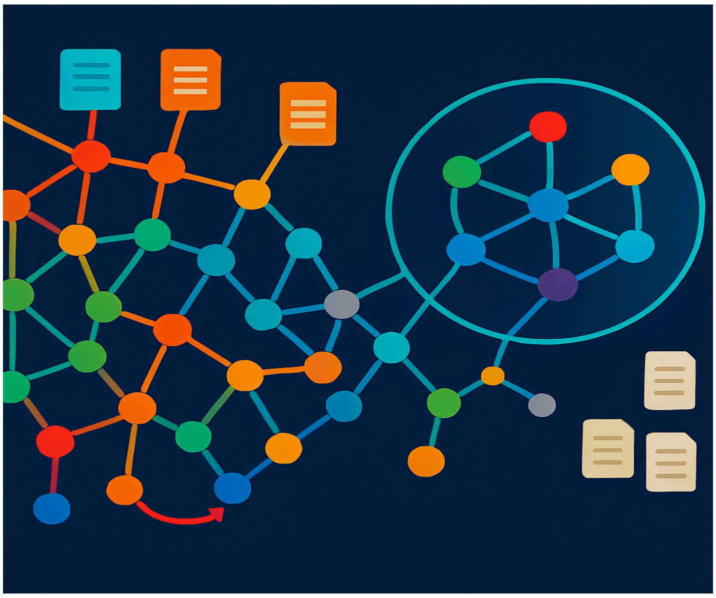
Capstone Project Phase A

25-2-R-2

Skeleton-Based Anomaly Detection in Citation Networks



Omer Goldstein

Ido Sakal

Supervisor: Dr. Dvora Toledano Kitai

Git

Table Of Contents

Abstract
Citation networks are dynamic systems that evolve as new academic works are published and referenced. These networks exhibit hierarchical structure, scale-free degree distributions, and temporally heterogeneous behaviors, features that traditional static models often fail to capture. To address this, we propose a geometry-aware and temporally sensitive framework that combines hyperbolic graph embeddings with continuous-time neural dynamics for the analysis of evolving citation networks.

The proposed method builds on the DISKNET framework, which combines Poincaré embeddings and Neural ODEs to model network dynamics. Node embeddings are initialized with Node2Vec, projected into hyperbolic space, and evolved over time. The resulting node positions are used to reconstruct a predicted citation graph, which is compared to the actual graph to detect anomalies based on deviation scores, using unsupervised methods like Isolation Forest.

The model focuses on identifying anomalous nodes whose citation behavior deviates from expected dynamics. Experimental validation on real-world citation datasets is expected to demonstrate the framework's effectiveness in capturing structural organization and temporal evolution, offering a foundation for future research in dynamic network modeling, anomaly detection, and scholarly influence analysis.

---

1. Introduction

The rapid growth of complex networked-structured data in scientific domains has prompted the development of sophisticated models that can capture both the intricate structural and temporal dynamics inherent in such systems. Among these, citation networks represent a particularly rich and dynamic form of scholarly communication, where nodes correspond to academic papers and directed edges denote citation relationships. These networks not only encode the structural interdependence of scientific knowledge but also evolve continuously as new papers are published and cited, often forming hierarchical, scale-free structures that encode both academic influence and domain-specific knowledge flows.

Traditional approaches to network analysis have often relied on static representations, which assume a fixed set of nodes and edges. While such models are effective in revealing structural properties at a particular moment in time, they are inherently limited in their capacity to capture temporal phenomena such as the rise of influential works, the formation of new research communities, or the propagation of ideas across disciplines, or the emergence of anomalous citation behaviors. Consequently, the static paradigm fails to address the essential question of how networks evolve and why certain dynamics deviate from expected trajectories.

In recent years, dynamic graph modeling has emerged as a critical research area aimed at addressing these limitations. By incorporating temporal information directly into the modeling process, dynamic graph frameworks enable more expressive understanding of network changes over time.

Recent advances in graph representation learning have further enriched this approach through graph embedding techniques, which transform high-dimensional, structured data into low-dimensional vector spaces while preserving both topological and semantic properties. These embeddings facilitate a wide range of downstream tasks, including node classification, link prediction, and anomaly detection.

This work focuses on the modeling of citation networks as dynamic systems embedded in non-Euclidean, specifically hyperbolic, geometric spaces. The inherent hierarchical structure and power-law degree distributions of citation networks render them well-suited for representation in negatively curved spaces, where hyperbolic embeddings can more naturally and efficiently encode hierarchical relationships with minimal distortion. The Poincaré disk model, in particular, offers a conformal and computationally tractable framework for such embeddings.

To capture temporal dynamics in this geometry, we employ Neural Ordinary Differential Equations (Neural ODEs), allowing us to model the continuous-time evolution of node embeddings. This methodology enables the forecasting of future states and facilitates the detection of anomalous patterns by comparing predicted and observed network configurations. The primary modeling framework explored in this work is the Dynamics-Invariant Skeleton Neural Network (DISKNET), which integrates hyperbolic embeddings and neural ODEs to learn a low-dimensional skeleton of the network and forecast future states and produce a compact, structure-preserving representation of the network's core dynamics. Deviations between these predictions and the actual network structure allow us to identify anomalous nodes, papers whose citation behavior deviate significantly from the expected pattern.

This project explores the integration of hyperbolic geometry and temporal modeling to address anomaly detection in evolving citation networks. By combining methods such as Node2Vec, hyperbolic projection, DISKNET modeling, and statistical anomaly detection (e.g., Isolation Forest), we aim to uncover latent structural patterns in scientific knowledge propagation and enhance the understanding of dynamic academic ecosystems. The work contributes to the intersection of machine learning, network science, and geometric representation learning, offering both theoretical insights and practical methodologies for the analysis of evolving complex systems.

The remainder of this document is organized as follows: Section 2 establishes the theoretical foundations underlying dynamic graph analysis, exploring how networks evolve over time and the mathematical frameworks needed to capture their temporal behavior. Section 3 introduces the core geometric and algorithmic components that form the basis of the proposed approach, focusing on the DISKNET framework and its formulation within hyperbolic space . Section 4 presents our comprehensive methodology for detecting anomalous citation patterns through temporal modeling and geometric embedding techniques. Section 5 discusses the anticipated results and evaluation expectations of the proposed anomaly detection framework. Section 6 outlines the technical infrastructure and computational tools required for implementation. Finally, Section 7 examines the theoretical and practical challenges inherent in developing the suggested geometry-aware temporal model for complex network analysis.

2. Theoretical Background

2.1 Graph Network

Graph-based data structures, commonly referred to as graph networks, offer a powerful and flexible framework for modeling complex systems composed of discrete entities and their relationships. A graph $G = (V, E)$ consists of a set of nodes (or vertices) V, representing individual entities (e.g., academic papers), and a set of edges E, which encodes relationships or interactions between these entities (e.g., citation between papers). They are widely used to model complex systems such as social interactions, biological pathways, and academic citations. A network can be either static (unchanging over time) or dynamic (evolving as new connections form or dissolve).

2.1.1 Graph Representation and Feature Modeling

To enable computational analysis and machine learning over such networks, each node is typically associated with a feature vector; a structured, fixed-length representation capturing information about the node. These features may include structural properties (like degree centrality or clustering coefficient), semantic content (such as keywords, abstracts, or topics extracted from the text), and temporal metadata (e.g., publication year). Feature-based modeling serves as the input to a wide range of machine learning techniques to network-related tasks such as node classification, link prediction, and anomaly detection.

2.1.2 Static Networks

Static networks refer to network structures in which the set of entities (nodes) and their connections (edges or relationships) are assumed to remain constant over time. These models provide a snapshot of the system at a particular moment and are widely used to analyze structural properties such as node centrality, clustering coefficients, and degree distributions.

Although static representations are effective in uncovering the underlying topology and global organization of a network, they fall short in capturing the temporal nature of many real-world systems. For example, they cannot account for the timing of interactions, the order in which connections are formed, or the evolving significance of individual node research areas over time. This temporal insensitivity limits their effectiveness in tasks that rely on sequential or time-aware behavior, such as anticipating future interactions, identifying emerging trends, or detecting irregular patterns that only appear through temporal analysis.

