Temporal Graph Neural Networks For Paper Recommendation on Dynamic Citation Networks

Cheng Huang[1,*,†], Mohammad Ausaf Ali Haqqani[1,†], Beichen Hu[1], Junhao Shen[1], Xihao Xie[1] and Jia Zhang[1]

[1]*Department of Computer Science, Southern Methodist University, Dallas, USA*

Abstract

Due to the rapid growth of scientific publications, identifying all related reference articles has become increasingly challenging but highly demanding. Existing methods mostly evaluate a candidate publication document from a static perspective, for example articles' content and their citation relationship or structure information in a static view. There is still a lack of research on the impact of the papers in a dynamic way. In contrast, our work adds a time dimension to the consideration. The collection of the embeddings of the paper over the past years thus forms a time-series update.For each particular date after a paper is published, its embedding is updated through a Temporal Graph Neural Networks(TGN), to enrich its content-based embedding with people's views of the paper, i.e., the papers that cite the paper. A learnable memory update module based on the Recurrent Neural Network(RNN) is applied to study the evolution of the embedding of the paper in order to predict its reference impact in a future timestamp. Such a TGN based model learns a new pattern of how people's views of the paper may evolve over time, aiming to guide paper recommendation more precisely. We performed extensive experiments on a new citation networks dataset which is built on 313,278 articles from PaperWithCode have demonstrated the effectiveness of the proposed approach.

Keywords

Temporal Graph Networks, Recommendation, Citation Networks, Graph Neural Networks

1. Introduction

In the dynamic landscape of scientific research, where the rapid proliferation of publications presents both opportunities and challenges, research trends and community views of papers evolve over time. Current methods for scientific document recommendation, including content-based filtering, collaborative filtering, co-occurrence, graph-based, global relevance, and hybrid models[1], mostly train models in a static context or on several static collections of paper content. The advent of Graph Neural Networks (GNNs) has marked a significant stride in learning representations of graph-structured data, enabling graph-based methods to effectively learn citation relationships between documents. However, most GNN models, designed with static graph structures, are not fully equipped to handle the ever-evolving nature of real-world citation networks. Here, the intricate interconnections between documents (nodes) are in constant flux, evolving with each new citation. In the field of scientific document recommendation, the consideration of papers' publication timestamps and dynamic citation relationships has been largely overlooked.

We introduce a dynamic paper recommendation model by leveraging Temporal Graph Neural Networks (TGN). To capture the evolving changes in node embeddings within citation networks, we utilize a state-of-the-art TGN-based memory module[2] to update node embeddings in a continuous-time sequence. Additionally, we have implemented a learnable message module to prevent excessive message interchanges over time. These enhancements are specifically designed to exploit the temporal dynamics of citation networks, offering a more comprehensive and nuanced approach to scientific paper recommendation. They enable more effective aggregation of continuous-time dynamic interactions within the network, while our improved embedding and memory techniques ensure that the evolving impact of each document is accurately captured and represented.

To simulate the documentation recommendation process, we employ the Graph Transformer convolutional(TransConv) layer[3] to compute node embeddings with an attention mechanism. This approach synergistically combines updated node embeddings from the TGN memory module with encoded time features and citation relationships. Consequently, our downstream recommendation tasks demonstrate a performance improvement of x% over existing state-of-the-art methods in terms of paper recommendation accuracy. This superior performance underscores our model's enhanced capability to capture both the temporal structure and evolving embeddings of papers in citation networks, offering a compre-

*Corresponding author.
†These authors contributed equally.
chenghuang@smu.edu (C. Huang); malihaqqani@smu.edu (M. A. A. Haqqani); beichenh@smu.edu (B. Hu); junhaos@smu.edu (J. Shen); xihaox@smu.edu (X. Xie); jiazhang@smu.edu (J. Zhang)

