המכללה
האקדמית להנדסה
בראודה
בכרמיאל

Software Engineering Department
Braude College

Capstone Project Phase A – 61998

Citation networks evolution using Dynamic Network Embeddings

25-1-R-16

Students:

Avishay Bar 316165984

Shay Bistrizky 206622086

Supervisors:

Prof. Zeev (Vladimir) Volkovich

Dr. Renata Avros

GIT Repository Link

Contents

1 Abstract

Citation networks are modeled as a series of graphs. In these graphs, nodes represent articles, and edges indicate citations from one article to another. Each graph corresponds to a snapshot at a specific timestamp [5]. These networks evolve with the addition of new research and references. Identifying public opinion leaders, whether articles or subjects, is essential for understanding research influence and emerging trends. Advanced programming techniques are necessary to analyze these dynamic networks effectively.

Dynamic network embeddings offer a powerful approach for analyzing evolving networks by transforming graphs into vectors within a shared vector space [4]. By employing methods such as dynamic Bernoulli embeddings [6] and sequential random walks [2], this research aims to model temporal changes effectively. Through tasks such as link prediction and evolving node detection, the approach demonstrates its capability to identify influential works and emerging research areas in citation networks.

In conclusion, dynamic network embedding methods provide a robust framework for uncovering key insights in citation networks, driving a deeper understanding of research trends and influence over time.

2 Introduction

Highly cited articles are instrumental in shaping the direction of research discourse. Spanning a vast array of diverse subjects, are often published in response to the popularity of certain subjects. By detecting such works, citation networks provide valuable insights into the evolution of research priorities and the propagation of knowledge.

As shown in Figure 1, highly cited articles serve as central nodes within citation networks, connecting diverse knowledge domains and facilitating interdisciplinary collaboration [3]. These central articles play a crucial role in driving innovation, highlighting significant research areas, and influencing funding priorities toward emerging fields. Their impact extends beyond individual research areas, shaping the evolution of disciplines and driving transformative shifts across the broader academic landscape. Over time, their influence grows as they continue to guide future research directions and inspire new discoveries.

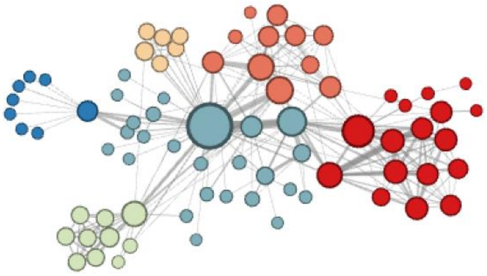

Figure 1: Illustration of highly cited articles as central nodes in a citation graph, highlighting their role in connecting diverse domains of knowledge and fostering interdisciplinary collaboration. *[3]*

Static analysis methods are techniques that analyze networks by assuming a fixed structure, where nodes and edges remain unchanged over time [1]. Traditional static analysis methods, such as DeepWalk [2] and Node2vec [3] models, are limited in capturing the dynamic nature of evolving networks. By treating the network as a static graph, these methods provide only a snapshot at a specific timestamp, ignoring crucial temporal variations [5]. As a result, these approaches fail to account for critical changes such as the emergence of new influential nodes or the shifting importance of edges over time. This static perspective makes it difficult to analyze trends, predict future interactions, or understand temporal patterns in the network's structure. For instance, while DeepWalk [2] model uses random walks [2] to learn embeddings, it assumes a fixed structure, ignoring how nodes and edges evolve. Similarly, Node2vec [3] model introduces flexibility in embedding generation but still lacks the ability to incorporate temporal dynamics, leading to incomplete representations of the network's evolution.

Dynamic networks address the limitations of static analysis by modeling nodes and edges that change over time. Unlike static networks, dynamic networks capture how structures evolve, providing critical insights into temporal interactions [5]. By tracking these changes, dynamic networks enable the analysis of trends, the identification of emerging patterns, and the understanding of how relationships develop over time.

Dynamic network embedding enhances this framework by generating low-dimensional representations of nodes that capture both structural relationships and temporal changes [4]. Using specialized techniques, these embeddings ensure proximity preservation by keeping similar nodes close in the embedding space, while also ensuring temporal continuity by minimizing unnecessary drift for stable nodes over time. This approach enables researchers to identify influential articles in citation networks, shaping funding decisions, academic hiring, and research prioritization.

Analysing citation dynamic networks over time provides significant contributions to understanding the evolution of scientific knowledge and research priorities. This allows for the detection of the identification of influential works at different stages of their lifecycle. For instance, articles that may initially receive limited attention can later emerge as central nodes, influencing research trajectories and fostering innovation. By capturing these temporal patterns, dynamic network analysis enables a more nuanced understanding of how scientific ideas propagate, evolve, and gain prominence over time.

The benefits of dynamic citation network analysis extend beyond academic insights. It provides actionable intelligence for decision-makers such as funding agencies, institutions, and policymakers. By identifying rising areas of influence and predicting future research directions, these analyses can guide the allocation of resources toward high-potential fields.

Additionally, tracking changes in the importance of nodes and edges helps in evaluating the long-term impact of research contributions, improving strategies for academic hiring, grant distribution, and collaborative initiatives. Ultimately, dynamic citation network analysis fosters a deeper understanding of scientific progress, facilitating data-driven decisions that enhance innovation and accelerate the advancement of knowledge.

3 Background

3.1 Dynamic Network Background

Dynamic networks are structured systems where the connections (edges) between entities (nodes) evolve over time, reflecting changes in relationships or interactions. Unlike static networks with fixed topologies, dynamic networks are represented by a series of time-dependent snapshots or continuous-time models, capturing the temporal variation in their structure [7]. These changes may include the addition or removal of nodes, the strengthening or weakening of connections, or shifts in the network's overall topology. However, in citation networks, edges are only added over time and not removed, as citations represent permanent references once established.

Definition 1 (Dynamic Networks): A dynamic network is a series of graphs $\Gamma = \{G_1,...,G_T\}$ and $G_t = (V_t,E_t)$, where T is the number of graphs, V_t is a node set and E_t includes all temporal edges within the timespan $[S_t,S_t+1]$. Each $e_i = (u,v,s_i) \in E_t$ is a temporal edge between the node $u \in V_t$ and the node $v \in V_t$ at the timestamp $s_i \in [S_t,S_t+1]$. *[5]*

Two critical factors in dynamic networks are the **proximity between nodes** and the **temporal continuity of nodes over time** [5]:

- **Proximity between nodes** refers to the measure of how closely connected two nodes are within the network at a certain timestamp graph. The proximity can be influenced by direct connections (e.g. a temporal edge between two nodes) or indirect relationships (e.g., shared temporal neighbours or paths).
- **Temporal continuity of stable nodes** involves maintaining consistent representations of nodes that remain relatively unchanged across time. This is crucial for capturing the evolution of nodes and their roles within the network without introducing unnecessary distortions. For example, consider an article in a specific scientific field that continues to be cited over decades. The node representing this article remains stable in terms of its role in the network, consistently attracting citations as new research builds upon its findings.

3.2 Graph2Vec and its inspiration for Dynamic Network Embeddings

Graph2Vec [1] is a neural network-based method designed to create fixed-length vector representations of entire graphs. These representations, called graph embeddings, capture the global structural properties of graphs. Graph2Vec is particularly effective for tasks like graph classification and clustering, as it ensures that structurally similar graphs are represented by vectors that are close to each other in the embedding space.

The method works by treating an entire graph as a single entity and breaking it down into smaller components called **rooted subgraphs** [1]. These rooted subgraphs represent local neighbourhoods within the graph, capturing its structural details. To extract these subgraphs, Graph2Vec uses a process called Weisfeiler-Lehman (WL) [8] relabelling, which systematically encodes the relationships and proximities of nodes and edges.