To better reflect the fluid nature of complex systems, researchers increasingly adopted dynamic network models, which we discussed in the following subsection.

2.1.3 Dynamic Networks

Dynamic networks extend the static framework by incorporating temporal dynamics, allowing the network structure to evolve over time. In such models, entities and their connections can appear, disappear, or change strength, reflecting the continual evolution of real-world systems. Each time step represents an updated version of the network, capturing changes such as the appearance of new nodes, the formation of new edges, and the potential fading influence of older publications.

This dynamic perspective enables a more realistic and fine-grained understanding of how networks function and change. It facilitates the analysis of temporal patterns, causal relationships, and the progression of influence or information across time. Dynamic networks are particularly well-suited for studying phenomena such as the spread of information or disease, the evolution of social or technological systems, and the detection of anomalies that unfold gradually or abruptly.

A variety of modeling techniques exist for modeling dynamic graphs, including discrete-time snapshots, temporal edge lists, and continuous-time frameworks. These representations form the foundation for time-aware computational methods and machine learning algorithms, especially for tasks such as anomaly detection and link prediction within temporally evolving networks.

2.1.4 Citation Networks

Citation networks [1] constitute a distinct class of graph-based structures in which nodes represent academic publications and edges denote citations from one paper to another. These networks capture the flow of knowledge and scholarly influence both within and across academic disciplines. A citation from paper A to paper B is represented as a directed edge pointing from A to B, indicating that A builds upon or is influenced by B. As such, citation networks inherently encode both structural and semantic relationships, reflecting the complex web of scholarly interconnectedness.

By nature, citation networks are dynamic systems. New publications are constantly being added to the network, each potentially citing existing work and thereby introducing new nodes and edges into the network. Over time, this incremental growth alters the network's topology: while some papers gain prominence and attract increasing citations, others may diminish in relevance. This dynamic behavior captures the temporal evolution of academic discourse and the shifting impact of scientific contributions over time.

Moreover, the temporal dimension of citation behavior is particularly significant. The sequence in which papers are published and subsequently cited can reveal emerging research trends, shifts in disciplinary focus, and anomalies such as delayed recognition or atypical cross-domain citations. These temporal patterns often cannot be captured adequately using static representations, which offer only a snapshot of the network at a single point in time. In contrast, modeling citation networks as dynamic systems enables more comprehensive analyses, such as identifying anomalous citation behaviors and characterizing the evolution of scholarly influence across time.

Thus, citation networks underscore the necessity of dynamic graph modeling in real-world applications, where timing, sequence, and temporal context are essential to understanding network behavior.

2.2 Graph Embedding

Embedding methods aim to encode the structural and semantic information of a graph into low-dimensional vector spaces, preserving the structural and semantic properties of the original network. This transformation facilitates the application of machine learning algorithms to graph-structured data by converting nodes and their relationships into numerical vectors that reflect important patterns and dependencies.

2.2.1 Embedding Methods in Static Graphs

In static citation networks, where the graph structure remains unchanged over time, embedding methods help uncover hidden patterns such as communities, influential entities, and patterns of similarity or co-occurrence. These representations enable various downstream tasks, including link prediction, classification, community detection, and anomaly detection.

The primary benefits of embedding methods in static networks include:

- Scalability: Compressing high-dimensional graph data into compact vectors supports faster computation and efficient storage.

- Pattern Discovery: Embeddings capture topological features and latent relationships not readily apparent in the raw graph structure.

- Integration with Machine Learning Models: Once nodes are represented as vectors, standard machine learning models can be effectively applied.

2.2.1.1 Matrix Factorization-based Embedding Methods

Matrix factorization-based embedding methods aim to represent static networks in low-dimensional vector spaces while preserving key structural characteristics. These methods begin by constructing a proximity matrix that quantifies the relationships or similarities between pairs of entities within the network. Common proximity measures may reflect co-occurrence, interaction strength, or topological distance. The resulting matrix is then decomposed using dimensionality reduction techniques such as Singular Value Decomposition (SVD), producing lower-dimensional representations whose product approximates the original proximity matrix [7]. One of the key strengths of matrix factorization approaches lies in their ability to capture both local and global patterns in the network, depending on the proximity metric used. However, these methods can be computationally intensive, particularly when applied to large-scale networks, which may limit their practical applicability without further optimization or approximation strategies.

An important representative of this family is HOPE (High-Order Proximity preserved Embedding) [7] a method specifically designed to preserve high-order and asymmetric transitivity in directed graphs, a property most embedding methods cannot preserve effectively . Its key innovation is representing each node with dual vectors (source and target) that capture directional relationships. The method reformulates the embedding problem to efficiently compute node representations without constructing complete proximity matrices, making it suitable for larger networks while maintaining theoretical guarantees on approximation quality.

2.2.1.2 Random Walk-based Embedding Methods

Random walk-based embedding methods generate low-dimensional representations by simulating sequences of transitions across the elements of a network or structured system. These methods are inspired by natural language processing techniques, particularly the Skip-gram model, where the co-occurrence of items within a contextual window reveals latent semantic or structural relationships. These approaches simulate random walks over the graph to generate node sequences that capture local and global proximity patterns. The key assumption is that elements appearing in similar contexts (i.e., within the same walking window) play similar roles or share similar properties. These sequences are then used to train models that learn embeddings by maximizing the likelihood of a node's neighborhood.

A key advantage of random walk-based methods lies in their scalability. They can be applied to large graphs without requiring explicit matrix computations, making them suitable for real-world networks such as citation graphs. Moreover, they offer a flexible trade-off between local and global structural capture depending on the walk parameters.

Among the most prominent techniques in this category are Skip-gram, DeepWalk, and node2vec.

One foundational technique in this category is the Skip-gram model [5], which was originally developed for learning word embeddings in natural language processing. It aims to predict the context of a central item (e.g., word, node, or event) within a fixed-size window. When applied in broader contexts, items (nodes) are treated as tokens in sequences generated by traversals (e.g., random walks), and their co-occurrence patterns are used to train embeddings. The model learns vector representations by maximizing the probability of observing neighboring nodes given a central node. As a result, elements that frequently appear in similar contexts acquire similar embeddings, capturing both semantic and structural similarity. This mechanism serves as the foundational learning mechanism for several embedding methods such as DeepWalk and node2vec.

DeepWalk [9] extends the Skip-gram model, by introducing a novel approach to learning latent representations of nodes by combining truncated random walks with the Skip-gram model. The method performs short random walks from each node to generate node sequences that resemble sentences. These sequences are then fed into the Skip-gram model to learn embeddings that preserve neighborhood similarity. This technique is particularly effective in capturing both local and global structures in the graph, making it particularly useful for community detection and classification tasks in citation networks. However, it lacks mechanisms to incorporate node attributes or bias the walk behavior toward specific structural features.

Building upon this, node2vec [2] enhances DeepWalk by introducing a flexible, biased random walk strategy that balances between breadth-first and depth-first graph exploration. By adjusting two hyperparameters: $\boldsymbol{p}$, which controls the likelihood of returning to the previous node, and $\boldsymbol{q}$, which biases the walk toward unexplored regions of the graph. The learned embeddings capture both homophily (similar nodes connected) and structural equivalence (nodes with similar roles). These properties make node2vec well-suited for complex citation networks, where capturing both topical similarity and structural roles is essential.

2.2.1.3 Machine Learning and Deep Learning Based Methods

Machine learning and deep learning approaches have emerged as powerful tools for encoding graph structure into vector representations using neural network architectures [13]. These approaches process both node features and topological information to learn embeddings that preserve meaningful graph properties. Unlike traditional matrix factorization, these techniques can capture non-linear relationships through multi-layer architectures and end-to-end differentiable optimization. Their adaptability makes them well-suited for a variety of graph structures while maintaining computational efficiency, particularly in large-scale applications.

Graph Convolutional Networks (GCN) [13] represent a foundational approach in this domain. GCN implements spectral graph convolutions through a localized first-order approximation that assigns weights to neighboring nodes based on the graph structure. The propagation rule incorporates symmetric normalization to prevent numerical instabilities during training. GCN has demonstrated strong performance in semi-supervised node classification tasks, as they effectively integrate node features with structural information. However, the model faces challenges with increased network depth due to over-smoothing effects that can lead to indistinguishable node representations.