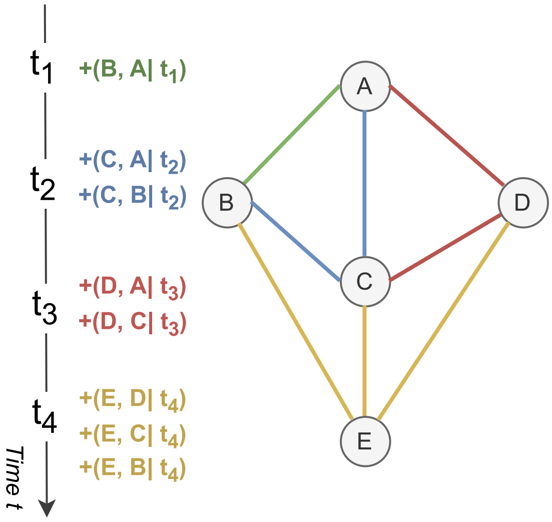


Figure 1: Illustration of dynamic citation networks, the graph will incrementally expand as time.

hensive understanding of academic influence and impact.

Our enhanced TGN-Transformer based recommendation(TGN-TRec) model marks a advancement in the field of scientific document recommendation and analysis. This research represents a vital contribution in helping researchers navigate the ever expanding corpus of scientific knowledge, as demonstrated by our comprehensive comparison with existing methods by using an abundant dataset that includes machine learning related papers from PaperWithCode, showcasing our method's superior performance. Moreover, the construction of a new citation network is another contribution in our work. This network not only enriches the existing dataset by join with reference relationship but also can be used as a benchmark for understanding the latest machine learning related works cross different domain. Thereby offering researchers a more nuanced and insightful tool for academic exploration.

2. Related Work

2.1. Graph Neural Networks

Graph Neural Networks (GNNs) have revolutionized the way we approach link prediction problems in graphs by enabling the learning of complex node representations that capture the structural context of each node within a graph [4]. Early GNN models focused on static graphs and relied on message passing mechanisms to embed nodes into a low-dimensional space, optimizing for various graph properties [5]. Recent advance of Graph Neural Networks[6, 7, 8] bring the research and model's potential in representational learning to a new level. These methods have been applied to a variety of tasks[9], ranging from social network analysis[10] to protein-protein interaction analysis[11], and knowledge graph areas[12].

2.2. Dynamics Graphs for Citation Networks

Unlike static networks, temporal networks are characterized by edges that form or dissolve over time, requiring specialized models that can account for these dynamics. The dynamic graph representation learning can learn dynamic graph that evolves over time or events[13]. There are two different dynamic graph: Discrete-time dynamic graph, Continuous-time dynamic graph. The Discrete-time dynamic graph(DTDG) are sequence of static graph snapshot over time, where the edges in each snapshot of graph have same timestamp. Discrete-time approaches segment the network into time slices and analyze each slice independently or in sequence, some approaches perform graph learn on graph snapshots by applying static methods[14, 15].

However, Continuous-time dynamic graph(CTDG) represent dynamic graph as the node pair interactions evolves over time. It can demonstrate graph's change in a more general way. Recent advance of continuous-time models aim to capture the network's evolution at a finer granularity, applying sequence based methods to update node information by capturing nodes' interaction sequentially[16, 17, 18]. The TGAT[19] propose a novel functional time-encoding module to efficiently learn dynamic interactions as graph evolves. The TGN[2]put temporal graph neural networks into a framework by proposing a RNN-based memory update module. These temporal models have been shown to be particularly effective in capturing the causality and sequential dependencies inherent in temporal networks.

2.3. Temporal Graph Neural Networks in Scientific Document Recommendation

GNN has been proved its successful application and greate potential power of application on recommendation systems[20, 21]. As a subdomain of GNN and application of recommendation system, citation networks based recommendation present a unique challenge for link prediction due to their directed nature, the evolution of research topics over time, and the presence of citation lags. Traditional heuristics such as the clustering analysis me have been applied to citation networks with limited success [22]. Machine learning approaches, particularly those employing GNNs, have shown improved performance by utilizing not only the content but also the network structure of the papers by message passing mechanism[1].