Graph2Vec uses a neural embedding model inspired by the Skip-Gram model [9] for learning graph representations. The Skip-Gram model, originally developed for word embeddings (e.g., Word2Vec [9]), is adapted in Graph2Vec [1] to treat rooted subgraphs as "words" and the entire graph as a "document." The goal of this adaptation is to predict which subgraphs are likely to co-occur within the same graph, based on their structural relationships. This prediction process involves optimizing the embeddings such that similar graphs have similar vector representations. To make the learning efficient, Graph2Vec employs a technique called negative sampling (explained further in section 3.6), where it focuses on distinguishing between the actual subgraphs in a graph and random subgraphs that do not belong to it.

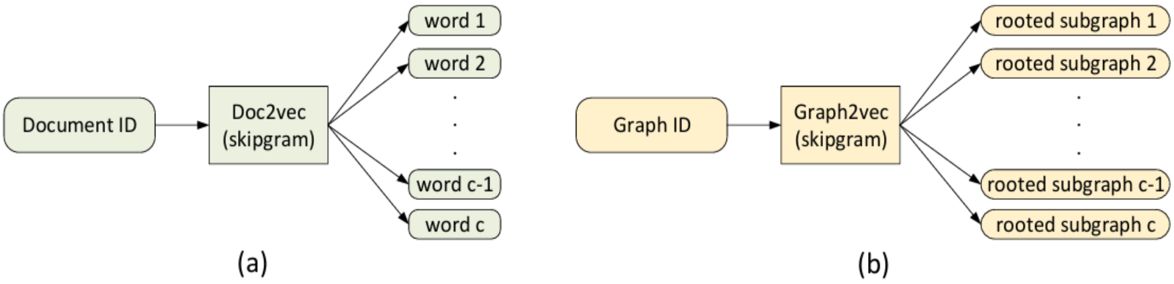

Figure 2: (a) Doc2Vec's SkipGram model - Given a document d, it samples c words from d and considers them as co-occurring in the same context. (b) Graph2Vec – Given graph G, it samples c rooted subgraphs around different nodes that occur in G. *[1]*

One of the key strengths of Graph2Vec is its flexibility. It does not require labelled data, making it suitable for unsupervised tasks where graphs need to be represented as vectors for further analysis.

In summary, Graph2Vec is a powerful tool for analysing graph-structured data. By transforming graphs into meaningful vector representations, it enables the use of machine learning techniques for a variety of tasks, including identifying similarities, classifying graphs, and grouping them into clusters based on their structural properties.

3.3 Citation Dynamic Network Specificities

Citation networks exhibit unique and complex properties that distinguish them from other types of dynamic networks:

- **Irreversibility of Edges:** A citation, once established, is permanent and unidirectional. This reflects the enduring nature of knowledge transfer, where older foundational works remain integral to the evolving research landscape.
- **Node Longevity:** Articles often remain relevant for years or even decades after publication. This longevity reflects their continued influence as foundational works or seminal contributions in a specific domain. Articles with sustained citation activity serve as anchors in the evolving structure, providing stability in the dynamic embedding space.
- **Citation Burstiness:** Citations are unevenly distributed over time. Articles may experience sudden surges in citations due to emerging trends, paradigm shifts, or

groundbreaking discoveries. These bursts are crucial for identifying trending publications and tracking shifts in research focus.

- **Hierarchical Structure of Influence:** Citation networks often exhibit a hierarchical flow of influence, where older, highly cited works act as "roots," and newer works branch out to build upon these foundations. Over time, articles may accumulate more citations, further reinforcing their importance in the network hierarchy.
- **Field-Specific Dynamics:** Different research fields exhibit distinct citation behaviors. For instance, fast-evolving fields such as artificial intelligence (AI) and bioinformatics experience rapid bursts of citations, while fields like mathematics or physics may have longer citation cycles where articles remain influential for extended periods.

Understanding these characteristics is crucial for designing methods tailored to citation network analysis. By accounting for the permanence of edges, the longevity of influential nodes, and the bursty nature of citation events, dynamic embedding methods can better capture the evolving structure and influence of articles within citation networks. These insights allow researchers to identify emerging trends, track the propagation of knowledge, and recognize the enduring impact of foundational works.

3.4 Random Walk

A random walk is a stochastic process that generates a path consisting of successive steps on a mathematical space, such as a graph. At each step, the next node is randomly selected from the neighbours of the current node [2].

Random walks are widely used for efficiently exploring graph structures, analysing connectivity, and identifying communities. Additionally, they enable effective sampling of large networks, especially when global computations are impractical or computationally expensive.

Basic random walk algorithm on graph: [2]

Input:

- $G = (V, E)$: A graph with nodes V and edges E.
- v_0 : stating node
- L : length of the walk
- W : Edge weights (optional)

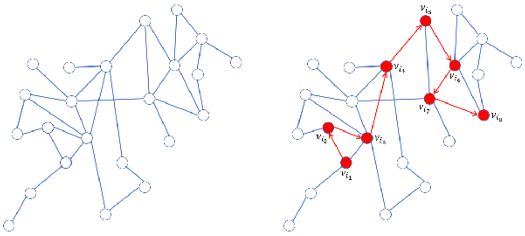

Figure 3: Construction sequence of nodes using random walks

Output:

- $[v_0, v_1, \dots, v_L]$: A sequence representing the random walk.

Steps:

1. Set $v_{current} = v_0, Path = [v_0]$
2. For $i = 1$ to L:

```
        a. Set $N = \{u|(v_{current}, u) \in E\}$
        b. If graph is unweighted:
            i. Select $v_{next}$ uniformly at random from N
        c. Else:
            i. For each $u \in N$ Compute the probably:
                $P(v_{current} \rightarrow u) = \frac{w_{v_{current},u}}{\sum_{k \in N} w_{v_{current},k}}$
            ii. Select $v_{next}$ based on the probability distribution.
        d. Append $v_{next}$ to $Path$
        e. Set $v_{current}$ as $v_{next}$
    3. Return $Path$
```

The random walk length (L) is a key parameter that directly influences the behaviour, outcome, and efficiency of graph exploration and the tasks that depend on the random walk, such as graph traversal, connectivity analysis, and node embedding generation:

- **Short walks** in a random walk process focus on exploring the local neighbourhood of a starting node and capturing immediate connectivity. These walks are particularly suitable for tasks requiring fine-grained, localized relationships, such as detecting tightly connected communities or performing micro-level analysis.
- **Long walks** explore a broader region of the graph, capturing global structures and distant connectivity patterns. This makes them useful for understanding macro-level graph properties, such as identifying how nodes are connected across different clusters or detecting bridge nodes that link communities.

Random walks are considered as efficient tools for exploring dynamic networks due to their localized and adaptable nature. They allow real-time updates as networks evolve, enabling tasks like community detection, anomaly detection, node embedding, and influence propagation in changing environments. By leveraging their flexibility and scalability, random walks provide a robust solution for understanding the complex temporal behavior of dynamic networks.

3.5 Stochastic Gradient Descent (SGD)

Stochastic Gradient Descent [10] is an optimization algorithm in machine learning and deep learning. It is an iterative method for minimizing an objective function, often the loss function of a model, which is parameterized by a set of weights or parameters.

Understanding the Mechanism of SGD:

1. **Iterative Updates:** SGD works by adjusting model parameters based on how the model performs on individual data points or small subsets of the given dataset.

2. **Random Sampling:** Instead of looking at the entire dataset, SGD uses a randomly selected data point from the dataset at each iteration. This makes it computationally efficient and suitable for large-scale data.

