TrackGAE: Tracking Dynamic Community Evolution with Graph Autoencoders

Maroun Haddad and Mohamed Bouguessa

Department of Computer Science, University of Quebec at Montreal
Montreal, Quebec, Canada
haddad.maroun@uqam.ca, bouguessa.mohamed@uqam.ca

Abstract. Tracking the evolution of communities (or clusters) in dynamic networks is a critical challenge in numerous applications, including social network analysis, biological systems, and financial modeling. Existing methods primarily focus on node membership overlap while overlooking structural and attribute-based transformations, leading to inconsistencies when clusters undergo structural or attribute changes. To address these limitations, we propose TrackGAE, a two-phase deep learning framework that leverages graph autoencoders to generate temporal representations of clusters and construct evolutionary sequences that preserve community identity. In the first phase, a Temporal Graph Autoencoder extracts structural and attribute-aware cluster embeddings. In the second phase, a Clustering Graph Autoencoder refines these embeddings using a proposed Deep-Pruning mechanism to generate high-quality cluster sequences. TrackGAE captures node membership, attributes, and structure, enabling accurate tracking of dynamic clusters over time. Preliminary results on the Yelp dataset demonstrate the suitability of our approach.

Keywords: Deep Graph Representation Learning · Dynamic Graphs · Community Tracking.

1 Introduction

Network data structures provide a natural framework for modeling relationships between interacting entities across diverse domains, such as social networks, biological systems, and communication networks. While graph theory offers a rich set of tools for extracting insights from these structures, clustering and community detection remain fundamental for identifying meaningful patterns. However, most existing studies focus on static networks, overlooking the inherent dynamism of real-world networks [1]. The evolution of network topology and content leads to shifts in internal communities, necessitating robust methods for tracking their formation, transformation, and dissolution over time.

Understanding and analyzing the behavior of evolving communities is critical in many fields where group dynamics outweigh individual behaviors. Examples include tracking group interactions in social networks, modeling population-level

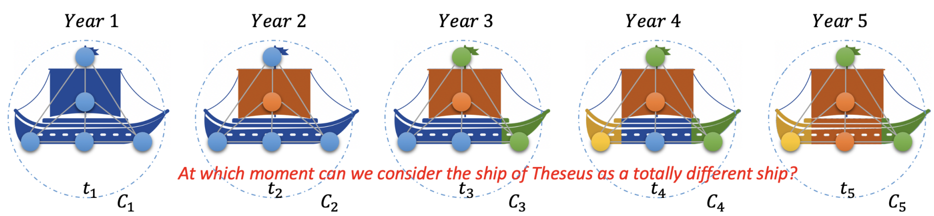

Fig. 1. The Ship of Theseus Paradox illustrated over time (t_1 to t_5).

immune responses to viruses, or identifying shifting trends in financial networks. However, tracking communities over time introduces unique challenges that extend beyond traditional static clustering methods.

A central challenge is what Cazabet and Rossetti [4] describe as the *preservation of identity problem*, also known as the *Ship of Theseus Paradox*. This paradox raises fundamental questions about identity: (i) If a ship gradually replaces its parts at each port, is it still the same ship upon reaching its final destination? (ii) If the removed parts are reconstructed elsewhere into a new ship, which one retains the original identity? This analogy directly applies to dynamic clusters in evolving networks. A community may lose and gain nodes across timesteps while maintaining a similar structure and attributes. As illustrated in Fig. 1, the Ship of Theseus undergoes gradual changes over time (t_1 to t_5), raising the question: at what point does it become a new entity rather than a continuation of the original? Similarly, in dynamic networks, identifying consistent sequences of clusters across time is essential for preserving the identity of evolving communities.

Various techniques have been proposed to address this problem [1, 2]. Different methods rely on distinct assumptions about cluster evolution, each with its own advantages and limitations. A common strategy involves dividing a network's temporal history into discrete snapshots (timesteps) and then matching clusters across consecutive timesteps to construct community evolution sequences. However, the effectiveness of this approach hinges on the criteria used for matching clusters.