To overcome some of these limitations, Graph Attention Networks (GAT) [10], introduce a self-attention mechanism that dynamically computes the importance of neighboring nodes in the graph structure. This mechanism utilizes a shared neural network to evaluate the relevance between node pairs, enabling the model to selectively focus on informative graph connections. GAT's distinctive multi-head attention approach independently calculates multiple sets of attention weights and combines their results to enhance model stability and representational capacity. This design allows GAT to achieve state-of-the-art performance on both transductive citation networks and inductive protein-protein interaction datasets without requiring matrix inversions or eigen decompositions.

2.2.2 Embedding Methods in Dynamic Graphs

Embedding methods [3] for dynamic graphs extend static techniques by incorporating temporal evolution patterns, enabling node and edge representations that reflect both structural and temporal dependencies. Unlike static embeddings, which assume a fixed graph topology, dynamic approaches model evolving node relationships over time. Two dominant modeling paradigms in recent work include recurrent neural architectures that update node embeddings sequentially, and attention-based mechanisms that adaptively weigh the importance of historical structural information. These methods face the inherent challenge of preserving both local structural properties while simultaneously capturing temporal dynamics, particularly in networks characterized by hierarchical or scale-free structures.

2.2.2.1 Snapshot-Based Methods

A prominent class of techniques within this domain is snapshot-based methods, which treat dynamic networks as sequences of static snapshots at discrete time intervals [14]. These methods typically learn independent embeddings for each network snapshot and then enforce temporal smoothness between consecutive time representations through regularization terms in the objective function. These approaches operate under the assumption that node representations evolve smoothly over time, imposing continuity constraints between embeddings at adjacent time steps. However, the snapshot paradigm often struggles to capture abrupt structural changes and fails to adequately model the diverse evolution patterns exhibited by different vertices in dynamic networks.

To address these limitations, the DynamicTriad [14] model enhances the snapshot-based approach by explicitly modeling the triadic closure process, the tendency of two connected nodes with a common neighbor to form a triangle. By analyzing the transitions of open triads to closed ones across time, quantifying transition probabilities to capture diverse vertex evolution patterns. The approach balances social homophily principles with controlled temporal smoothness constraints, enabling effective representation of both gradual network evolution and abrupt structural changes that traditional continuity-focused methods struggle to accommodate.

2.2.2.2 Deep Learning and Self-Attention Mechanism-Based Methods

Deep learning models for dynamic graphs aim to capture non-linear relationships, evolving structures, and temporal dependencies in a unified computational framework. These models often extend neural architectures, such as Graph Convolutional Networks (GCNs) and Transformer-based encoders to process time-varying graph data more effectively.

One modeling direction involves dynamically adapting the model parameters to reflect temporal changes in the graph. For example, certain architectures incorporate recurrent dynamics into the parameter space of GCNs, allowing the parameters themselves to evolve over time in response to shifts in the graph structure. An alternative approach focuses on encoding temporal information directly into the input or message-passing process, thereby enabling the model to generate time-aware node representations.

The integration of Self-attention mechanisms into deep-learning models enhances their ability to model dynamic graphs. This integration enables flexible and parallel processing of both structural and temporal features, facilitating the learning of richer, context-aware node embeddings that evolve as the graph changes.

One notable implementation of the parameter adaptation approach is EvolveGCN [8], which constitutes a significant advancement in dynamic graph representation learning. Rather than focusing solely on node embeddings, EvolveGCN models graph dynamics by evolving the GCN parameters themselves through recurrent neural networks. Two variants are proposed: EvolveGCN-H, which treats GCN weights as hidden states in a Gated Recurrent Unit (GRU) that processes node embeddings, and EvolveGCN-O, which uses Long Short-Term Memory (LSTM) cells to directly evolve the weight matrices, independent of node features. By focusing on model adaptation rather than static node embeddings, EvolveGCN effectively handles graphs where nodes frequently appear and disappear over time, thus addressing the limitations of previous methods that required consistent node presence across temporal snapshots.

Another prominent model is the Temporal Graph Attention Network (TGAT) [11] is specifically designed for dynamic graphs and operates in a fully inductive setting. Contrary to methods that rely on static adjacency matrices or fixed temporal snapshots, TGAT models the temporal evolution of graph structures using continuous-time dynamic embeddings and attention-based message passing.

TGAT integrates both temporal encoding and self-attention mechanisms to jointly capture structural and temporal dependencies. Temporal encoding is performed by projecting timestamps into a high-dimensional space using functional time encoding techniques, allowing the model to distinguish events not only by their occurrence but also by their timing. These encodings are then combined with node features and passed through a multi-head attention mechanism, which dynamically weighs the importance of past interactions. As a result, TGAT can adaptively focus on the most pertinent historical contexts when computing node representations.

A key advantage of TGAT lies in its ability to generalize to previously unseen nodes and edges, making it suitable for inductive learning tasks. The model can successfully predict future links by leveraging the temporal and structural patterns of past interactions, outperforming static graph baselines and recurrent models, particularly in settings sparse or irregular interactions.

2.3 Self-Attention Mechanism

The self-attention mechanism has become a foundational component in deep learning architectures, particularly for modeling sequential and relational data. In the context of dynamic graphs, self-attention offers a flexible framework for capturing both temporal and structural dependencies, allowing nodes to adaptively focus on the most relevant aspects of their historical states or neighborhood structures. Unlike recurrent architectures that process sequences in a fixed, step-by-step manner, self-attention computes pairwise relationships in parallel. This parallelism allows each node to assign varying degrees of importance to other nodes or prior time steps, based on learned relevance scores. This mechanism operates by comparing queries and keys to compute attention weights, which are then used to aggregate information from the corresponding values. As a result, node representations are updated by selectively attending to the most informative components of the graph.

This dynamic weighting is particularly useful in evolving networks, where the significance of historical interactions can vary over time. Self-attention effectively captures both short and long-range dependencies, making it well-suited for modeling irregular temporal patterns and abrupt structural changes.

As described in [6], the attention mechanism improves both the performance and interpretability of deep learning models by enabling them to emphasize critical information while filtering out noise. Furthermore, the use of multi-head attention enables the model to capture complex interaction patterns across diverse temporal and structural dimensions. Additionally, the inherently parallel nature of self-attention offers computational advantages over traditional sequence-based models.

2.4 Hyperbolic Space

Hyperbolic geometry is a non-Euclidean geometric space characterized by constant negative curvature. Unlike Euclidean geometry, in which the sum of angles in a triangle equals 180 degrees and parallel lines remain equidistant, hyperbolic geometry permits the divergence of parallel lines, and results in triangle angle sum less than 180 degrees. These fundamental properties lead to an exponential expansion of volume as a function of distance from a reference point, enabling the embedding of hierarchical structures with significantly lower distortion compared to Euclidean space. The negative curvature produces a metric where distances increase exponentially with radius rather than linearly, a feature that makes hyperbolic space particularly suitable for representing complex networks that exhibit power-law degree distributions, features commonly observed in real-world systems such as social networks, brain connectivity patterns, and infrastructure networks [4].

Several models exist to represent hyperbolic space, including the Poincaré disk, the hyperboloid model, and the Klein model. Among these, the Poincaré disk model is particularly useful in network science due to its conformality, preserving angles and enabling intuitive visualizations of hierarchical relationships.

In the context of dynamic graph embedding, the nonlinear distance metrics of hyperbolic space effectively reflect evolutionary patterns across multiple temporal scales. This property proves especially valuable in analyzing dynamic phenomena such as information diffusion, synchronization processes, or cascading failures in networked systems. Moreover, the dimensional efficiency of hyperbolic embeddings offers computational advantages for modeling large-scale dynamic networks, enabling the preservation of both structural and temporal properties while reducing the computational complexity. This unique combination of expressive power and efficiency positions hyperbolic geometry as a powerful tool in studying complex temporal network dynamics.

Building on the geometric properties of hyperbolic space, recent methods have leveraged hyperbolic embeddings to enhance the representation of hierarchical structures that evolve over time in dynamic networks. In particular, models such as Hyperbolic Temporal Graph Networks (HTGN) [12] have been proposed to explicitly capture temporal dynamics while preserving multi-scale hierarchical organization.

