is used as the base encoder. The learning rate is set to $4e^{-3}$ and the weight decay is set to $5e^{-4}$. For the hyperparameter α, we set it to $1e^{-4}$. As for baselines, we follow the optimal parameter settings suggested by them.

3.2 Results and analysis

(1) Comparison with baselines. We compared the performance of DCDMT with seven state-of-the-art methods across five real-world dynamic networks. Table 2 summarizes the average performance of these models across five datasets. Our observations and analysis are as follows:

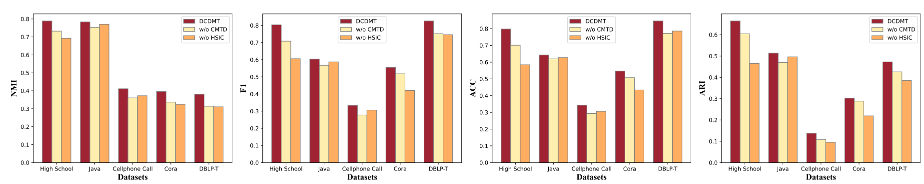
- DCDMT achieves excellent performance in most cases. Specifically, it performs the best on the Cora, High School, and DBLP-T datasets. On the Cora dataset, DCDMT's performance improvement is particularly notable, with NMI, F1, ACC and ARI increasing by 5.2%, 2.27%, 0.92% and 0.34%, respectively, compared to the second-best method.
- On datasets with significant community member transitions, such as DBLP-T, DCDMT demonstrates a clear advantage, achieving improvements of 2.39% in NMI and 1.38% in ARI. This advantage arises that the proposed CMTD mechanism enables the model to differentiate between stable and unstable nodes, translating these transition patterns into supervision signals for contrastive learning. In contrast, CLDG, which is also based on contrastive learning, fails to capture community member transitions effectively. As a result, its clustering performance is much lower than that of DCDMT. This highlights how CMTD enhances the model's ability to adapt its dynamic representations to evolving network structures.
- Our model exhibits a slight disadvantage on Java, with a 1.01% and 1.69% decrease in ACC and ARI compared to the best-performing model. The primary reason for this is the relatively smooth dynamics of Java snapshots. In such cases, emphasizing the independence between snapshots may result in suboptimal performance, as the model might fail to capture the continuity in community structures. Specially, the lack of significant community member transitions reduces the effectiveness of CMTD module.

(2) Ablation study. To further show the effectiveness of CMTD module and HSIC module, we conduct the ablation experiments. Specifically, we first respectively remove CMTD and HSIC modules to produce two variants: w/o CMTD and w/o HSIC, and then conduct the performance comparisons with DCDMT. The results are shown in Fig. 2. As we can see, compared to other variants, our method achieves the best performance across all datasets, demonstrating the necessity of incorporating the CMTD and HSIC modules for dynamic community detection. When the CMTD module is removed, the performance degradation is

Table 2: The average performance (%) of snapshots on every dynamic network. The best results are highlighted in bold, and the second-best results are underlined.

Model	High School				Java				Cellphone Call				Cora				DBLP-T			
	NMI	F1	ACC	ARI	NMI	F1	ACC	ARI	NMI	F1	ACC	ARI	NMI	F1	ACC	ARI	NMI	F1	ACC	ARI
VGAE	69.25	76.25	75.57	57.9	64.63	48.51	55.32	38.38	32.88	25.9	27.52	8.18	29.18	43.59	43.49	13.35	2.79	48.53	63.91	3.83
CLDG	71.17	<u>80.08</u>	<u>78.93</u>	60.82	74.95	57.12	63.5	<u>51.57</u>	34.23	26.2	27.53	8.85	<u>34.41</u>	<u>53.29</u>	<u>53.74</u>	<u>29.89</u>	17.14	72.25	74.11	23.6
DGCN	71.67	65.61	67.79	53.54	75.25	58.11	62.66	50.58	37.39	26.81	31.73	13.56	17.82	29.89	34.65	9.02	13.11	50.84	51.15	22.8
DySAT	<u>74.3</u>	76.27	77.47	<u>64.45</u>	<u>77.58</u>	<u>59.4</u>	**65.27**	**53.05**	38.07	27.72	30.07	11.94	25.79	43.0	43.84	18.06	<u>35.67</u>	<u>82.06</u>	<u>84.43</u>	<u>45.87</u>
VGRNN	64.98	71.27	69.86	46.37	68.08	42.86	54.56	40.27	39.71	30.22	<u>33.45</u>	**14.01**	28.55	43.13	46.52	17.04	26.1	72.62	77.7	29.27
DyVGRNN	63.38	75.24	75.4	53.72	53.54	29.83	39.03	24.53	32.73	22.9	25.82	8.42	30.69	46.32	46.95	19.41	31.38	78.97	81.72	39.5
VGRGMM	68.58	75.02	75.27	56.18	69.15	53.23	58.59	44.2	<u>40.43</u>	<u>31.4</u>	31.81	13.33	33.95	45.11	48.92	25.24	33.09	77.16	76.75	35.81
DCDMT	**78.87**	**80.36**	**79.78**	**66.45**	**78.34**	**60.34**	<u>64.26</u>	51.36	**41.13**	**33.39**	**34.27**	<u>13.74</u>	**39.61**	**55.56**	**54.66**	**30.23**	**38.06**	**82.58**	**84.59**	**47.25**