3. **Formula for Update:** At each iteration, the model parameters are updated as follows:

$$\theta = \theta - \eta \cdot g$$

θ: Model parameters
η: Learning rate, controlling the size of the step in the parameter update
g: Gradient of the loss function computed

4. **Stochastic Nature:** Due tot the fact that SGD uses random samples, the updates have some stochasticity. This introduces noise into the optimization process, which can help the algorithm escape minimum point values.

By leveraging iterative updates and random sampling, SGD offers a scalable solution for training models on large datasets. While its stochastic nature introduces noise, this can be advantageous in navigating complex optimization landscapes, helping escape local minimum points and improving overall performance.

3.6 Dynamic Bernoulli Embeddings

Dynamic Bernoulli embeddings [6] are a novel approach in dynamic network analysis that aim to preserve the proximity and temporal continuity of nodes in a dynamic network. Unlike methods that treat networks as static structures, this technique incorporates temporal data by embedding nodes into a low-dimensional vector space over discrete time steps. Dynamic Bernoulli embeddings utilize random walks to generate sequences of nodes. This approach ensures that stable nodes maintain continuity across time while capturing the evolution patterns of dynamic nodes.

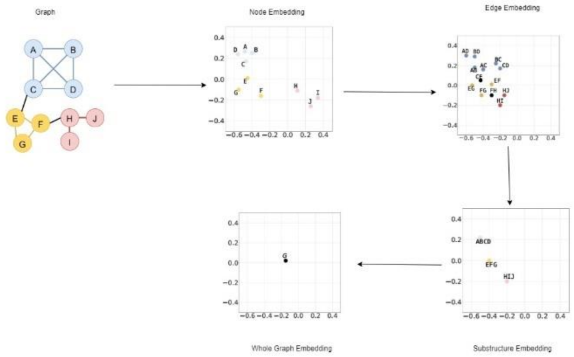

Figure 4: A single iteration of dynamic Bernoulli embedding within the same vector space.

Key Components of the Calculation Process: [5]

1. Embedding Vectors:

- **Node embedding** $\left(y_g^{(t)}\right)$: A low-dimensional vector that represents node g at time t. It captures the node's relationships within the network at that timestamp.
- **Context vector** (α_g): A shared vector for each node, representing its role or interactions across all timestamps. It acts as a reference point to determine how nodes relate to their context.
- **Context Size** (cs)**:** The number of neighboring nodes considered around a given node in a random walk sequence for learning embeddings. For example, if cs=4, the two nodes before and two nodes after the given initial node in the sequence are part of its context.

2. <u>Node Indicator Vector $\left(x_i^{(t)}\right)$:</u>

   - This is a binary vector where $x_{ig}^{(t)}$=1 if node g is part of the context for the i-th position in a random walk at time t and otherwise $x_{ig}^{(t)}$ =0.
   - It ensures that the model correctly identifies which nodes are part of a given context.

3. <u>Context Relationship Score (η_{ig}):</u>

   - This score measures how well a node g fits its context.
   - Calculated as:

$$\eta_{ig} = y_g^{(t)T}\left(\sum_{k\in\left[i-\frac{c_s}{2},i+\frac{c_s}{2}\right]\wedge k\neq i}\ \sum_{g\in V_t} \alpha_g x_{kg}^{(t)}\right)$$

$y_g^{(t)}$: Embedding of node g at time t.
α_g: Context vector of node g.
c_s: Size of the context window.

4. <u>Likelihood Functions:</u>

   To ensure embeddings preserve proximity and temporal continuity, the model defines two likelihood functions:

- **Positive Likelihood** ($\mathcal{L}_{pos}$):

$$\mathcal{L}_{pos} = \sum_{i=1}^{L}\sum_{g\in V_i} x_{ig}^{(t)} log\sigma(\eta_{ig})$$

This measures how well the model predicts the true relationships between nodes in the network.

- **Negative Likelihood** ($\mathcal{L}L_{neg}$):

$$\mathcal{L}_{neg} = \sum_{i=1}^{L}\sum_{g\sim\phi} \log\left(1-\sigma(\eta_{ig})\right)$$

Here, ϕ represents a sampling distribution for "negative" nodes (nodes not part of the true context). Negative sampling reduces computational complexity.

Negative Sampling: Instead of calculating $\mathcal{L}_{neg}$ for all possible negative nodes (which would be computationally expensive), a small subset is sampled at random. This speeds up training while retaining accuracy. The number of "negative samples" drawn during the embedding training process is defined as negative sample size (ns).

5. <u>Regularization:</u>

   To ensure temporal consistency and prevent overfitting, two regularization terms are added:

   - For the context vector:

$$\mathcal{L}_a = -\frac{\lambda_1}{2}\sum_{g\in V_i} |\alpha_g|^2$$

This ensures that the context vectors remain stable and bounded.

- For temporal continuity:

$$\mathcal{L}_y = -\frac{\lambda_1}{2}\sum_{g\in V_t} \left|y_g^{(1)}\right|^2 - \frac{\lambda}{2}\sum_{\substack{g\in V_t \\ t\in[1,T]}} \left|y_g^{(t)} - y_g^{(t-1)}\right|^2$$

This penalizes large changes in embeddings between consecutive timestamps.

Basic DBE (Dynamic Bernoulli Embedding) algorithm on graph: [5]

Input:

- $y_g^{(t)}$: A low-dimensional vector that represents node g at time t.
- α_g: Current context vector.
- $W_g^{(t)}$: A sequence of nodes representing the random walk.
- V_t : Nodes group of timestamp t.
- cs : Context size.
- ns : Negative sample size.

Output:

- Updated values of both $y_g^{(t)}$ and a_g

Steps:

```
1. For each $w_k^{(t)}$ in $W_g^{(t)}$:
    a. For each $v_i$ in $w_k^{(t)}$:
        i. $V_{v_i} = NegativeSampling(V_t, v_i, ns)$
        ii. Minimize loss $\mathcal{L}_y(y, \alpha)$ by SGD($y_g^{(t)}, a_g, V_g, cs$)
        iii. Update $y_g^{(t)}$ and $a_g$
    b. End for
2. End for
```

Dynamic Bernoulli Embeddings [6] provide a robust framework for analyzing dynamic networks by preserving both the temporal continuity and proximity of nodes. The model achieves this by leveraging random walks, context-aware embeddings, and regularization techniques. By focusing on computational efficiency through negative sampling and carefully balancing the likelihood and regularization terms, this approach can effectively model the evolution of networks over time.

3.7 Dynamic Network Construction

Definition 2 (Dynamic Network Embeddings): Given a dynamic network Γ = {G1,...,GT }, dynamic network embeddings aim to project a node ∂ ∈ Vt into a low-dimensional vector space by a mapping function f : ∂ → y(t) | ∂ ∈ R^D, D ≪ max|V_t|, t ∈ [1,T]. *[5]*

Dynamic networks are constructed from timestamped events, such as citations. Each snapshot $G_t = (V_t, E_t)$ is derived using either: [5]

1. **Fixed Time Intervals (ω):**
   - Edges E_t are defined based on events occurring within the time window $[S_t, S_t + \omega)$.
   - This approach ensures temporal consistency but may result in sparse graphs if events are unevenly distributed.

2. **Fixed Number of Events (ε):**
   - Each snapshot contains a predefined number of events, ensuring uniform density across graphs.
   - However, this approach may break temporal consistency, which can be addressed using overlapping windows.

To ensure smooth transitions and maintain continuity across the dynamic network, an overlap is introduced between adjacent graphs. This overlap creates a connection between consecutive graphs, preserving the flow of information and minimizing disruptions in the network's temporal dynamics.