HTGN addresses the dual challenge of modeling temporal network evolution while preserving multi-level hierarchical structures. This framework extends beyond conventional approaches by explicitly capturing nested organizational patterns that dynamically reconfigure over time.

By embedding an evolving network into hyperbolic manifolds in discrete time steps, HTGN enables the tracking of hierarchical position trajectories across temporal sequences. This methodology captures dynamic transitions between hierarchical states, facilitating a quantitative analysis of structural reorganization processes, including community formation, dissolution, and shifts in hierarchical roles within complex networks.

The integration of temporal dynamics with hyperbolic embeddings presents a significant methodological advancement, allowing for the simultaneous analysis of hierarchical organization and temporal evolution within a unified geometric framework. This synergy facilitates the examination of how hierarchical structures reorganize through time within a unified mathematical framework, offering analytical advantages for understanding dynamic processes in complex systems.

3. Preliminary

3.1 Poincaré Space

The Poincaré disk model is a two-dimensional representation of hyperbolic geometry defined within a unit disk. In this model, geodesics (i.e., equivalent to straight lines) are represented either as diameters passing through the origin or as circular arcs orthogonally intersecting the disk's boundary. This space is defined as an *n*-dimensional ball in Euclidean space, centered at its origin, with a radius of $\frac{1}{\sqrt{|c|}}$ where *c* denotes the (negative) curvature parameter (typically set to $c = -1$) of the hyperbolic space. Formally, it is given by:

$$(1)\ H^{n,c} = \left\{ x \in R^{n} \mid ||x|| < \frac{1}{\sqrt{|c|}} \right\}$$

This space is equipped with a unique Riemannian metric, a method for measuring distances and angles), that accounts for its constant negative curvature. Notably, when $c = 0$, the Poincaré ball model degenerates to the standard Euclidean space R^n, as the curvature vanishes.

This mathematical structure is particularly suitable for network analysis, as it naturally accommodates complex hierarchical relationships while preserving key structural properties.

A central advantage of this model is its ability to maintain accurate distance relationships between nodes, avoiding the distortions commonly encountered in Euclidean embeddings. This allows for effective representation of both topological features and temporal dynamics in networks. By preserving natural proximity and degree relationships, the Poincaré disk model provides significant advantages when analyzing complex network structures and their evolutionary dynamics. To effectively analyze the DiskNet framework [4] in the subsequent sections, it is essential to first outline the fundamental mathematical formulations that govern operations within the Poincaré disk model.

Hyperbolic Distance Function

The hyperbolic distance between two points *x* and *y* in the Poincaré disk is defined by the following metric:

$$(2)\ d_H^c(x, y) = \frac{1}{\sqrt{|c|}} arcosh(1 - \frac{2c||x-y||^2}{(1+c||x||^2)(1+c||y||^2)})\ ,\ c < 0$$

As mentioned above, $c < 0$ is the curvature parameter. The arcosh function introduces a non-linear (exponential) relationship between Euclidean and hyperbolic distances, which is critical for preserving hierarchical structures in network embeddings.

Logarithmic Map

To facilitate computations in standard Euclidean space, the logarithmic map projects a point *z* from hyperbolic space $H^{n,c}$ to the tangent space $T_x H^{n,c}$(subspace of the Euclidean space at a reference point $x \in H^{n,c}$. The logarithmic map is given by:

$$(3)\ log_z^c\left(\theta^H\right) = \theta_Z^E = \frac{2}{\sqrt{|c|}*\lambda_z^c} arctanh(\sqrt{|c|} * \left\|- z \oplus_c \theta^H\right\|) * \frac{-z \oplus_c \theta^H}{\left\|-z \oplus_c \theta^H\right\|}$$

Here, the term $\lambda_z^c = \frac{2}{1+c||z||^2}$ acts as a scaling factor based on the point's location within the disk $H^{n,c}$, and θ^H and θ^E denote the vector representations in hyperbolic and Euclidean spaces, respectively.

The logarithmic map is employed in the project to: Model changes in a node's state over time by using a displacement vector in the tangent space, measure distances and differences between node embeddings, and enable differential computations required during model training.

The operation $\oplus_c$ denotes Möbius addition, a fundamental operation in hyperbolic geometry that preserves negative curvature and respects the geometric structure of the Poincaré disk.

Möbius addition is defined by:

$$(4)\ \ x \oplus_c y \ = \ \left(\frac{(1+2c\langle x,y\rangle+c\|y\|^2)*x+(1-c\|x\|^2)*y}{1+2c\langle x,y\rangle+c^2\|x\|^2\|y\|^2}\right) \ , \ c \ < \ 0$$

where, the inner product $\langle x, y\rangle$ and norm $||x|| \ = \sqrt{\langle x, x\rangle}$ refer to the standard Euclidean definitions.

Möbius addition is used to "translate" points within the disk while preserving the hyperbolic structure, for example, in modeling temporally adjusted node positions or evolving embeddings over time.

This transformation is particularly essential, as conventional neural network operations are inherently defined in Euclidean space, thus necessitating a consistent mapping between hyperbolic and Euclidean geometric frameworks.

When updating node embeddings through gradient-based optimization, it is essential to account for the curvature of hyperbolic space. This is achieved using the Riemannian metric tensor, which ensures that the optimization process remains consistent with the underlying hyperbolic geometry. Specifically, the gradient in hyperbolic space $\nabla_H(\cdot)$ is computed by appropriately scaling the Euclidean gradient $\nabla_E(\cdot)$ as follows:

$$(5)\ \nabla_H(\theta) \ = \frac{1}{\lambda_z^2} * \nabla_E(\theta)$$

where, $\lambda_z^c = \frac{2}{1+c||x||^2}$ is the conformal factor associated with the Poincaré ball model.

This scaling increases sharply as the point *x* approaches the boundary of the disk, capturing the exponential expansion characteristic of hyperbolic space. Consequently, the metric tensor plays a critical role in maintaining geometric consistency during learning, particularly when embeddings evolve near the disk's periphery.

3.2 The DiskNet Model

The Dynamics-Invariant Skeleton Neural Network (DISKNET) [4] is a deep learning framework designed to model long-term dynamics in complex networks. By integrating hyperbolic embeddings with neural ordinary differential equations (ODEs), DISKNET predicts future states of dynamic systems through a compact, structure-preserving representation (termed a “skeleton”) of the original graph. This skeleton captures essential topological and dynamical features, enabling efficient and accurate modeling.

A central challenge in modeling network dynamics—typically governed the equation

$$(6)\ \frac{dx_i}{dt} = f(x_i) + \sum_{j \neq i} a_{ij} g(x_i, x_j),$$

is identifying a low-dimensional representation that preserves the network’s intrinsic behavior. Here f denotes the self-dynamics of node i, and g represents pairwise coupling dynamics. DISKNET addresses this by learning an assignment matrix P that maps nodes to super-nodes while maintaining topological similarity and dynamical fidelity. Additionally, it learns aggregation and lifting functions to translate between the original and skeleton networks with minimal information loss. Traditional approaches often focus solely on topological features, overlooking that "long-term dynamics in complex networks are predominantly governed by their inherent low-dimensional manifolds, i.e., skeletons" [4].

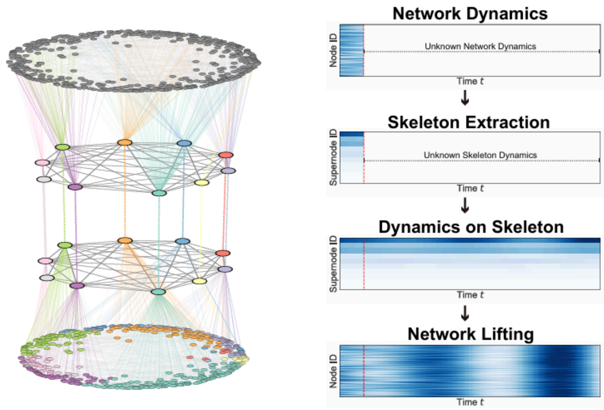


Figure 1: Long-term prediction via the skeleton of complex network dynamics. [4]

To achieve this, DISKNET introduces a Hyperbolic Renormalization Group (RG) module, which leverages the natural ability of hyperbolic geometry to preserve both topological structure and dynamic behavior. The key insight behind this module is that hierarchical and scale-free characteristics, common in real-world networks, are naturally embedded in negatively curved hyperbolic space.