To capture dynamic nature of entities for a continuous-time bipartite graph scenario, reseachers applied Temporal Graph Sequential Recommender(TGSRec)[23] to capture dynamics collaborative signals from both users and items in a sequential patterns. However, there are barely research about recommendation method for scientific documentation that consider communities view to the exisiting paper evolving with the new citation. Different from[15] that only predict citation counts by using GNN on static snapshots of citation networks over years, our model consider edge-level timestamp that is a continuous-time evolving dynamic citation networks. For supporting the scientific documentation recommendation system, our model not predict citation counts but potential citation probabilities in future time span.

3. Proposed Method

We introduce our paper recommendation system based on the Temporal Graph Neural Network.The majorcomponents of the model contains a Temporal Neural Network(TGN) Memory Module [2] as encoder to learn paper citatation relationships in a dynamic way, and a attention-based prediction module for paper recommendation as decoder. We further implemented TGN-based encoder, by setting a self-learnable message module to adaptively compute message between nodes in order to prevent excessive messaging passing in a evolving graph.

3.1. Static Graph Representation Learning

In a static graph $\mathscr{G} = (\mathscr{V}, \mathscr{E})$, where node set $\mathscr{V} = \{v_1, v_2, ...v_N\}$, and N is number of nodes, $\mathscr{E}$ denotes collection of edges e_{ij}, where $e_{ij} = (v_i, v_j)$ for all $i, j = 1, 2, ...N$. In a graph neural networks scenario, we usually have a node features $\mathscr{X} = \{x_1, x_2, ...x_N\}$. The topological method of graph neural networks usually use message passing framework which create hidden nodes embedding $\mathbf{z}_i$ by aggregate neighbor nodes' information in the form:

$$\mathbf{z}_i = \gamma_\Theta\left(\mathbf{x}_i, \bigoplus_{j \in n(i)} \mathbf{m}_{ij}\right), \qquad \mathbf{m}_{ij} = \phi_\Theta\left(\mathbf{x}_i, \mathbf{x}_j, \mathbf{e}_{j,i}\right),$$

where e_{ij} is edge features, m_{ij} is message computed by a message module, where $\bigoplus$ denote differentiable, permutation invariant function, e.g., sum, mean, min, max or multiply, γ_Θ and ϕ_Θ denote learnable functions such as linear or attentional layer. $n(i)$ denotes neighbors node for node v_i.

3.2. Dynamic Graph Representation Learning

In a Dynamic Citation Networks, the node interactions are sequence of citation relationships between papers, for each edge(paper v_i cite paper v_j) $e_{ij}(t)$ have a timestamp t, since citation networks only have addition operation, and citation relation happens when a new node is added to the graph. We also consider the influence transferring of a existing paper in the citation networks, so the message passing is bidirectional, the existing node's embedding changes when a new node added into the graph. The temporal graph can be denote as $\mathscr{G}(\mathrm{T}) = (\mathscr{V}(\mathrm{T}), \mathscr{E}(\mathrm{T}))$, $\mathscr{G}(t)$ represent temporal citation graph in timestamp t where $t \in \mathrm{T}$. Thus the hidden node embedding st timestamp t is $\mathbf{X(t)} = \{x_1(t), x_2(t), ...x_N(t)\}$.

3.3. Memory

To capture long-term memory when a new node has been added to the graph, we use the memory module proposed in TGN[2]. The existing papers in the temporal citation graph will update their memory when new paper cite them, this module also allow the existing papers keep their original features and interaction history with other papers in a compress format. Different from the implementation in TGN, our model take papers' text embedding from SciBert as their initial state $S(t_0)$ when they are added to the citation graph. It will aggregate the message from their neighbor paper and update the memory when other new papers cite them over time. We use the same annotation to represent memory module in our model. In the memory module, we have a memory updater that is a recurrent neural network cell for updating the papers' embedding in a sequential manner. This module can save the initial memory from the paper's abstract and a historical interaction between papers along with the time evolving. In our model, we use GRU [24] as the updater, and it take aggregated information from the paper to cite events on timestamp t, the memory update format shows as follows:

$$\begin{aligned}
r &= \sigma(W_{ir}m_i(t) + b_{ir} + W_{hr}s_i(t^-) + b_{hr}) \\
z &= \sigma(W_{iz}m_i(t) + b_{iz} + W_{hz}s_i(t^-) + b_{hz}) \\
n &= \tanh\left(W_{in}m_i(t) + b_{in} + r * (W_{hn}s_i(t^-) + b_{hn})\right) \\
s_i(t) &= (1 - z) * n + z * s_i(t^-)
\end{aligned}$$

where $s_i(t)$ is the update state of node i in memory, $s_i(t^-)$ is the previous state of node i before receiving the aggregate message $m_i(t)$ from its new nodes interaction on time t.

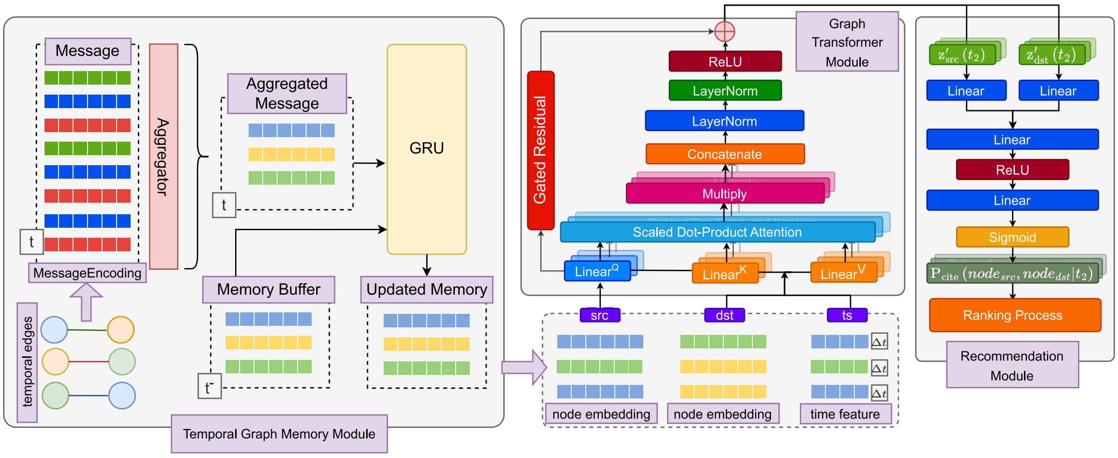


Figure 2: The illustration of the model processing batch of interactions include t_1 and t_2, the temporal graph module compute messages for each interactions and using an aggregate function to merge messages that send to each node, the GRU cell will take aggregated messages and previous state/meomory of each node and output is nodes' state/memory in latest timestamp. For performing transformer convolution operation in Graph Transformer Module, we use source node state, destination node state and edge attribute(encoded time different) as Q, K, V to a scaled dot-product attention layer. The output node embeddings of graph transformer module are used for computing citation score in Recommendation Module.

3.4. Message Module

3.4.1. Message Encoding

The messages are computed from every interaction event between new publication papers and existing papers, for considering impact transferring of a paper, we use bi-directional message passing and the message computed by following rules:

$$\mathbf{m}_i(t) = \mathrm{msg_s}\left(\mathbf{s}_i\left(t^-\right), \mathbf{s}_j\left(t^-\right), \Delta t\right), \tag{1}$$

$$\mathbf{m}_j(t) = \mathrm{msg_d}\left(\mathbf{s}_j\left(t^-\right), \mathbf{s}_i\left(t^-\right), \Delta t\right) \tag{2}$$

where $\mathbf{m}_i(t)$ represents the message that will be sent from node i to node j and verse versa. We referring the implementation in TGN which concatenate state of node $i(\mathbf{s}_i\left(t^-\right))$, node $j(\mathbf{s}_j\left(t^-\right))$ in last timestamp and Δt and encoded time difference between the current timestamp t and last timestamp t^-. msg is the message encoding module, which can be directly concatenate or processed by a self-learned linear layer where we have discussion in experiment.