Most existing benchmarks focus solely on node membership, neglecting structural and attribute-based transformations. While some models account for key events like the disappearance and reappearance of communities [3, 5], they remain shallow in their approach, relying exclusively on topological features such as node centrality [6], node membership overlap [7], and pairwise cluster similarity metrics [5]. These methods fail to incorporate richer feature representations that capture the structure, attributes, and temporal evolution of communities, leading to inconsistencies in tracking, particularly in cases where communities undergo structural or attribute changes while retaining a core set of nodes.

To address these shortcomings, we propose an approach that leverages representation learning to encode three essential aspects of a community:

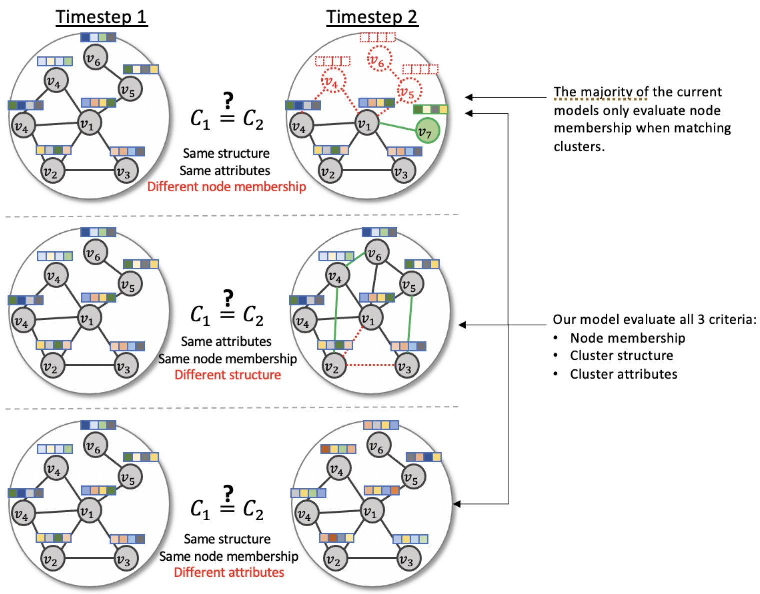

Fig. 2. The importance of node membership, structure, and attributes in tracking evolving communities.

1. Node Membership – preserving core members while allowing gradual transitions.
2. Cluster Structure – ensuring topological integrity over time.
3. Cluster Attributes – maintaining semantic consistency beyond mere node overlap.

Fig. 2 illustrates why these three factors are crucial for accurately tracking evolving communities. While traditional models rely only on node membership, they fail in cases where clusters undergo structural changes or attribute shifts. Our method addresses this by integrating structural, attribute, and membership features, ensuring that dynamic clusters are tracked more reliably over time. The following section details our approach.

2 Methodology

We propose TrackGAE, a two-phase deep learning framework designed to generate descriptive temporal representations of clusters and leverage them for accurate sequence generation. The first phase focuses on learning expressive cluster embeddings, while the second phase applies a clustering mechanism on a synthetic *Supergraph* to extract coherent evolutionary sequences. This ensures that evolving communities are tracked while preserving their node membership, structural integrity, and attribute consistency over time. As illustrated in Fig. 3, the TrackGAE framework operates as follows:

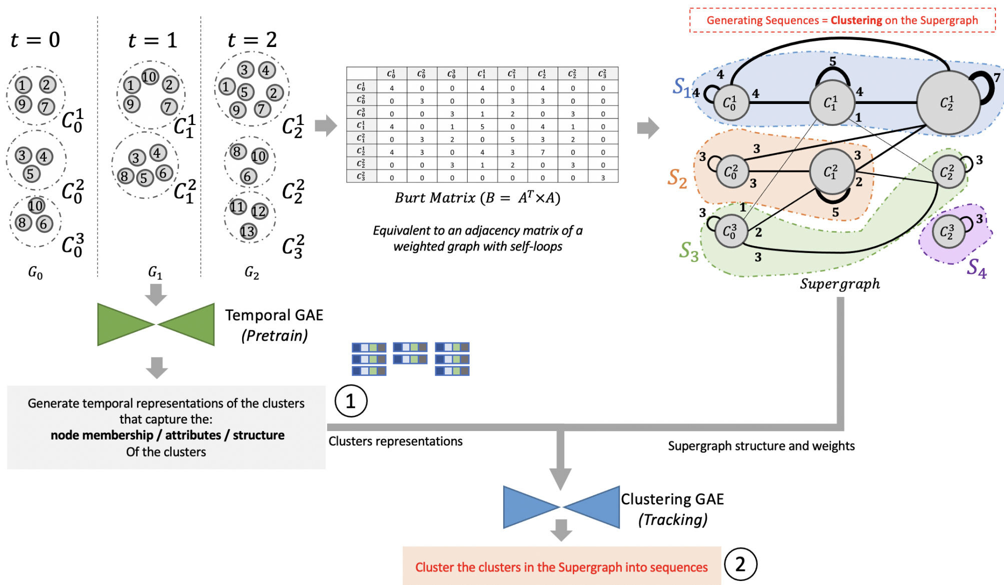

Fig. 3. TrackGAE pipeline.

- **Phase 1: Temporal GAE - Pretraining**: At each timestep, a Temporal Graph Autoencoder (Temporal GAE) generates cluster embeddings that capture the topology and attributes of clusters. These embeddings are aggregated to form a comprehensive representation of each cluster.
- **Phase 2: Clustering GAE - Tracking**: A Supergraph is constructed (using the Burt Matrix), where nodes represent clusters and edges reflect node-sharing intensity between clusters across timesteps. The Clustering GAE, enhanced with a Deep-Pruning mechanism, refines the Supergraph to extract coherent sequences of evolving communities.

Details of each phase are described in the follows.

2.1 Phase 1: Temporal GAE - Pretraining

In this phase, we generate node embeddings at each timestep and aggregate them to form cluster embeddings. To achieve this, we employ a Graph Autoencoder (GAE) at each timestep, denoted as $GAE_t\text{-}SUM(G_t)$, which learns structured representations of the graph. The encoder consists of two graph convolutional layers, denoted as l_1 and l_2, which process the adjacency matrix A_t and node features X_t to produce hidden representations H_t. These representations are then concatenated to form a more expressive encoding of each node.

To incorporate temporal dependencies, we introduce a Gated Recurrent Unit (GRU) that propagates node embeddings from one timestep to the next. This mechanism ensures that only the most relevant aspects of past representations are retained, allowing the model to capture temporal dynamics in evolving