Within this module, learnable hyperbolic embeddings are assigned to both original nodes and a set of coarser-grained super-nodes. Initialization is guided by physics-informed priors: angular coordinates in the Poincaré disk model are used to seed the positions of super-nodes, preserving structural features such as degree distribution and clustering coefficient, which are essential for maintaining the network's integrity during coarse-graining.

These hyperbolic embeddings are projected into Euclidean space using logarithmic maps to enable computation. The learnable assignment matrix P is then computed as

$$(7)\quad P = softmax(\widetilde{C}_S\, \widetilde{C}^T)$$

where $\widetilde{C}$ and $\widetilde{C}_S$ are transformed representations of original and super-nodes, respectively. The assignment process is regularized by two loss terms: L_E, which encourages near-discrete (one-hot) assignments, and L_R, which ensures topological consistency by comparing the original adjacency matrix A, which encodes the connectivity structure of the original network, with the reconstructed skeleton graph. The RG module uses A to guide the skeleton extraction process, ensuring that important relational patterns among nodes are retained after coarse-graining. In particular, the topological loss L_R compares the original adjacency structure with the reconstructed skeleton graph, penalizing assignments that distort the original graph's connectivity.

Once assignments are learned, the dynamic states of nodes are aggregated to super-nodes in the skeleton level using graph convolutional operations. Specifically, the state of each super-node is computed as:

$$(8)\quad X_S = PH\ ,$$

where H is the representation of original nodes obtained through a Graph Convolution Network (GCN). This aggregation enables dynamic simulation on the coarse skeleton while retaining key behavioral patterns of the original system.

After identifying a compact skeleton representation of the original network through the Hyperbolic Renormalization Group module, DISKNET models the temporal evolution of the network's dynamics directly on this reduced structure. This is achieved through a neural ordinary differential equation (Neural ODE) framework that captures continuous-time dynamics over the skeleton graph.

Unlike traditional discrete-time models, which update node states at fixed intervals, Neural ODEs define dynamics as a continuous process:

$$(9)\quad \frac{dZ_s}{dt} = f(Z_s)\ +\ g(Z_s, A_s)$$

where, Z_s are latent super-nodes states, and A_s is the skeleton adjacency matrix. The function *f* (self-dynamics) representing the self-dynamics of each super-node is implemented as a Multi-Layer Perceptron (MLP), while g (interaction dynamics) capturing interactions with neighbors is modeled via a Graph Neural Network (GNN). This formulation directly mirrors the general form of network dynamics presented in equation (6), allowing DISKNET to predict trajectories of the skeleton dynamics over arbitrary time horizons that can then be solved as an initial value problem, which refers to solving an ODE where we know the state of the system at an initial time (*t*=0) and want to determine its state at some future time *T*:

$$(10) \quad Z_{s,T} = Z_{s,0} + \int_0^T f(Z_{s,t}) + g(Z_{s,t}, A_s)dt$$

This continuous modeling approach offers two major advantages:

1) allows DISKNET to simulate long-term behavior and predict node dynamics at arbitrary time points. In particular, it aligns with the inherently continuous nature of many real-world systems.
2) Another significant advantage of operating on the skeleton graph is efficiency. It significantly reduces computational overhead, since dynamics are modeled over a compact set of super-nodes.

Once the supernode trajectories are predicted, DISKNET lifts them back to the original graph to recover node-level dynamics. A naive copy of supernode states to subnodes is insufficient, as it overlooks heterogeneity within each group such as differences in node degrees or local behavior.

To address this, DISKNET incorporates a super resolution module based on degree-based clustering. Nodes are clustered using K-means on the logarithm of their degrees, exploiting the fact that real world networks often follow power law distributions and that node degree correlates with radial position in hyperbolic space. This approach refines the coarse predictions by adjusting them with each cluster’s historical node behavior.

Despite its simplicity, this module effectively reconstructs fine grained node dynamics. Notably, prediction accuracy saturates with a relatively small number of clusters, indicating both efficiency and robustness.

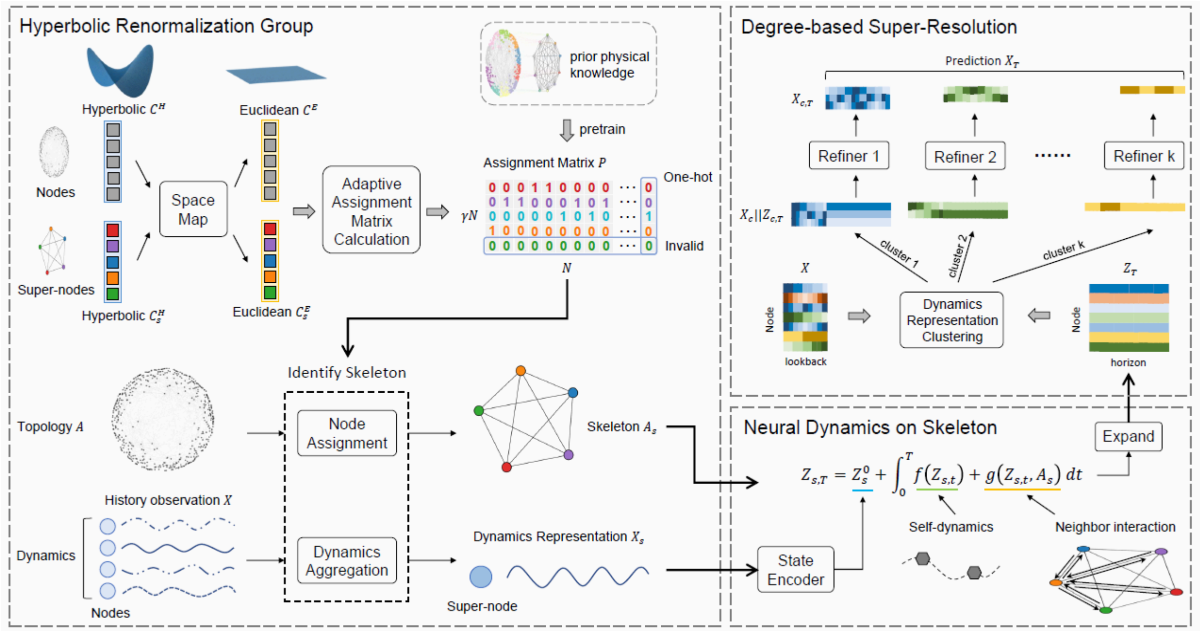


Figure 2: Overall framework of DiskNet: (1) Hyperbolic Renormalization Group, which identifies the representation and skeleton of network dynamics; (2) Neural Dynamics on Skeleton, which models the dynamics of super-nodes on the skeleton; and (3) Degree-based Super-Resolution, which lifts the predicted values of super-nodes to the original nodes. [4]

3.3 Anomaly Detection

Dynamic complex networks, such as citation systems, continuously evolve as new publications and citation links emerge over time. These networks are characterized by high
dimensionality, heavy-tailed degree distributions, and the spontaneous formation of community structures, which collectively pose significant challenges for effective anomaly detection.

Citation networks, in particular, are directed, acyclic, and inherently heterogeneous. As such, anomaly detection methods often fail to capture structurally subtle yet semantically anomalous citations patterns, such as cross-disciplinary citations or links between otherwise disconnected scientific domains. To address this, Skeleton-based approaches have been proposed, which aim to reveal these latent dynamics and interdependencies of the network into a more compact and interpretable representation. Recent work [5] proposes that such skeletons can be effectively learned using hyperbolic embeddings, due to the curvature and scale-free compatibility of hyperbolic space.

3.3.1 Isolation Forest

Isolation Forest is an efficient approach for anomaly detection based on the principle of recursively partitioning data using a tree structure. The method constructs *isolation trees*, where data points are split by randomly selecting features and corresponding thresholds. The core of this mechanism relies on the observation that anomalous instances require fewer splits to be isolated compared to normal points, as they typically occupy sparse regions in the feature space.