3.4.2. Message Aggregator

We follow the same aggregation mechanism defined in [2] to aggregate message to a given node$i$, in our implement we compare mean aggregator and last aggregator(keep most recent message in each batch).

$$\overline{\mathbf{m}}_i(t) = \mathrm{agg}\left(\mathbf{m}_i\left(t_1\right), \ldots, \mathbf{m}_i\left(t_b\right)\right)$$

where $t_1, ..., t_b <= t_N$, t_N is the latest timestamp in each batch's interaction.

3.5. Graph Transformer Module

Once the memory/state of each node is updated, we employ the Graph Transformer Convolution module[3] to calculate the embedding of the newly added node positioned between nodes i and node j using an attention mechanism. To simulate the recommendation process effectively, the embedding of the source node serves as the query (Q), while the embedding of the destination node, along with the timestamp features, act as key (K) and value (V) inputs for optimally fitting the scaled dot-product attention operator. This setup facilitates the updating of node embeddings in the following manner:

$$\mathbf{s}_i'(t) = \mathbf{W}_1\mathbf{s}_i(t) + \sum_{j\in\mathcal{N}(i)} \alpha_{i,j}\left(\mathbf{W}_2\mathbf{s}_j(t) + \mathbf{W}_6\phi\left(t - t^-\right)\right)$$

The attention coefficients $a_{i,j}$ computed by:

$$\alpha_{i,j} = \mathrm{softmax}\left(\frac{\left(\mathbf{W}_3\mathbf{s}_i(t)\right)^\top\left(\mathbf{W}_4\mathbf{s}_j(t) + \mathbf{W}_6\phi\left(t - t^-\right)\right)}{\sqrt{d}}\right)$$

where the $\phi(\cdot)$ is the generic time encoding function, d is the hidden size of each head.

3.6. Recommendation Predictor

In scientific paper recommendation scenario, we assume given a number of pairs of positive node pairs and negative node pairs, based on the embeddings generated from previous Temporal GNN based model, the prediciton module can clearly identify the correct citation and noise information(negative edges) efficiently. We compute the edges scores by using two linear layers to learn the embedding of the source node and the embedding of the destination node at t and combined the output to another linear layer. The feedforward network function shows as follows:

$$Score_{e_{ij}} = h_{out}(\mathbf{RELU}(h_i(\mathbf{s_i}(\mathbf{t})) \parallel h_j(\mathbf{s_j}(\mathbf{t}))))$$

where h_i, h_j and h_{out} are Linear layer, $\mathbf{s_i}(\mathbf{t})$ and $\mathbf{s_j}(\mathbf{t})$ are source node embedding and destination node embedding from GNN at time t.

4. Experimental Setup

4.1. Dataset

For exploring a novel approach to paper recommendation, our dataset source is the well-known machine learning community PaperWithCode[1]. The Papers with Code community focuses on creating a platform that associates machine learning papers, code, and datasets. It covers the latest machine learning-related papers in fields including Computer Science, Physics, Astronomy, Mathematics, and Statistics. To build the citation networks, we retrieved the reference lists of the papers by querying each paper's ArXiv ID from the SemanticScholar API[25]. After a filtering process, our dataset for the citation networks includes 313,278 articles from 1900 to 2023(show as in Table 1). The citation network contains 2,233,780 edges from 1985 to 2023. We use the number of days between the citing paper's publication date and the earliest paper's publication date as a basis to compute the edges' timestamps. The edges are sorted by timestamp, allowing the model to train the dynamic citation networks sequentially as the citation relationships are established. We utilized the abstracts and titles of all papers to generate text embeddings using SciBERT[26]. These generated embeddings are used as node features in our model.

4.2. Evaluation Metrics

To assess the performance of our TGN-TRec model for scientific paper recommendation, we employ two evaluation metrics, each offering unique insights into different aspects of the model's effectiveness. These metrics include the Mean Reciprocal Rank (MRR), Precision@K and Recall@K. Below, we detail each of these metrics and explain their relevance in the context of our model evaluation.