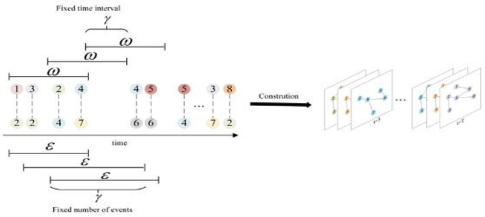

Figure 5: Dynamic network construction by the fixed time *[5]*

There are four main matrices that represent the Dynamic Network Embeddings:

1. **Embedding Matrix (M_t):** each row corresponds to a node's low-dimensional vector representation at time t. These embeddings are specifically optimized to maintain both the proximity of nodes within the graph and the temporal continuity across different time steps.
2. **Adjacency Matrix (A_t):** encodes the connections between nodes in the graph at time t. In this matrix, $A_t[i,j] = 1$ indicates the presence of an edge between nodes j. This matrix is instrumental in calculating node similarities and serves as a foundation for generating random walks within the graph.

3. **Context Matrix (α):** stores context vectors for nodes, capturing neighborhood information derived from random walks. These context vectors are dynamically updated during training to maximize the likelihood of observing true neighboring nodes while minimizing the likelihood of random connections through negative sampling.
4. **Temporal Regularization Matrix:** designed to align the embedding matrices $M_t$and M_{t+1} by penalizing significant differences in the embeddings of stable nodes across consecutive timesteps. This mechanism ensures smooth temporal transitions in node representations, maintaining consistency and stability in the embeddings over time.

The embedding process can be summarized as follows:

- **Input:** A dynamic network $\Gamma = \{G_1, \dots, G_T\}$, random walk parameters (r, L), embedding size (D), context size (cs), and negative sampling size (ns).
- **Process:** For each snapshot G_t , random walks generate sequences of nodes, which are used to update embedding and context matrices (M_t and α).
- **Output:** Embedding matrices $M_1, \dots, M_T$ that represent the temporal and structural properties of the network.

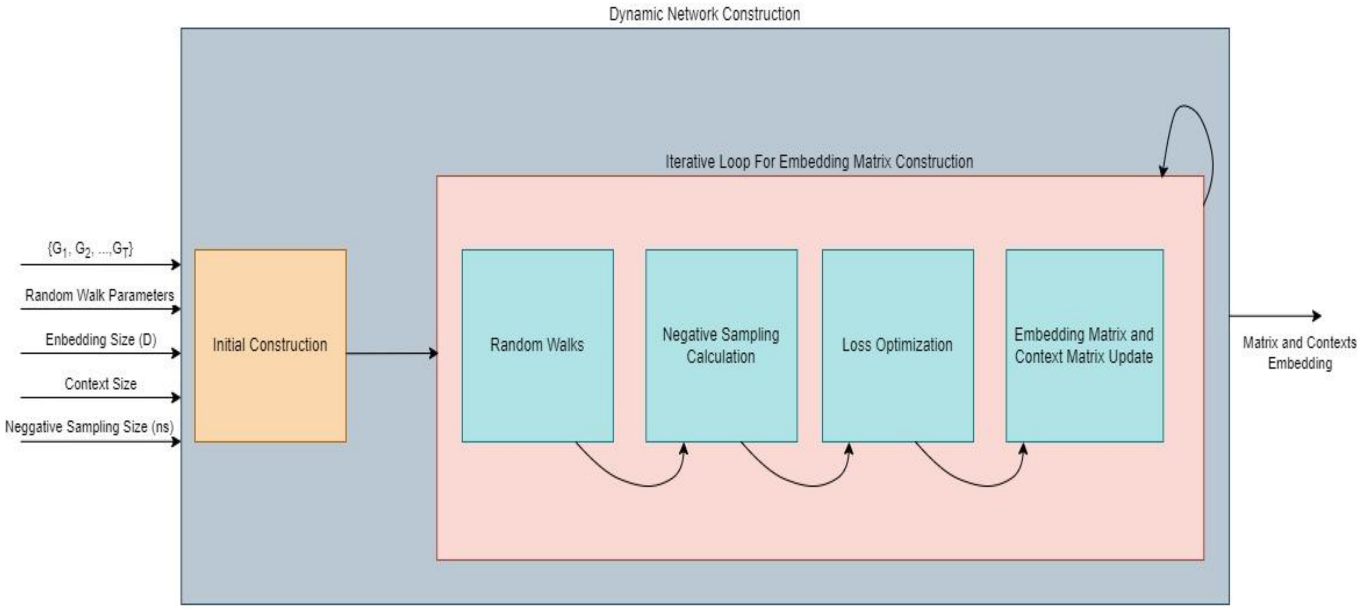
Dynamic Network Construction
Iterative Loop For Embedding Matrix Construction
{G1, G2, ...,GT}
Random Walk Parameters
Enbedding Size (D)
Context Size
Neggative Sampling Size (ns)
Initial Construction
Random Walks
Negative Sampling Calculation
Loss Optimization
Embedding Matrix and Context Matrix Update
Matrix and Contexts Embedding

Figure 6: Dynamic Network Embedding model structure

4 Expected Achievements

The primary goal of this research is to explore and characterize the dynamic behavior of citation networks over time, leveraging advanced embedding techniques to uncover meaningful patterns and trends in academic influence and impact.

Since citations form a complex and ever-changing network where the importance and relationships of articles fluctuate over time, this research project aims to explore the potential of dynamic network embeddings to uncover patterns and trends in citation dynamics. By transforming a series of static citation graphs into a temporal network representation, the project seeks to investigate the behaviour of articles over time, identifying archetypes such as "rising stars", "falling stars", and "steady classics". The insights gained advance our understanding of how scientific impact evolves.

Through the characterization of citation graphs, this research provides a deeper understanding of the underlying dynamics that drive the evolution of scientific influence over time. By identifying patterns, the study sheds light on how articles impact their fields over time. Such characterizations are essential for uncovering trends in knowledge dissemination, mapping the lifecycle of academic influence, and identifying key contributions that shape the trajectory of scientific progress.

The project's primary objectives are centered around the following key components:

1. Develop a solid framework for representing citation networks as dynamic systems using Dynamic Bernoulli Embedding Model [5].
2. Derive meaningful characterizations of articles from the embedding space, focusing on identifying and distinguishing different characterizations.
3. Validate the proposed methodology against real-world citation data, ensuring accuracy in capturing dynamic behaviors and uncovering significant trends.
4. Optimize the embedding model to enhance computational efficiency and scalability, making the framework suitable for large and complex datasets.

This project aims to contribute both a methodological framework and actionable insights into the dynamics of academic citations.

5 Approach

5.1 Initial Construction Method

The initial construction of dynamic networks plays a pivotal role in determining the quality of the embeddings generated and the insights derived from them. For our approach, we explore two primary methods of construction: fixed time intervals and fixed event count [5]. Each method serves different purposes, and the selection depend on the nature of the data and the downstream tasks.

Using fixed time intervals, we segment the network into snapshots based on predefined temporal windows (e.g., days, weeks, or months). Each snapshot aggregates events occurring within its respective time window. This method ensures temporal consistency and is well-suited for datasets where events are evenly distributed over time. In citation networks, snapshots could represent sets of articles published within a specific period, showing how research and ideas develop and connect over time. Time intervals should align with the way articles are published and cited over time, allowing us to capture meaningful changes in the network without making graphs too sparse or too dense. It is essential to choose intervals that are not so short that they create overly sparse graphs, nor so long that they hide important changes in citation trends.