The algorithm builds an ensemble of binary trees through random sub-sampling. For each isolation tree, a random subset of data points is selected, and at each internal node, a random feature and split value are chosen. This recursive process continues until each data point is isolated or a predefined maximum tree depth is reached. Unlike traditional decision trees, which aim to optimize classification accuracy, isolation trees focus on the ease of isolating individual points. An anomaly score is calculated based on the normalized average path length required to isolate each point across all trees. Higher scores indicate a greater likelihood of anomalous behavior, providing an effective threshold-based classification without requiring labeled data or assumptions about normal behavior patterns.

In the context of citation networks, the algorithm processes node embeddings to capture structural patterns that deviate from typical citation behaviors. Such deviations may represent cross-disciplinary citations, novel research directions, or other atypical citation patterns. By quantifying how easily a node's representation can be isolated, Isolation Forest provides a data-driven approach for identifying potentially anomalous relationships in an unsupervised manner. This makes it particularly suitable for dynamic citation networks, where standard citation patterns evolve over time and anomalous patterns are not explicitly defined in advance.

4. Approach

4.1 Problem Formalization

The proposed approach for detecting anomalies in academic citation networks builds upon the DISKNET framework and leverages the intrinsic properties of hyperbolic geometry to effectively model dynamic graph structures over time. This section outlines the comprehensive methodology for identifying citation anomalies through a multi-stage pipeline.

We formally define a dynamic citation network as a temporal sequence of graph snapshots, denoted as $G_1, G_2, \ldots, G_T$, where each snapshot $G_t = (V_t, E_t)$ represents the state of the citation network at time t. The temporal evolution of this network is characterized by:

- V_t: The set of nodes (academic papers) present at time t, which typically grows as new publications enter the network.
- E_t: The set of directed edges (citations) between papers at time t.

This formulation captures the evolving nature of citation networks, where both nodes and edges accumulate over time as new research is published and cited. The temporal granularity (e.g., monthly, quarterly, or annual snapshots) is configurable based on the dataset characteristics and analytical requirements.

The primary objective is to identify anomalous nodes, papers that exhibit unexpected citation behavior deviating significantly from the network's predicted evolutionary trajectory. Such anomalies may include papers that experience sudden citation surges, unexpected cross-disciplinary citations, or citation patterns that contradict established academic influence flows.

4.2 Stage 0: Node Representation in Hyperbolic Space

We first generate Euclidean feature vectors for each node using an established graph embedding technique such as node2vec. This initial embedding captures both homophily (i.e., similar nodes tend to be connected) and structural equivalence (nodes with similar roles in the network).
These Euclidean embeddings are then transformed into the Poincaré disk model using a logarithmic mapping function. This transformation leverages the natural ability of hyperbolic geometry to represent hierarchical structures with minimal distortion, as described in equation (3).

4.3 Stage 1: Learning Dynamic Representations in Poincaré Space

At this stage, the DISKNET model is employed to the temporal sequence $G_1, G_2,., G_T$ to capture both spatial and temporal patterns in hyperbolic space. For each node v at time t (denoted as v_t), the model learns:

- A hyperbolic position vector $z_v^t \in H^d$, representing its location in the Poincaré disk.
- A temporal dynamic model learned via Neural Ordinary Differential Equation (Neural ODEs), describing how z_v^t evolves over time.

This dual representation captures both the static structural role of each paper within the citation network and its dynamic evolution over time.

4.4 Stage 2: Forecasting Future Positions in Hyperbolic Space

Using the learned Neural ODE, the future position z_v^{t+1} of each node is predicted, reflecting its expected evolution in the latent hyperbolic space at the next time step $t + 1$.

These predictions reflect the model's expectation regarding how each paper's influence and connectivity will naturally evolve based on historical patterns and underlying network structure.

4.5 Stage 3: Reconstruction via Lifting Operation

The forecasted hyperbolic embeddings are then mapped back to the original network space through a reconstruction operation:

1. The predicted graph $\hat{G}_{t+1}$ is reconstructed based on the set of forecasted positions $\{z_v^{t+1}\}$.
2. The suggested reconstruction criterion for edge formation is based on hyperbolic proximity. Specifically, an edge between nodes u and v is established in the reconstructed graph if their hyperbolic distance falls below a predefined threshold τ_r : $(u, v) \in \hat{E}_{t+1} \Leftrightarrow d_H^c(z_u^{t+1}, z_v^{t+1}) < \tau_r$, where d_H^c is the hyperbolic distance function defined in the Poincaré disk model.

This reconstruction produces a predicted network structure that represents the expected evolution state of citations at the next time-step.

4.6 Stage 4: Anomaly Identification Through Comparative Analysis

To identify anomalies, the predicted graph $\hat{G}_{t+1}$ is compared against the actual observed graph G_{t+1} :

1. For each node v, a deviation score is computed, quantifying the discrepancy between its predicted and actual citation patterns.
2. Several metrics can be employed for this comparison, including:
    - Absolute difference in degree centrality (citation count).

- Precision and recall of predicted citations links.

The deviation score provides a quantitative measure of how significantly a paper's actual citation behavior diverges from its forecasted trajectory.

4.7 Stage 5: Anomaly Classification

In the final stage, nodes are classified as either normal or anomalous based on their deviation scores:

1. A node is considered anomalous if its deviation score exceeds a predefined threshold, indicating a substantial deviation from expected citation behavior.
2. The threshold determination can be approached through either:
    - Statistical methods, e.g., using an unsupervised outlier detection model such as Isolation Forest.
    - Iterative validation, where the process is repeated across multiple time intervals random samples to identify nodes that consistently exhibit high anomaly scores across iterations.

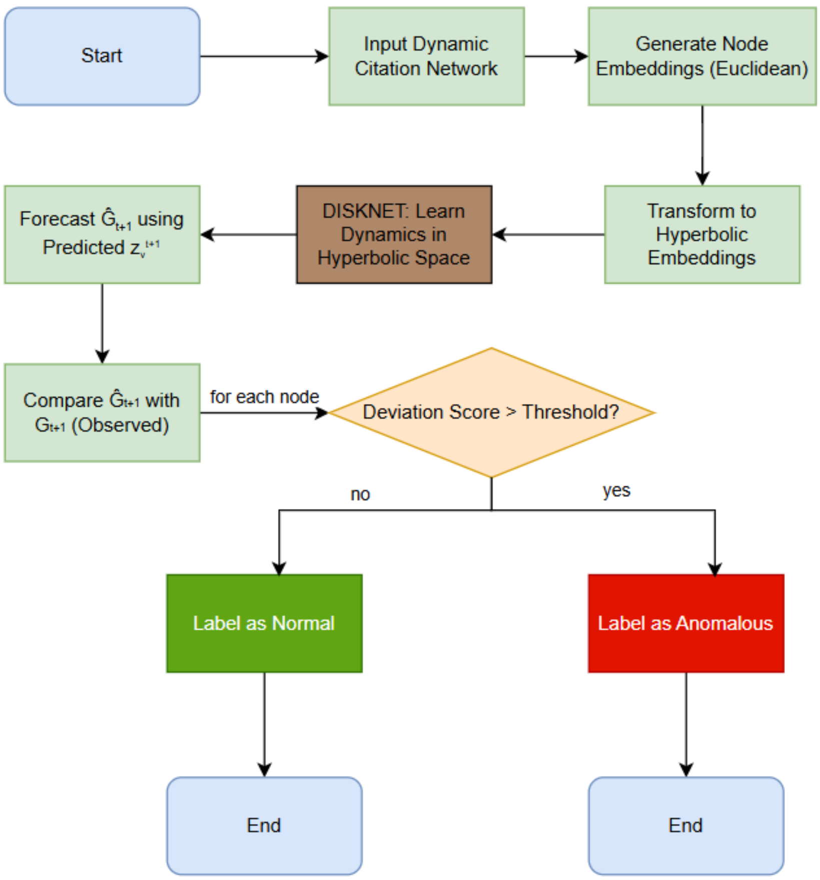


Figure 3: Anomaly Detection Pipeline in Dynamic Citation Networks Stage 0-5

4.8 Validation of Anomaly Detection Module

To evaluate the performance and robustness of the proposed anomaly detection framework, we design a controlled validation procedure based on synthetic anomaly injection and iterative testing. This approach enables systematic assessment of the model's ability to detect anomalous patterns under varied and reproducible conditions.