Mean Reciprocal Rank (MRR): We use MRR to evaluate the process that computes scores for a list of positive edges and negative edges, ordered by the probability of correctness. The metric is defined as:

$$\text{MRR} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{\text{rank}_i} \tag{3}$$

where rank_i defines the rank of the positive edge in a given list of candidate edges. Q is the total number of queries (edges) for a given source node.

Precision@K and Recall@K help to assess the effectiveness of a model in predicting a set of papers candidates by given a query paper, where 'K' in refers to the number of top recommendations considered in the evaluation. Precision@K and Recall@K are defined as:

$$\text{Precision@}K = \frac{\text{Number of Relevant Items in Top K}}{K} \tag{4}$$

$$\text{Recall@}K = \frac{\text{Number of Relevant Items in Top K}}{\text{Total Number of Relevant Items}} \tag{5}$$

4.3. Baseline

In our experimental setup, we compare our Temporal Graph based paper recommendation model under various settings with three leading static graph models: GraphSAGE[7], GAT[8] and GIN[27]. This comparison aims to highlight the effectiveness of our continuous-time approach versus traditional static models in citation network analysis. To ensure fairness, we create equivalent-sized snapshots for training, validation, and testing across all models by setting all papers before 2021 as training data, 2021-2022 as validation and 2022-March 2023 as the testing data. We explore different configurations of our TGN Transformer Based Recommendation(TGN-TRec) model as model variants for Paper recommendation:

Message Modules: We assess the impact of using a simple Identity Message Module (as per TGN's vanilla implementation) against a more complex, self-learned Message Module.

Memory Initialization: We compare memory initialization using semantic information from paper abstracts and titles (via SciBERT[26]) against a structure-only approach.

Aggregator: We assess the model with different aggregation approach, where mean stand for average messages for each node and last stand for only keep latest message for aggregation in each batch.

5. Results and Discussion

This section presents the evaluation of our TGN-TRec models in different configuration with different state-of-art baseline models by applying them to the task of scientific document recommendation. The effectiveness

[1]https://paperswithcode.com

Table 1
Dataset Statistics by Year

Year	Number of Papers	Reference Count				
		Total	Mean	Median	Min	Max
<=2010	1096	37235	33.973540	25	0	539
2011	374	14256	38.117647	32	0	231
2012	819	33290	40.647131	34	0	326
2013	3438	115239	33.519197	28	0	434
2014	5087	186356	36.633772	30	0	992
2015	8385	326163	38.898390	33	0	691
2016	12008	472393	39.339857	33	0	645
2017	16715	649326	38.846904	33	0	2644
2018	26399	1040789	39.425319	34	0	1613
2019	36890	1500774	40.682407	36	0	1149
2020	50401	2250294	44.647805	39	0	1576
2021	56821	2619295	46.097306	41	0	1086
2022	61783	2909810	47.097260	42	0	696
2023	33062	1569814	47.480915	43	0	772

Table 2
Experiment Results

Encoder	initialization	Message[1]	Aggregator	MRR	Recall[2]			Precision[3]		
					@10	@20	@50	@10	@20	@50
GAT	yes	N/A	self-attention	0.952	0.442	0.630	0.891	0.902	0.817	0.622
SAGE	yes	N/A	mean	0.960	0.442	0.631	0.891	0.900	0.817	0.623
GIN	yes	N/A	N/A	0.970	0.447	0.637	0.893	0.900	0.813	0.617
TGN-TRec	no	Id[4]	mean	0.9375	0.430	0.620	0.881	0.890	0.800	0.600
TGN-TRec	no	Sl[5]	mean	0.7817	0.445	0.635	0.871	0.902	0.820	0.620
TGN-TRec	no	Id[4]	last	0.9384	0.440	0.631	0.891	0.901	0.817	0.615
TGN-TRec	no	Sl[5]	last	0.7717	0.442	0.631	0.891	0.906	0.812	0.610
TGN-TRec	yes	Id[4]	mean	0.965	0.442	0.631	0.891	0.902	0.817	0.622
TGN-TRec	yes	Sl[5]	mean	0.960	0.440	0.620	0.881	0.902	0.817	0.622
TGN-TRec	yes	Id[4]	last	0.970	0.450	0.680	0.920	0.921	0.831	0.641
TGN-TRec	yes	Sl[5]	last	**0.975**	**0.460**	**0.690**	**0.940**	**0.925**	**0.835**	**0.645**