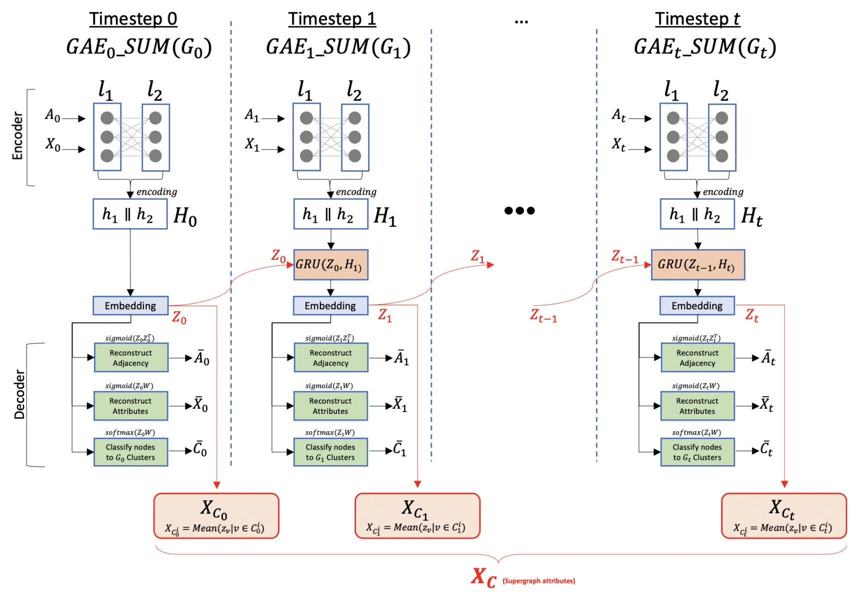

Fig. 4. Phase 1: Temporal GAE - Pretraining.

graphs. Specifically, at each timestep t, the hidden state from the previous timestep, Z_{t-1}, is integrated with the current hidden representation H_t through the GRU, producing temporally-aware node embeddings Z_t.

To further enhance the quality of representations, we introduce three auxiliary learning objectives in the decoder:

- **Adjacency Reconstruction**: We apply a sigmoid function to $Z_t Z_t^T$ to reconstruct the adjacency matrix $\bar{A}_t$, ensuring that structural properties are captured.
- **Attribute Reconstruction**: We reconstruct the original node features $\bar{X}_t$ using a transformation of the embeddings, enforcing semantic consistency.
- **Cluster Classification**: A softmax function is applied to classify nodes into their respective clusters, providing self-supervision to encourage cluster-aware representations.

Once node embeddings Z_t are learned, cluster embeddings X_{C_t} are obtained by computing the mean of the node embeddings belonging to each cluster C_t. This aggregation ensures that the cluster embeddings effectively summarize the properties of their respective nodes. Fig. 4 illustrates Phase 1. The final output of this phase is a set of cluster embeddings across all timesteps, denoted as X_C, which will be used in the second phase for constructing the Supergraph and generating community evolution sequences.

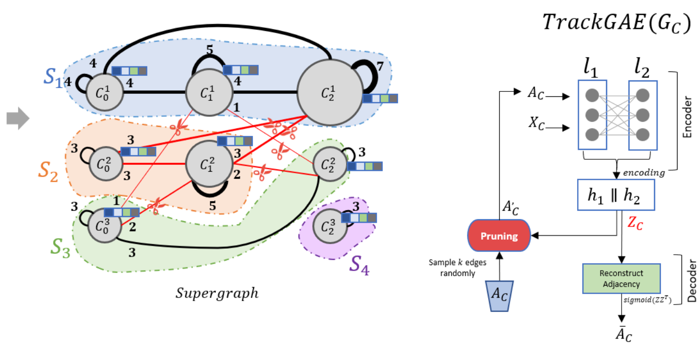

Fig. 5. Phase 2: Clustering GAE with Deep-Pruning.

2.2 Phase 2: Clustering GAE - Tracking

In this phase, our goal is to track the evolution of clusters over time by leveraging the cluster embeddings generated in Phase 1. To achieve this, we construct a Supergraph, where each node represents a cluster from a specific timestep, and edges capture the degree of node overlap between clusters. The adjacency matrix of this Supergraph is derived from the Burt Matrix, which encodes shared node memberships between clusters across different timesteps. The cluster embeddings obtained in Phase 1 serve as node attributes in this Supergraph, ensuring that both topological and semantic properties of clusters are preserved.

To identify evolving sequences of clusters, we treat this problem as a clustering task on the Supergraph. Specifically, we employ a Clustering Graph Autoencoder, which learns low-dimensional representations of the Supergraph nodes and refines the connectivity structure between them. The clustering process is reinforced by a novel mechanism called Deep-Pruning, which iteratively adjusts edge weights based on the similarity between cluster embeddings.