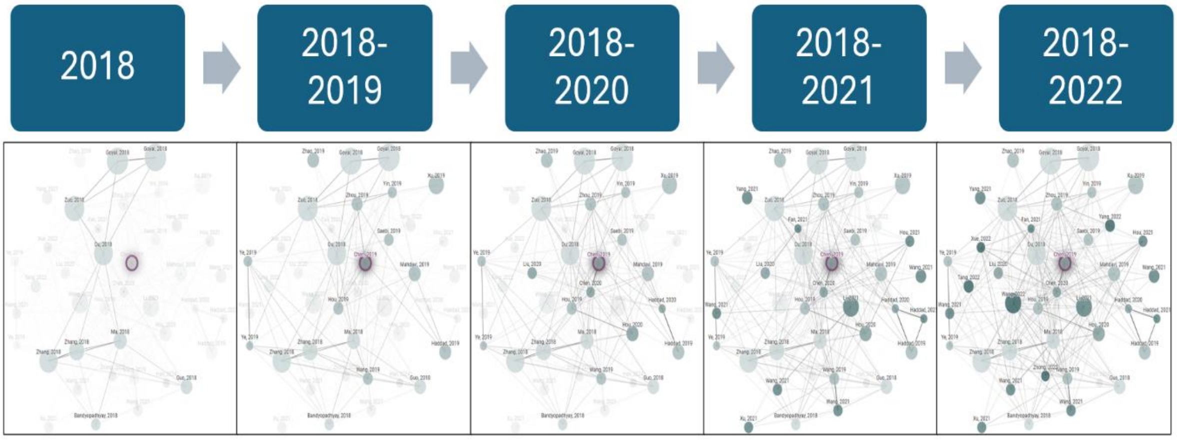

Figure 7: Fixed Time Intervals of one year example by ConnectedPapers

Alternatively, the fixed event count method segments the network into snapshots containing a predetermined number of events, ensuring uniform density across snapshots. This is particularly useful for datasets with uneven temporal distributions, such as email communication networks where activity may spike at certain times. Event counts should align with the average density of the dataset, maintaining balanced snapshots that are comparable across time. The use of overlapping windows is crucial for preserving temporal continuity and ensuring smooth transitions between snapshots. This approach helps maintain the consistency of dynamic relationships across time intervals, reducing disruptions and enabling a more accurate representation of the network's evolution.

Our approach assumes that the event-based method, which segments the network into snapshots containing a fixed number of events, is better suited for dynamic citation networks. This assumption stems from the observation that citation events are often unevenly distributed over time, and using fixed event counts ensures uniform density across snapshots. By maintaining this balance, we can generate embeddings that remain stable and comparable, allowing for more accurate insights into temporal patterns and trends.

5.2 Decision-Making on Embedding Size

The embedding size is a crucial parameter that defines the dimensionality of node representations in citation networks. Based on insights from research in network representation learning and node classification tasks, embedding sizes between 64 and 128 dimensions are recommended for citation networks. This range ensures that the embeddings can preserve essential structural and contextual information while maintaining computational efficiency. Citation networks typically have simpler structures compared to other large-scale networks like social networks, with nodes (papers) connected through citations that often reflect clear relationships based on topics, fields, or temporal trends. This makes embeddings in the 64–128-dimension range effective for representing such straightforward interactions.

Studies on algorithms like DeepWalk [2], node2vec [3], applied on datasets such as Cora, PubMed, and Citeseer, indicate that embedding sizes within this range are sufficient for node classification tasks. Lower-dimensional embeddings maintain performance for classification tasks while reducing computational overhead. Moreover, in citation networks, embeddings need to capture meaningful structural similarities, such as shared references, while avoiding overfitting to sparsely connected nodes. The 64–128-dimension range achieves a balance between these needs, ensuring efficient and meaningful representations [11].

The survey on network representation learning highlights the importance of scalable embedding techniques for applications like node classification and clustering. Citation networks, as discussed in the survey, benefit from embeddings that preserve both the network's local and global structures while maintaining low computational costs. Our approach aligns with these findings by focusing on embedding sizes that facilitate efficient analysis without compromising the quality of insights into citation dynamics.

For citation networks, embedding sizes is systematically tuned during the analysis phase. This process involves evaluating how different sizes impact tasks like detecting evolving nodes and maintaining temporal continuity. The goal is to ensure embeddings effectively represent the dynamic structure and influence of nodes within citation networks while minimizing computational burdens. Adopting dimensions within the 64–128 range ensures a balance between representation quality and efficiency [11], enabling accurate and scalable analysis of dynamic citation networks.

5.3 Decision-Making on Number of Walks and Context Size

The selection of the number of walks per node and context size is a critical step in optimizing the dynamic network embedding process [5]. Both parameters influence how well the model captures temporal changes and structural relationships in the network.

The context size governs the range of neighboring nodes incorporated during the embedding process. A smaller context size emphasizes localized relationships, aiding in the detection of tightly knit communities. Conversely, a larger context size captures extended dependencies, thereby accommodating a broader network perspective. While specific empirical results are yet to be determined, we plan to experiment with different context sizes to identify an optimal value that effectively captures citation relationships. Our approach explores how varying context sizes influence the ability to detect changes and interactions within dense or evolving research networks, particularly focusing on the balance between local and extended dependencies.

The number of walks per node affects how thoroughly the network is explored. A higher number of walks provides more comprehensive coverage, enabling better detection of evolving nodes and their dynamic relationships. However, increasing the number of walks must be balanced with computational efficiency to ensure practicality in large networks. For citation networks, testing different values for the number of walks help identify a configuration that optimally captures key citation patterns while maintaining scalability. By studying the interplay between the number of walks and other parameters, such as context size, we aim to enhance the detection of evolving nodes and improve the representation of dynamic citation relationships.

By applying these principles to citation networks, we aim to optimize the interplay between these parameters to enhance the detection of evolving nodes and improve the representation of dynamic citation relationships.

5.4 Negative Sampling Method

Negative sampling is a critical technique for efficiently training dynamic network embeddings. Instead of calculating the relationships of all possible pairs of nodes, it focuses on a subset of non-context nodes, helping the model distinguish between meaningful connections and random noise.

This method ensures embeddings learn to separate true relationships from irrelevant ones. For instance, during training, for a target node in a random walk, negative sampling selects nodes outside its context (unrelated nodes) as negative examples. These examples serve to sharpen the model's ability to identify genuine proximity relationships within the network.

To achieve this, we assume a moderate number of negative samples, starting with a value of 10, as suggested in the original article. However, this value is evaluated and adjusted based on experimental results to find the optimal trade-off between computational efficiency and embedding quality. The unigram distribution raised to the power of 0.75, as described in the original article [5], is employed, which prioritizes frequent nodes while still including fewer common ones. By doing so, the model effectively balances the representation of highly connected nodes and those with fewer connections.

The approach includes systematically tuning the number of negative samples during the analysis phase. This involves testing various configurations to ensure embeddings capture meaningful patterns in citation networks while minimizing noise and computational overhead. This strategy aims to preserve both proximity relationships and temporal continuity, enabling robust and scalable network analysis.

5.5 Tracking Evolution Articles in Citation Networks

To detect a trending article in a dynamic citation network using the Dynamic Network Embedding approach, we analyze how the node embeddings (representing articles) evolve over time. A trending article exhibits distinct and measurable behaviors that reflect its increasing importance and impact within the network:

- **Rapid Increase in Degree:** A trending article receives a sudden and notable rise in citations (edges) within a short time window. This rapid increase signifies growing recognition of the article's relevance and its influence on subsequent research. Such articles often address emerging topics, present groundbreaking methods, or offer significant insights that capture the attention of the academic community.
- **Significant Embedding Drift:** The position of the article in the embedding space changes substantially between consecutive timesteps. This drift occurs as the article attracts citations from diverse or unexpected areas, signaling its expansion beyond its original domain. Embedding drift highlights the article's evolving role and its capacity to influence interdisciplinary research or spawn new lines of inquiry.
- **Centrality Growth:** Over time, the article becomes more central within the citation network. Centrality measures, such as betweenness, closeness, or eigenvector centrality, indicate how influential a node is within the network. An article experiencing centrality growth acts as a "hub" connecting multiple research areas, fostering collaborations, and serving as a foundation for subsequent work.