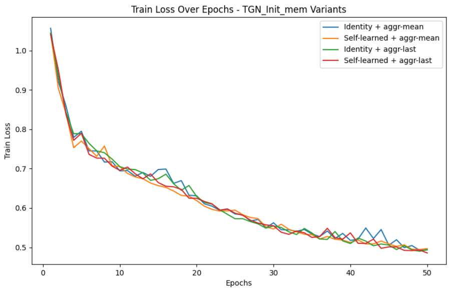
[1] Message encoding technique used in the model.
[2] Recall at different cutoffs.
[3] Precision at different cutoffs.
[4] "Id" stands for Identity.
[5] "Sl" stands for Self-learned.

of each model is assessed based on their performance in several metrics, including Mean Reciprocal Rank (MRR), Recall, and Precision at various cutoffs.

5.1. Quantitative Evaluation

We conducted extensive experiments on the Papers with Code dataset to compare the performance of the proposed TGN-TRec models against traditional static graph models like GAT and GIN. The evaluation metrics used for this comparison are MRR, Recall, and Precision, which are pivotal for assessing the recommendation quality in scientific literature.

The results tabulated in Table 2 provide a comprehensive comparison of the models' performance across various metrics. Notably, the TGN-TRec models with initialized memory exhibit superior Mean Reciprocal Rank (MRR), suggesting their enhanced ability to prioritize relevant documents. The precision metrics further validate the models' effectiveness, with the TGN-TRec variants maintaining a high degree of accuracy in the top K recommendations.



(a) Train Loss for TGN-TRec Init Memory Variants

(b) Train Loss for TGN-TRec Zero Memory Variants

Figure 3: Training Loss evolution over epochs for the TGN-TRec models with initialized and zero-initialized memory.

5.2. Training Dynamics

The training dynamics of TGN-TRec models offer a window into the learning effectiveness of these systems. By visualizing the loss function's decline across epochs, as depicted in Figure 3, we can infer not only the convergence of the training process but also the pace and stability of learning. This visualization acts as a diagnostic tool, helping to identify potential overfitting or underfitting, and whether the learning rate is appropriately tuned. A smooth, consistent decline indicates a well-tuned model making steady progress towards optimization.

A critical aspect of the TGN-TRec models' training dynamics is the role of initialization, particularly the use of SciBERT embeddings. Initialization with these embeddings appears to provide a head start to the model by leveraging pre-learned contextual representations, as reflected in the early epochs' rapid loss reduction. This suggests that the model can efficiently abstract higher-level features from the data without needing to learn from scratch, thus potentially reducing training time and resource consumption.

In this context, the TGN-TRec models' validation performance, as shown in Figure 4, encompasses several key metrics, including Mean Reciprocal Rank (MRR), Average Precision Score (APS), and Area Under the Curve Score (AUCS). These metrics collectively provide a multi-faceted view of the model's predictive power, robustness against overfitting, and its overall reliability in ranking and recommendation tasks. The TGN-TRec models, through their training dynamics, exhibit signs of such robustness. The consistency in performance metrics across epochs, particularly in scenarios with initialized memories, suggests that the model is learning a stable representation of the data that can withstand the variability inherent in real-world applications.