Synthetic Anomaly Injection

To simulate realistic yet controlled anomalies, the following modifications are applied to a clean version of the citation network:

1. Injection of artificial nodes: Between 5% and 10% of new artificial nodes are added to the graph. These nodes represent fictitious papers that are absent from the original dataset.

2. Random citation behavior: Each synthetic node is connected to 30% of the existing nodes through randomly assigned edges. These edges simulate atypical citation behavior, such as unexpected cross-domain patterns or irregular citation bursts of citations.

This procedure introduces local noise and anomalies without significantly altering the citation network's global structure, thus allowing for targeted evaluation of the detection module.

Iterative Anomaly Detection

To assess stability and consistency, the anomaly detection pipeline is executed over K independent iterations. In each iteration:

1. A new random sample of artificial nodes and anomalous edges is injected into the graph.

2. The full detection pipeline is applied, including feature extraction, hyperbolic projection, temporal modeling (via DISKNET), and deviation-based anomaly scoring.

3. Detected anomalies are recorded and compared across iterations to assess consistency and generalization.

Evaluation Metrics

1. Recall

This measures the proportion of synthetic anomalies (injected nodes) that the model correctly identifies:

$$(11)\ Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

- **High Recall** (close to 1) indicates that the model successfully detects most of the artificially injected anomalies, demonstrating sensitivity to abnormal citation behavior.

- **Low Recall** (close to 0) suggests that many anomalies go undetected, potentially due to underfitting or insufficient modeling of dynamic patterns.**H**

2. Precision

This measures the proportion of detected anomalies that are actually injected (i.e., correct):

$$(12)\ Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

- **High Precision** indicates that the anomalies flagged by the model are indeed correct, implying low false alarm rates and higher trustworthiness.

- **Low Precision** implies that many normal nodes are mistakenly labeled as anomalies, which could reduce the credibility of the detection system and overload downstream validation efforts.

By tracking both metrics, we balance the model's ability to **catch true anomalies (Recall)** and **avoid misclassifying normal behavior as abnormal (Precision)**.

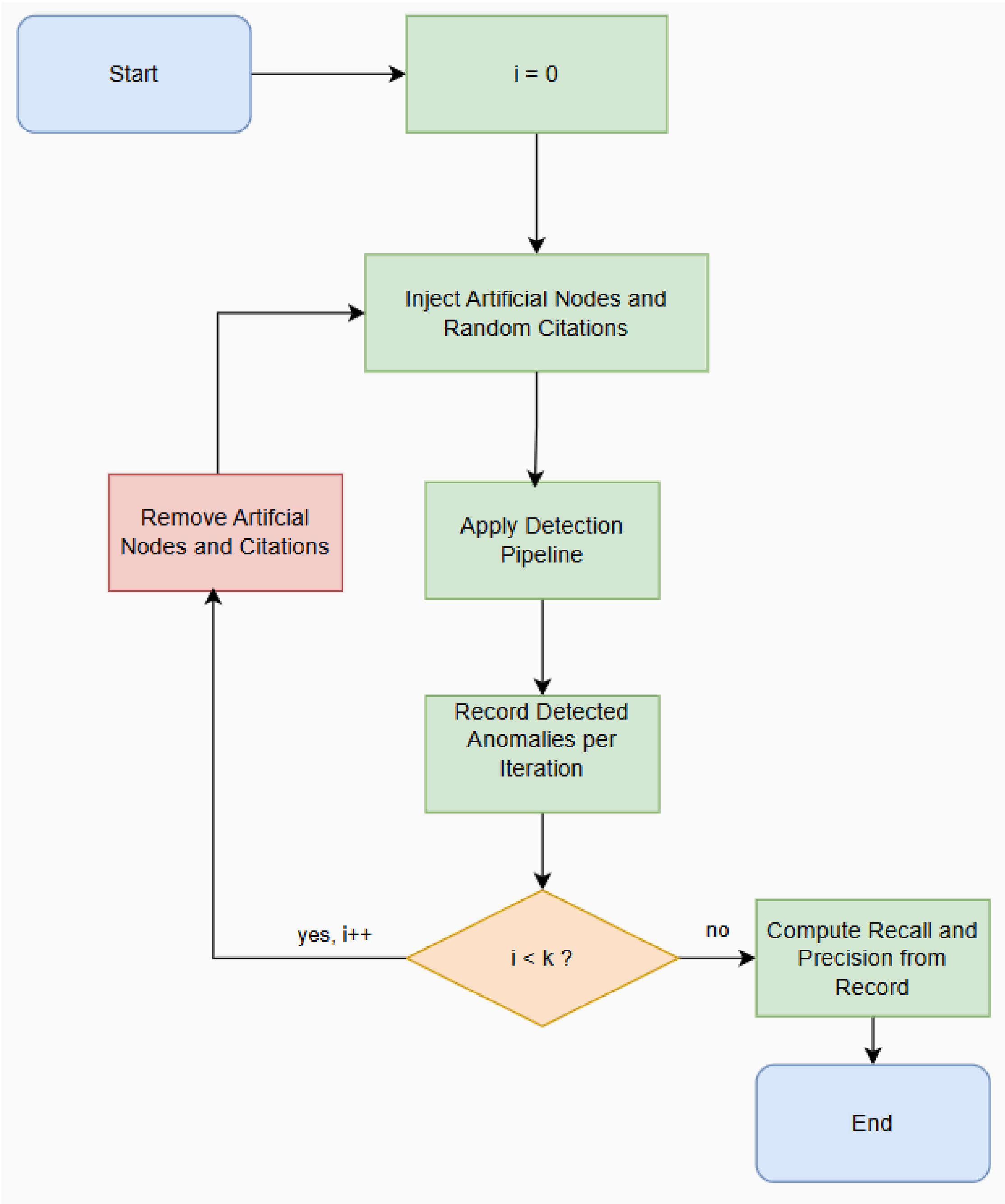


Figure 4: Validation process pipeline, step 4.8

5. Expected Results

This study is expected to provide a coherent framework for identifying atypical citation behaviors through dynamic analysis of scholarly networks. By applying the proposed methodology to real-world citation datasets (i.e., CORA, PubMed), the model is anticipated to detect deviations that reflect either emerging interdisciplinary influences or irregular patterns not aligned with prevailing citation trends.

The forecasted positions of nodes within the latent space, when compared against actual citation behaviors, are expected to produce measurable deviation scores. These scores should enable the ranking and classification of papers according to the degree of divergence from the predicted citation patterns. Anomalous instances are anticipated to correspond to phenomena such as delayed recognition, sudden citation bursts, or structurally uncharacteristic references.

In addition to anomaly detection, the model is expected to demonstrate consistency in forecasting node trajectories over time in the hyperbolic latent space. While not designed to explicitly predict future citation links, the accurate modeling of temporal dynamics is expected to support a deeper understanding of the evolving structure of scholarly networks. Given the model's use of hyperbolic embeddings to preserve hierarchical relationships and Neural Ordinary Differential Equations (Neural ODEs) to capture continuous temporal dynamics, we estimate that it can achieve a precision (equation 11) of approximately 80% and a recall (equation 12) around 85% under standard benchmark conditions. These projections are grounded in the model’s theoretical ability to capture subtle structural and temporal deviations more effectively than traditional Euclidean or discrete-time models.

The hyperbolic space facilitates clearer separation of structurally distinct nodes, while the continuous-time formulation provided by Neural ODEs improves the modeling of smooth but nonlinear citation trajectories. Together, these components are expected to improve the system’s ability to detect anomalies that deviate meaningfully from expected citation patterns over time.

6. Implementation Framework: Codebase, Language & Libraries

The implementation of this project will be based on the Python programming ecosystem, which provides a mature and versatile environment for scientific computing, machine learning, and graph -based modeling. Python’s readability, wide community support, and extensive range of libraries make it particularly well-suited for experimentation, rapid prototyping, and deep learning workflows.