These behaviors collectively provide a comprehensive view of an article's trajectory within the evolving citation network. By tracking degree growth, embedding drift, and centrality changes, researchers can identify not only articles gaining influence but also emerging research themes and fields. Combining these metrics offers a robust, multi-dimensional method for detecting trending articles and understanding their evolving impact on the research landscape.

5.6 Classification of Citation Embedding Nodes

Dynamic citation networks are rich and complex systems that evolve continuously over time, capturing the ever-shifting landscape of academic influence. By analyzing these networks using embedding techniques, we can classify nodes (articles) into meaningful categories that reveal their roles in the propagation of scientific knowledge.

To explore and analyse the results of identifying node classifications, a distribution visualization tool helps. This tool provides a clear representation of changes in node embeddings over time, highlighting nodes with significant shifts in their positions within the embedding space. By visualizing the distribution of these changes, the identifying process of node classifications can be much easier.

5.6.1 Emerging Nodes

Emerging Nodes represent nodes that have recently been published and started receiving attention within a short period. These nodes often have low initial visibility but show potential for future growth.

Emerging Nodes are essential for identifying early-stage trends within dynamic citation networks. These nodes serve as early indicators of future breakthroughs, providing valuable insights that can guide researchers toward promising fields of study. By highlighting these emerging contributions, researchers can allocate resources more effectively and foster innovation. Additionally, Emerging Nodes contribute significantly to the growth of the citation network and expanding the boundaries of scientific knowledge.

A formal method for identifying emerging nodes in dynamic networks can be developed by leveraging the principles of the Dynamic Bernoulli Embedding (DBE) model:

Each node i in the dynamic citation network is represented by an embedding vector $\mathrm{v}_i(t)$ at timestamp t, learned from the DBE model.

The DBE model optimizes node embeddings to reflect the likelihood of observed edges within a dynamic network. To identify the observed edges at a specific timestamp t, it is common to analyze $A_{ij}(t)$, the adjacency matrix representing the network at t. When a node experiences a rapid increase in connectivity, gaining many new connections between timestamps t and $t + \Delta t$, the optimization process adjusts its embedding $\mathrm{v}_i(t + \Delta t)$, bringing it significantly closer to the embeddings of its new neighbors. This significant shift in the embedding space maximizes the displacement, defined as

$$\max \{\Delta \mathbf{v}_i = \| \mathbf{v}_i(t + \Delta t) - \mathbf{v}_i(t) \|\}$$

5.6.2 Steady Nodes

Steady Nodes represent articles with consistent levels of influence and citations over time, maintaining their status as foundational or highly regarded works.

Stable nodes serve as the foundational pillars of academic fields, offering consistent and enduring influence over time. These nodes act as reliable reference points for research and providing a backbone for the development. Their long-term impact ensures the relevance of seminal works, often shaping the trajectory of subsequent studies for decades.

Similarly to the method mentioned for identifying emerging nodes, a formal approach for identifying steady nodes in dynamic networks can be developed using the principles of the Dynamic Bernoulli Embedding (DBE) model. In this framework, each node i is represented by an embedding vector $\mathrm{v}_i(t)$ at timestamp t, optimized to reflect the likelihood of observed edges in the network. To identify steady nodes, the adjacency matrix $A_{ij}(t)$, representing the network structure at t, is analyzed for stability over time.

Steady nodes are distinguished by their consistent influence and stable connectivity, which result in minimal changes to their embeddings across consecutive timestamps. The displacement of a node, calculated as:

$$\min \{\Delta \mathbf{v}_i = \| \mathbf{v}_i(t + \Delta t) - \mathbf{v}_i(t) \|\}$$

In contrast to emerging nodes, which exhibit significant shifts due to rapid increases in connectivity, steady nodes maintain a consistent presence, emphasizing their role as enduring and foundational components within the dynamic structure.

5.6.3 Rising Stars

Rising Stars are nodes that demonstrate a consistent and significant increase in their influence over time, signifying articles that rapidly gain citations and recognition.

Rising Stars play a crucial role in the dynamics of citation networks by representing impactful research that gains recognition early in its lifecycle. Identifying these nodes is essential for highlighting contributions that significantly influence academic discourse and shape the direction of future research. Moreover, Rising Stars often become prime candidates for funding and collaboration opportunities, as their increasing prominence signals potential breakthroughs and innovation.

A method for identifying Rising Stars in dynamic networks involves analysing the velocity of nodes in the embedding space. Nodes experiencing rapid increases in connectivity undergo significant changes in their embeddings to reflect new relationships. The velocity of a node i in the embedding space at time t is defined as $\mathrm{v}_i^{\text{velocity}}(t) = \frac{\mathbf{v}_i(t+\Delta t)-\mathbf{v}_i(t)}{\Delta t}$, representing the rate of change in its position. This velocity directly indicates the growth in a node's connectivity and influence within the network. Rising Stars, defined by consistent and sustained increases in connectivity and influence, exhibit **high velocity over consecutive time steps,** distinguishing them within the dynamic network.

5.6.4 Falling Stars

Falling Stars are nodes that once had significant influence but show a marked decline in citations and impact over time.

Falling Stars play a crucial role in understanding the lifecycle of academic impact. These nodes represent works that, while once influential, experience a decline in relevance and connectivity over time. By identifying Falling Stars, it becomes possible to map shifts in research focus and recognize the fading significance of older paradigms. This analysis provides insights into the processes of knowledge deprecation and the evolution of academic fields, highlighting how certain contributions lose prominence as new ideas and discoveries take their place.

Falling Stars are nodes that start with high influence or connectivity but gradually lose prominence as their relevance and connections diminish over time. In the embedding space, these nodes drift away from the high-influence cluster, which is represented by the centroid $\mathrm{v}_i^{\text{center}}(t)$. The centroid $\mathrm{v}_i^{\text{center}}(t)$ is calculated as the average position of all nodes within the high-influence cluster at time t:

$$\mathbf{v}_i^{\text{center}}(t) = \frac{1}{N}\sum_{j\in\text{cluster}} \mathbf{v}_j(t)$$

where N is the number of nodes in the cluster. The drift vector $\mathrm{d}_i(t)$ quantifies the distance of node i from the centroid and is defined as:

$$\mathrm{d}_i(t) = \mathrm{v}_i^{\text{center}}(t) - \mathrm{v}_i(t)$$

The magnitude of the drift vector, $\parallel \mathrm{d}_i(t) \parallel$, measures how far node i is from the center of influence. Falling Stars are characterized by an increasing $\parallel \mathrm{d}_i(t) \parallel$ over time, indicating their consistent movement away from the high-influence cluster as their connectivity and relevance decline.