As illustrated in Fig. 5, during training, the Deep-Pruning mechanism updates the Supergraph structure by progressively reducing edge weights between clusters with diverging representations, while reinforcing connections between clusters that maintain high structural and attribute similarity. This iterative refinement process results in the formation of disconnected subgraphs, where each subgraph represents a coherent sequence of clusters that evolve over time. These extracted subgraphs correspond to the final community evolution sequences. The final output of this phase consists of high-quality sequences of dynamic clusters, where each sequence preserves the identity of an evolving community by considering node membership, structure, and attributes.

3 Primary Results

3.1 Dataset Construction

We construct our dataset from the Yelp Database[1], focusing on the evolution of restaurant communities over time. The Yelp dataset was chosen due to its rich temporal dynamics and availability of both structural and attribute data. Our dataset comprises 5,420 restaurants from North America, processed into temporal snapshots using a 12-month sliding window with a 1-month stride, ensuring continuous tracking of evolving restaurant interactions. In our graph representation, nodes represent restaurants, and edges are formed between two restaurants if they have been reviewed by the same user within the same 12-month window. Each restaurant is characterized by 20 dynamic attributes, recalculated at each timestep, including customer ratings, review frequency, and category distributions. To extract evolving communities, we apply Markov Clustering (MCL) independently at each timestep. MCL was chosen for its ability to detect natural clusters in graphs without requiring the number of clusters to be specified in advance. Small clusters (size ≤ 4) are discarded to ensure robust and meaningful community structures. This dataset construction effectively captures both structural and attribute dynamics, making it a suitable benchmark for evaluating algorithms.

3.2 Comparing Algorithms and Evaluation Metrics

The proposed method is evaluated against three baseline methods: Mutual Transition [3], Greene et al. [5], and Graph Edit Distance (GED) [6]. We assess the quality of the tracked community sequences based on three properties: **(1) Node Membership Consistency**: Measures the stability of core members within evolving clusters. **(2) Attribute Homogeneity**: Evaluates whether cluster attributes remain coherent over time. **(3) Structural Coherence**: Ensures that the topological structure of clusters remains stable.

3.3 Node Membership Consistency

To evaluate the consistency of node membership across timesteps, we compute two key metrics: the Pearson Correlation Coefficient between cluster vectors derived from the normalized Burt matrix and the Average Proportion of Nodes Persisting (APNP). As shown in Table 1, TrackGAE achieves a Pearson correlation of 0.515, which is higher than Mutual Transition (0.486) and Greene et al. (0.486), while slightly lower than GED (0.498). Similarly, in terms of APNP, TrackGAE obtains a score of 0.458, which surpasses all competing algorithms.

[1] https://business.yelp.com/data/resources/open-dataset/

Table 1. Node Membership Consistency Evaluation.

Method	**Pearson Correlation**	**APNP**
Mutual Transition	0.486	0.396
Greene et al.	0.486	0.396
GED	0.498	0.397
TrackGAE	**0.515**	**0.458**

3.4 Attribute Homogeneity

To assess attribute consistency across evolving clusters, we employ four clustering metrics: Davies-Bouldin Index, Silhouette Score, Calinski-Harabasz Score, and Reverse Canberra Distance (RCD). The results are presented in Table 2.

Table 2. Attribute Homogeneity Evaluation.

Method	**DB Index ↓**	**Silhouette ↑**	**CH Score ↑**	**RCD ↑**
Mutual Transition	0.989	-0.031	3.745	0.646
Greene et al.	0.989	-0.031	2.837	0.895
GED	0.662	0.024	2.314	**1.003**
TrackGAE	**0.610**	**0.029**	**4.665**	0.996

TrackGAE achieves a Reverse Canberra Distance of 0.996, surpassing Mutual Transition and Green, but remaining slightly below GED (1.003). The Davies-Bouldin Index for TrackGAE is measured at 0.610, significantly lower than Mutual Transition (0.989) and Green (0.989), indicating that the generated sequences are more compact and well-separated.