Programming Language

Python (version 3.10 or higher)
Python offers robust tools for data manipulation, numerical computation, and model development. Its ecosystem supports advanced neural modeling and graph processing, making it ideal for implementing the proposed dynamic citation network pipeline.

Core Libraries and their Roles

Library	Purpose
PyTorch	Deep learning framework used for implementing Neural ODEs and GNN components
torchdiffeq	Enables numerical solvers for Neural Ordinary Differential Equations
NetworkX	Used for For constructing, analyzing, and manipulating citation graph structures
Gensim	Supports r text preprocessing and feature extraction (e.g., TF-IDF, BoW)
scikit-learn	Used for feature scaling, dimensionality reduction (PCA), and anomaly detection (e.g., Isolation Forest)
Geoopt / geoopt-manifolds	Manages hyperbolic operations, including Möbius operations and Poincaré embeddings
NumPy / SciPy	Provides efficient tools for linear algebra and matrix operations
Matplotlib / Seaborn	Use for visualizations of graph dynamic and embeddings trajectories

Runtime Environment

The project is designed to operate in a reproducible, cloud-based, and GPU-accelerated environment to support efficient training, evaluation, and experimentation. Google Colab Pro+ will serve as the primary computational platform, providing access to high-performance GPUs necessary for training Neural ODEs and processing large-scale citation networks with hyperbolic embeddings.

This environment offers the necessary computational resources to handle the complex mathematical operations required involved in manifold optimization, logarithmic and exponential mappings, and iterative anomaly detection procedures. Additionally, the platform's accessibility and integration with Python-based machine learning frameworks (e.g., PyTorch, scikit-learn), ensures compatibility and ease of development throughout all phases of this research project.

The choice of Google Colab also enhances accessibility and collaboration while maintaining a cost-effective infrastructure for model prototyping and experimentation. Overall, this platform is well-suited to meet the computational and research demands of the proposed system.

7. Design and Implementation Challenges

7.1 Challenges During the Design Phase

1. **Selection of Geometric Framework**
   Adopting hyperbolic geometry over the Euclidean model required theoretical careful justification. Although hyperbolic space is well-suited for modeling hierarchical and scale-free nature of citation networks, it introduces mathematical complexity that must be rigorously managed throughout the model.

2. **Graph Dynamics Modeling Strategy**
   Designing a framework that combines spatial and temporal modeling, combining Node2Vec initialization, hyperbolic projection, and Neural Ordinary Differential Equations (ODEs), posed significant design challenges. Ensuring theoretical consistency between embedding spaces and temporal transitions required careful sequencing and consistency checks, particularly in aligning discrete graph snapshots with continuous-time dynamic modeling.

3. **Skeleton Graph Construction**
   A central challenge in the design phase was understanding how to incorporate a skeleton-based representation, as proposed in the DISKNET framework. The goal was to ensure that the reduced structure would eventually preserve topological and dynamic properties of the original citation network. Selecting a method that supports coarse-graining while minimizing information loss required careful analysis of DISKNET’s assumptions and adaptation potential to our use case.

4. **Anomaly Definition and Measurement**
   Formulating a generalizable definition of “anomalous” citation behavior was conceptually challenging. The goal was to design a deviation scoring and comparative reconstruction method needed to capture meaningful divergence across time and datasets, without overfitting to local noise or transient structural variations.

7.2 Expected Challenges During Implementation

1. **Scalability of Hyperbolic Embeddings**
   Handling large citation datasets in hyperbolic space introduces significant computational demands. Operations such as Möbius addition, logarithmic and exponential maps, and gradient computation on manifolds are more complex than their Euclidean equivalents and may become performance bottlenecks.

2. **Numerical Stability in Neural ODEs**
   Training Neural ODEs over long temporal horizons or with sparse graph structures may lead to numerical instability. Careful selection of solvers, step sizes, and regularization techniques must be taken with ODE to prevent issues such as exploding or vanishing gradients and to ensure convergence during training.

3. **Sparse and Noisy Citation Data**
   Real-world citation networks often suffer from incompleteness, ambiguous references, and inconsistent metadata. These inconsistencies may affect the quality of the initial feature extraction and propagate through the learning pipeline, impacting the accuracy of anomaly detection pipeline.

4. **Evaluation Ground Truth Limitations**
   Validating the anomaly detection module is particularly difficult due to the lack of labeled ground truth in real-world scenarios. The proposed synthetic injection strategy partially addresses this, but evaluation may still be limited in scope.

5. **Interfacing Between Geometric and Statistical Modules**
   The system integrates multiple paradigms—hyperbolic geometry, deep learning, and statistical outlier detectionת each with its own assumptions and data representations.

   Ensuring smooth integration between geometric modeling (DISKNET), temporal dynamics (ODEs), and statistical classifiers (e.g.,Isolation Forest) require careful handling of vector transformations and data consistency across modules.

6. **Dataset Selection Challenges**
   Selecting an appropriate citation dataset poses several critical challenges that directly impact the feasibility and reliability of the proposed implementation. Many publicly available datasets suffer from key limitations that hinder their suitability for dynamic network modeling.

   First, a lack of temporal granularity in many datasets prevents year-specific citation analysis required for modeling time-evolving citation behavior.

Additionally, some datasets are outdated or no longer actively maintained, limiting their relevance to current citation dynamics and scholarly trends. Another major limitation is the absence of pre-computed node features such as textual embeddings or author metadata, which necessitates extensive preprocessing. Moreover, datasets often include non-academic or noisy entries, requiring additional filtering steps to isolate legitimate scholarly publications. Finally, incomplete or missing citation edges are a common issue, resulting in networks that do not accurately reflect the true structure of scholarly influence. Such deficiencies can compromise both the training process and the validity of the model's predictions, especially in tasks involving link prediction and anomaly detection.

References

[1] Eugene, Garfield (1979) “Citation Indexing: Its Theory and Application in Science, Technology, and Humanities.” NewYork: Wiley.

[2] Grover, A., & Leskovec, J. (2016, August). node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 855-864).

[3] Li, H., Jiang, H., Ye, D., Wang, Q., Du, L., Zeng, Y., . & Chen, C. (2024). DHGAT: Hyperbolic representation learning on dynamic graphs via attention networks. *Neurocomputing*, *568*, 127038.

[4] Li, R., Wang, H., Piao, J., Liao, Q., & Li, Y. (2024, August). Predicting Long-term Dynamics of Complex Networks via Identifying Skeleton in Hyperbolic Space. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (pp. 1655-1666)**.**

[5] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

[6] Niu, Z., Zhong, G., & Yu, H. (2021). A review on the attention mechanism of deep learning. Neurocomputing, 452, 48–62.

[7] Ou, M., Cui, P., Pei, J., Zhang, Z., & Zhu, W. (2016, August). Asymmetric transitivity preserving graph embedding. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 1105–1114).

[8] Pareja, A., Domeniconi, G., Chen, J., Ma, T., Suzumura, T., Kanezashi, H., . & Leiserson, C. (2020, April). Evolvegcn: Evolving graph convolutional networks for dynamic graphs. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 34, No. 04, pp. 5363–5370).

[9] Perozzi, B., Al-Rfou, R., & Skiena, S. (2014, August). Deepwalk: Online learning of social representations. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 701–710).

[10] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., & Bengio, Y. (2017). Graph attention networks. arXiv preprint arXiv:1710.10903.

[11] Xu, D., Ruan, C., Korpeoglu, E., Kumar, S., & Achan, K. (2020). Inductive representation learning on temporal graphs. arXiv preprint arXiv:2002.07962.

[12] Yang, M., Zhou, M., Kalander, M., Huang, Z., & King, I. (2021, August). Discrete-time temporal network embedding via implicit hierarchical learning in hyperbolic space. In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining (pp. 1975–1985).

[13] Zhang, S., Tong, H., Xu, J., & Maciejewski, R. (2019). Graph convolutional networks: A comprehensive review. Computational Social Networks, 6(1), 1–23.

[14] Zhou, L., Yang, Y., Ren, X., Wu, F., & Zhuang, Y. (2018, April). Dynamic network embedding by modeling triadic closure process. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 32, No. 1).