5.7 Model Flow Chart

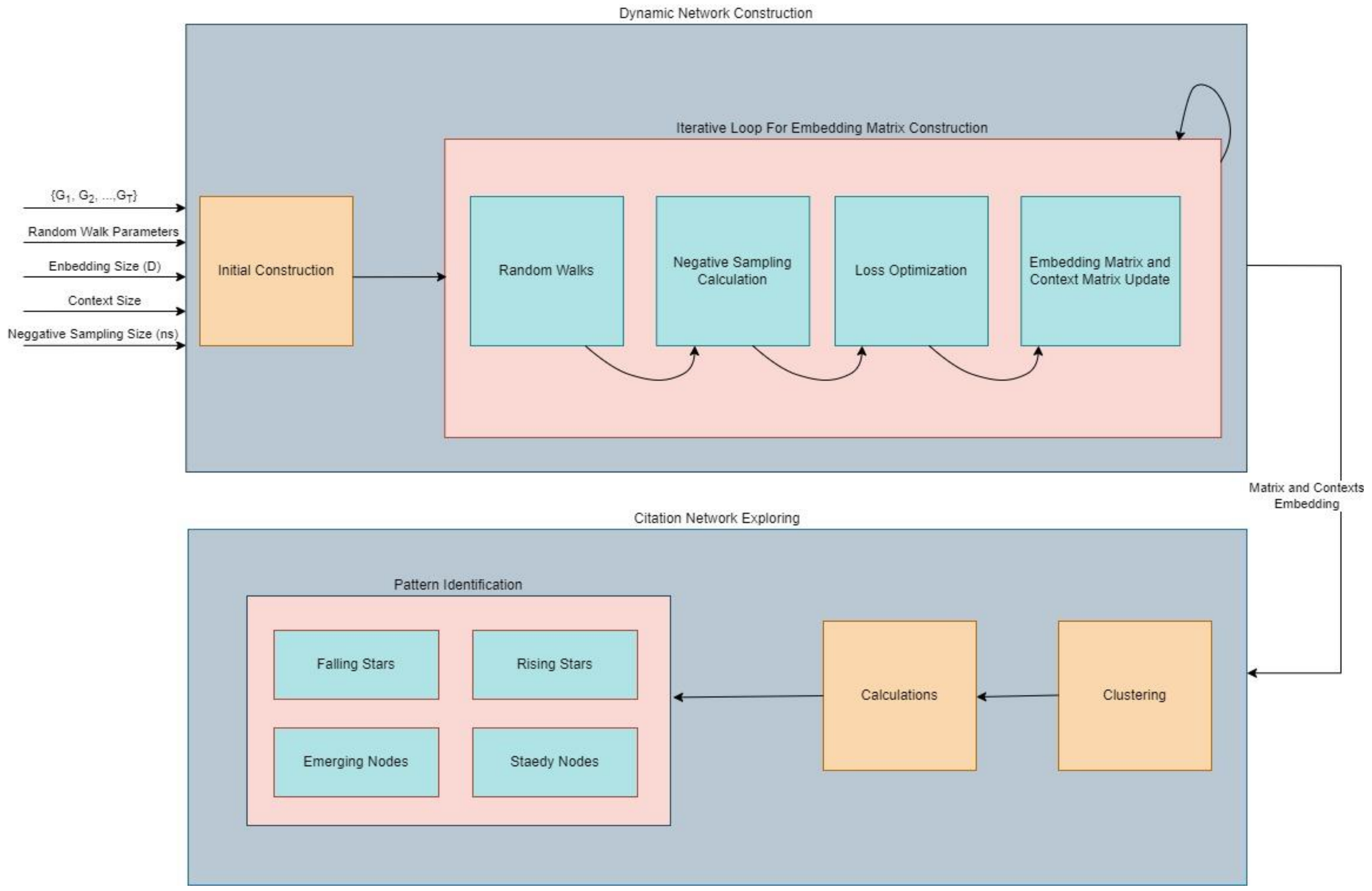
Dynamic Network Construction
Iterative Loop For Embedding Matrix Construction
$\{G_1, G_2, \ldots, G_T\}$
Random Walk Parameters
Enbedding Size (D)
Context Size
Neggative Sampling Size (ns)
Initial Construction
Random Walks
Negative Sampling Calculation
Loss Optimization
Embedding Matrix and Context Matrix Update
Matrix and Contexts Embedding
Citation Network Exploring
Pattern Identification
Falling Stars
Rising Stars
Emerging Nodes
Staedy Nodes
Calculations
Clustering

Figure 8: Final Model Flow Chart

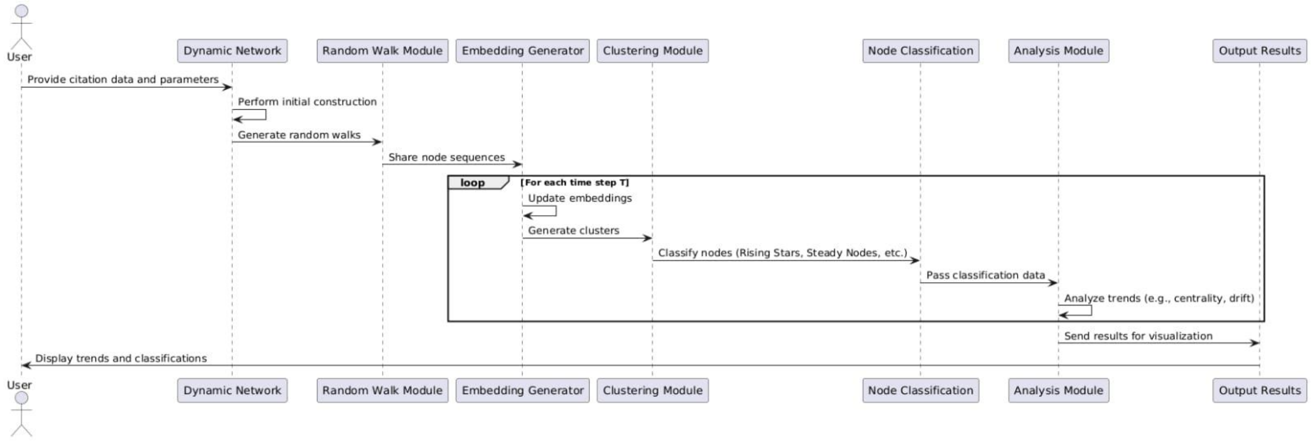

Figure 9: Sequence diagram for the classification process in dynamic citation networks using the Dynamic Bernoulli Embedding (DBE) model.

Input:

- $\Gamma = \{G_1, G_2, \dots, G_T\}, G_i = (V_i, E_i)$: Dynamic network of citations.
- L, r: Random Walk parameters.
- D : Embedding size parameter.
- cs : Context size.
- ns : Negative sample size.

Output:

- Characterizations of Rising Stars, Steady Nodes, Falling Stars and Emerging Nodes.

Steps:

```
1. $\Gamma_{\text{fixed}} = \mathit{InitialContruction}(\Gamma)$
2. for $t = 1$ to T do
    1. for each $g \in V_t^{fixed}$ do
        a. $W_g^{(t)} = RandomWalk\left(G_t^{fixed}, g, r, L\right)$
        b. $DBE(y_g^{(t)}, \alpha_g, W_g^{(t)}, V_t^{fixed}, cs, ns)$
    2. End for
3. End for
4. $\mathit{ClustedDistribution} = \mathit{Clustering}(y_g^{(t)}, \alpha_g, W_g^{(t)})$
5. for $ClusteredGroup$ in $\mathit{ClustedDistribution}$ do
    1. for $V$in $ClusteredGroup$ do
        a. Calculate $v^{center}$ and $d_v$
        b. $\mathit{FallingStarNode} = \mathit{min}(\mathit{FallingStarNode}, d_v)$
    2. End for
    3. Insert $\mathit{FallingStarNode}$ to $\mathit{FallingStarGroup}$
6. End for
7. for $t = 2$ to T do
    1. for each $g \in V_t^{fixed}$ do
        a. calculate $\Delta \mathrm{v}_i$ and $\mathrm{v}_i^{\mathrm{velocity}}$
        b. $\mathit{EmergingNode} = \mathit{max}(\mathit{EmergingNode}, \Delta v_i)$
        c. $\mathit{SteadyNode} = \mathit{min}(\mathit{EmergingNode}, \Delta v_i)$
        d. $\mathit{RisingStarNode} = \mathit{max}(\mathit{RisingStarNode}, v_i^{velocity})$
    2. End for
8. End for
9. Return $\mathit{RisingStarNode, SteadyNode, EmergingNode, FallingStarNode}$
```

6 Evaluation Plan

This evaluation plan outlines the methodology and criteria for assessing the performance, effectiveness, and scalability of the framework used to classify nodes in a dynamic citation network as Rising Stars, Steady Nodes, Falling Stars, or Emerging Nodes. The evaluation aligns with the project's expected achievements and ensures that the system meets its objectives**.**

6.1 Classification Accuracy

Classification accuracy is measured using metrics to assess how effectively the framework categorizes nodes into predefined categories (Rising Stars, Steady Nodes, Falling Stars, and Emerging Nodes). These metrics identify strengths and weaknesses in predictions, ensure reliable and meaningful results, and highlight areas for improvement. By evaluating prediction correctness, completeness, and balance, the metrics provide a comprehensive assessment of the framework's classification performance.