particularly noticeable on the DBLP-T dataset. The main reason for this is the presence of significant community member transition information in DBLP-T, which CMTD is able to capture and use to help the model effectively adapt to the evolving community structures. Without the HSIC module, the performance drop is the most pronounced on High School. Compared to DCDMT, the differences in NMI, F1, ACC and ARI of w/o HSIC are 9.61%, 19.76%, 21.33% and 19.89%, respectively. This is due to the presence of local snapshot structure changes in High School. The introduction of HSIC significantly improves the model's ability to capture its local non-smoothness information.



(a) NMI (b) F1 (c) ACC (d) ARI

Fig. 2: The ablation studies on CMTD and HSIC modules. The average performance of snapshots on every dynamic network are presented here.

4 Conclusion

In this paper, we propose DCDMT, a model designed to address two key challenges for improving dynamic community detection: exploiting community member transition information and capturing local non-smoothness. Specifically, CMTD is designed to capture both stable and unstable nodes in the process of community member transition, providing useful self-supervised information for contrastive learning across snapshots. To capture local non-smoothness information, we apply HSIC module to emphasize the embeddings independence of different snapshots. By flexibly combing the losses of contrastive learning and HSIC, DCDMT can be optimized to accurately identify dynamic community structures. Extensive experiments on five real-world dynamic networks demonstrate the superiority of DCDMT.

Acknowledgments. This work was supported by the National Natural Science Foundation of China (62477016, 62077045, 62077028, 62377028), and Guangdong Basic and Applied Basic Research Foundation (2024A1515011758).

References

1. Cheng, J., et al.: Unveiling community structures in static networks through graph variational bayes with evolution information. Neurocomputing **576**, 127349 (2024)
2. Flossdorf, J., Jentsch, C.: Change detection in dynamic networks using network characteristics. IEEE Trans. Signal Inf. Process. Netw. **7**, 451–464 (2021)
3. Gao, C., Zhu, J., et al.: A novel representation learning for dynamic graphs based on graph convolutional networks. IEEE Trans. Cybern. (2022)
4. Gretton, A., Bousquet, O., Smola, A., Schölkopf, B.: Measuring statistical dependence with hilbert-schmidt norms. In: International conference on algorithmic learning theory. pp. 63–77. Springer (2005)
5. Hajiramezanali, E., et al.: Variational graph recurrent neural networks. Adv. Neural Inf. Process. Syst. **32** (2019)
6. He, C., Cheng, J., Chen, G., Guan, Q., Fei, X., Tang, Y.: Detecting communities with multiple topics in attributed networks via self-supervised adaptive graph convolutional network. Information Fusion **105**, 102254 (2024)
7. Kipf, T.N., Welling, M.: Variational graph auto-encoders. arXiv preprint arXiv:1611.07308 (2016)
8. Li, T., et al.: Exploring temporal community structure via network embedding. IEEE Trans. Cybern. (2022)
9. Liu, J., Xu, C., Yin, C., Wu, W., Song, Y.: K-core based temporal graph convolutional network for dynamic graphs. IEEE Transactions on Knowledge and Data Engineering **34**(8), 3841–3853 (2020)
10. Liu, Y., Xia, J., Zhou, S., Yang, X., Liang, K., Fan, C., Zhuang, Y., Li, S.Z., Liu, X., He, K.: A survey of deep graph clustering: Taxonomy, challenge, application, and open resource. arXiv preprint arXiv:2211.12875 (2022)
11. Niknam, G., Molaei, S., Zare, H., Pan, S., Jalili, M., Zhu, T., Clifton, D.: Dyvgrnn: Dynamic mixture variational graph recurrent neural networks. Neural Networks **165**, 596–610 (2023)
12. Pareja, A., Domeniconi, G., Chen, J., et al.: EvolveGCN: Evolving graph convolutional networks for dynamic graphs. In: Proc. of the AAAI Conf. on Artificial Intelligence. vol. 34, pp. 5363–5370 (2020)
13. Park, N., Rossi, R., Koh, E., et al.: CGC: Contrastive graph clustering for community detection and tracking. In: Proc. ACM Web Conf. pp. 1115–1126 (2022)
14. Rossetti, G., Cazabet, R.: Community discovery in dynamic networks: A survey. ACM Comput. Surv. **51**(2), 35:1–35:37 (2018)
15. Sankar, A., et al.: DySAT: Deep neural representation learning on dynamic graphs via self-attention networks. In: Proc. 13th Int. Conf. Web Search Data Mining. pp. 519–527 (2020)
16. Xu, Y., Shi, B., Ma, T., et al.: CLDG: Contrastive learning on dynamic graphs. In: 2023 IEEE 39th Int. Conf. Data Eng. (ICDE). pp. 696–707. IEEE (2023)