3.5 Structural Coherence

Structural integrity is another critical factor in dynamic cluster tracking. We evaluate structural consistency using *NetSimile*-based structural similarity. The results are summarized in Table 3. TrackGAE achieves a Reverse Canberra Distance of 0.471, outperforming all baseline models. It also achieves the lowest Davies-Bouldin Index (1.892), indicating well-formed and compact clusters. However, in terms of the Silhouette Score, TrackGAE achieves -0.068, which, while an improvement over Mutual Transition (-0.196) and Greene et al. (-0.196), remains slightly below GED (-0.044), suggesting that there is room for improvement.

Overall, the results demonstrate that TrackGAE provides steady improvements in node membership consistency and structural coherence. These improvements highlight the effectiveness of TrackGAE in preserving community identity over time by integrating node membership, structural integrity, and attribute consistency. However, these findings are preliminary and based on a single

Table 3. Structural Coherence Evaluation.

Method	**DB Index** ↓	**Silhouette** ↑	**CH Score** ↑	**RCD** ↑
Mutual Transition	3.401	-0.196	3.827	0.386
Greene et al.	3.401	-0.196	3.827	0.386
GED	1.996	**-0.044**	4.314	0.455
TrackGAE	**1.892**	-0.068	**4.805**	**0.471**

dataset (Yelp), which limits the generalizability of the approach. A more comprehensive empirical evaluation, including comparisons with additional state-of-the-art methods and testing on larger, more diverse datasets, is necessary to fully assess the applicability of TrackGAE to real-world dynamic networks.

4 Conclusion

We presented TrackGAE, a novel framework for tracking evolving communities in dynamic graphs by integrating graph autoencoders with a Deep-Pruning mechanism. TrackGAE preserves node membership consistency, structural coherence, and attribute homogeneity over time. While effective, challenges remain, including dataset sparsity and the need to tune pruning parameters. Future work will focus on improving generalization through self-supervised learning, designing adaptive pruning strategies based on cluster similarity, and refining evaluation metrics for long-term evolution. We also plan to extend the empirical validation to more diverse datasets and state-of-the-art methods to assess scalability and broader applicability.

References

1. M. Mazza, G. Cola, and M. Tesconi, "Modularity-Based Approach for Tracking Communities in Dynamic Social Networks," *Knowledge-Based Systems*, vol. 281, p. 111067, 2023.
2. X. Jia, X. Li, N. Du, Y. Zhang, V. Gopalakrishnan, G. Xun, and A. Zhang, "Tracking Community Consistency in Dynamic Networks: An Influence-Based Approach," *IEEE Trans. Knowl. Data Eng.*, vol. 33, pp. 782–795, Feb. 2021.
3. E. G. Tajeuna, M. Bouguessa, and S. Wang, "Modeling and Predicting Community Structure Changes in Time-Evolving Social Networks," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 6, pp. 1166–1180, 2019.
4. R. Cazabet and G. Rossetti, "Challenges in Community Discovery on Temporal Networks," in *Temporal Network Theory*. Springer, pp. 1–25, 2019.
5. D. Greene, D. Doyle, and P. Cunningham, "Tracking the Evolution of Communities in Dynamic Social Networks," in *Proceedings of the 2010 IEEE International Conference on Advances in Social Networks Analysis and Mining*, pp. 176–183, 2010.
6. P. Brodka, S. Saganowski, and P. Kazienko, "GED: The Method for Group Evolution Discovery in Social Networks," *Social Network Analysis and Mining*, vol. 3, no. 1, pp. 1–14, 2013.
7. M. Takaffoli, J. Fagnan, F. Sangi, and O. R. Zaïane, "Tracking changes in dynamic information networks," in *Proceedings of the International Conference on Computational Aspects of Social Networks (CASoN)*, pp. 94–101, 2011.