- **Confusion Matrix:** Summarizes classification performance by detailing the counts of true positives, true negatives, false positives, and false negatives for each category. It highlights misclassification patterns and areas where the framework struggles or fails to perform effectively. This metric helps test the classifications of citations by identifying where the model frequently misclassifies nodes, enabling targeted analysis and refinement of specific categories such as Rising Stars or Emerging Nodes.
- **Precision Matrix:** Measures the proportion of correctly classified nodes in a specific category out of all nodes predicted to belong to that category. It evaluates the quality of predictions by determining how accurate the classifications are. Precision helps test citation classifications by ensuring that the model reliably predicts categories without overestimating or including irrelevant nodes, which is especially critical for identifying Rising Stars or Falling Stars.
- **Recall Matrix:** Represents the proportion of correctly classified nodes in a specific category out of all nodes that truly belong to that category. It assesses the framework's effectiveness in identifying all relevant instances within each category. Recall is valuable for testing citation classifications by ensuring the model captures all important trends and does not overlook significant nodes, such as consistently cited articles (Steady Nodes) or newly influential ones (Emerging Nodes).

Classification accuracy is also be evaluated by applying the framework to synthetic networks with predefined node behaviors. These networks are designed to simulate clear and controlled patterns for each category, such as Rising Stars, Steady Nodes, Falling Stars, and Emerging Nodes. By comparing the framework's classification results with the expected outputs, this scenario assesses the accuracy and reliability of the model in identifying and categorizing nodes correctly, highlighting its strengths and areas needing improvement.

6.2 Scalability Evaluation

Scalability is a critical aspect of evaluating the framework's ability to handle dynamic citation networks of varying sizes and complexities. To assess this, the framework is tested on datasets that gradually increase in size, starting with small and simple networks and progressing to large, complex structures. These tests aim to determine how well the framework adapts to changes in scale while maintaining accuracy and computational efficiency.

The evaluation involves measuring key performance metrics such as execution time, memory usage as the size of the networks grows. Particular attention is given to identifying any bottlenecks or performance degradation as the dataset size increases. This step ensures the framework is not only accurate but also scalable, making it suitable for real-world applications where citation networks can be vast and continuously evolving.

6.3 Stress Evaluation

Stress evaluation is essential to evaluate the framework's ability to handle large and complex citation networks effectively. The citation field encompasses a vast amount of data, with millions of academic articles, connections, and evolving trends. Therefore, the framework must be capable of processing these large-scale datasets efficiently to remain practical for real-world applications. These tests examine how the framework performs under varying workloads, ensuring it remains reliable and scalable as the size and complexity of the datasets increase.

Stress testing push the framework to its limits by using extremely large, complex, or highly interconnected networks. This simulates the challenges of real-world citation data, where the volume and interconnections grow continuously. The results reveal the framework's robustness and stability under extreme conditions, ensuring it can manage unexpected workloads or spikes in computational demands.

6.4 Success Criteria

Success criteria for the framework focus on achieving high classification accuracy across all predefined categories, including Rising Stars, Steady Nodes, Falling Stars, and Emerging Nodes. This requires the framework to correctly and consistently classify nodes, ensuring predictions align with expected patterns and behaviours.

Another critical success criterion is the framework's ability to detect meaningful trends within the data. This includes identifying influential articles, steady contributors, and emerging or declining trends in citation behaviour.

Lastly, the system must process large datasets efficiently, maintaining scalability as the size and complexity of the citation network increase. The framework's performance should not degrade significantly with larger datasets, and it should demonstrate computational efficiency in terms of time and memory usage. This scalability ensures the framework is suitable for application in the vast and evolving citation field, meeting the demands of real-world use cases.

6.5 Unit Testing

The testing plan consists of four main categories: Dynamic Network Construction, Random Walk Generation, Embedding Generation, and Node Classification. Each category features specific tests designed to ensure accuracy, reliability, and scalability, validating the construction of networks, the consistency of embeddings, and the correct classification of nodes for effective analysis of citation networks.

Unit Under Test	Test Case	Input	Expected Output
Dynamic Network	Network Construction with Fixed Time Intervals	Citation data split into yearly intervals.	Correctly structured networks for each year, with accurate nodes and edges.
	Network Construction with Fixed Event Counts	Citation data with a specified number of events per snapshot.	Uniformly dense networks with appropriate overlaps.
Random Walk	Random Walks on Static Graphs	A static graph with defined edges.	Correct sequences capturing both local and global structures.
	Random Walks on Dynamic Graphs	A dynamic graph with changing nodes and edges.	Sequences adapting to graph changes over time.
Embedding Generator	Embeddings for Static Nodes	Nodes with consistent relationships over time.	Stable embeddings with minimal drift.
	Embeddings for Evolving Nodes	Nodes with rapidly changing connectivity.	Embeddings reflecting the change in proximity and influence.
Calculation and Classification	Classification of Rising Stars	Nodes with increasing centrality and connectivity.	Nodes correctly categorized as Rising Stars.
	Classification of Falling Stars	Nodes with declining centrality and connectivity.	Nodes correctly categorized as Falling Stars.
	Classification of Emerging and Steady Nodes	Nodes with new or consistent citation activity.	Accurate classification as Emerging or Steady Nodes.

7 References

[1] Annamalai, Narayanan; Mahinthan, Chandramohan; Rajasekar, Venkatesan; Lihui, Chen; Yang, Liu; Shantanu, Jaiswal;, “graph2vec: Learning Distributed Representations of Graphs,” 2017.

[2] B. Perozzi, R. Al-Rfou and S. Skiena, “DeepWalk: Online Learning of Social Representations,” 2014.

[3] G. Aditya and L. Jure, “node2vec: Scalable Feature Learning for Networks,” 2016.

[4] S. Eger and A. Mehler, “On the Linearity of Semantic Change: Investigating Meaning Variation via Dynamic Graph Models,” 2017.

[5] Chuanchang, Chen; Yubo, Tao; Hai, Lin;, “Dynamic Network Embeddings for Network Evolution Analysis,” 2019.

[6] M. Rudolph and D. Blei, “Dynamic Bernoulli Embeddings for Language Evolution,” 2017.

[7] G. H. Nguyen, J. B. Lee, R. A. Rossi, A. K. Nesreen, K. Eunyee and K. Sungchul, “Continuous-Time Dynamic Network Embeddings,” 2018.

[8] N. Huang and V. Sledad, “A Short Tutorial on The Weisfeiler-Lehman Test And Its Variants,” 2022.

[9] T. Mikolov, K. Chen, G. Corrado and J. Dean, “Efficient Estimation of Word Representations in Vector Space,” 2013.

[10] H. Robbins and S. Monro, “A stochastic approximation method,” 1985.

[11] D. Zhang, J. Yin and Z. Xingquan, “Network Representation Learning: A Survey,” 2018.