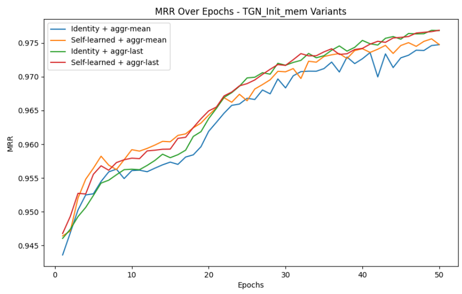
Finally, the training dynamics also shed light on the computational complexity of the TGN-TRec models. The rate of loss function decline provides indirect evidence of the model's efficiency. A steep initial decline followed by a plateau suggests that the model quickly captures the primary structure of the data but then requires more nuanced adjustments to refine its understanding. This can influence decisions around early stopping and computational resource allocation, ensuring that the model remains both effective and efficient.

5.3. Discussion

The remarkable performance of TGN-TRec models, especially those initialized with paper's text embedding, raises important considerations for recommendation systems. Their ability to accurately rank relevant papers indicates a significant advancement in the capabilities of these systems. The consideration of temporal dynamics represents a paradigm shift from static analysis to a more fluid and realistic interpretation of citation networks, which could revolutionize how academic influence is understood and quantified.

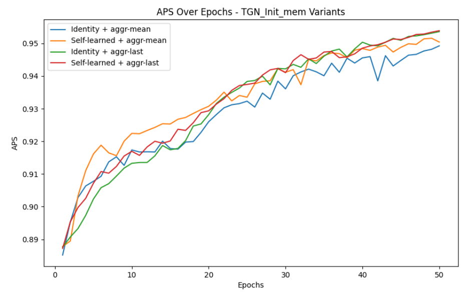
Traditional models like GAT[8] and GIN[27], despite being slightly provide same performance as TGN-TRecs without text embedding initialization. However, the the MRR evaluation here only consider the rank of positive candidate and negative candidate pair, so the MRR score is relative less representative for evaluate model performance. In addition, since the baseline models was originally design for static data, the baseline models are easily falling into overfit.

The dynamic nature of citation networks is a core aspect that TGN-TRec models capture more effectively than static models. This capability is not just a technical enhancement; it reflects the evolving nature of scientific discourse and knowledge dissemination. By accounting for the temporal aspect, TGN-TRec models align more closely with the real-world process of academic influence and its growth over time, providing a more nuanced tool

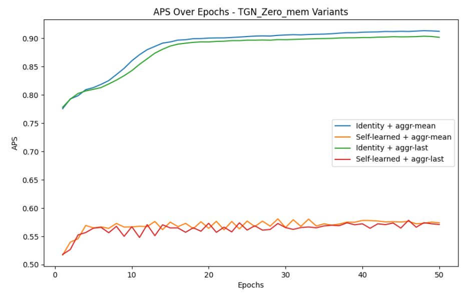


(a) MRR for TGN-TRec Init Memory Variants

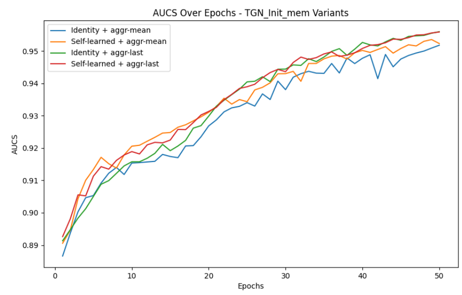
(b) MRR for TGN-TRec Zero Memory Variants



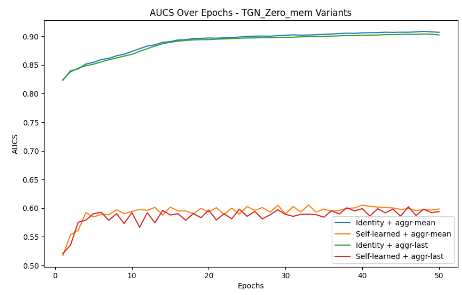
(c) APS for TGN-TRec Init Memory Variants



(d) APS for TGN-TRec Zero Memory Variants



(e) AUCS for TGN-TRec Init Memory Variants



(f) AUCS for TGN-TRec Zero Memory Variants

Figure 4: Validation MRR, APS, and AUCS evolution over epochs for the TGN-TRec models with initialized and zero-initialized memory.

for predicting future trends in scientific literature.

The potential integration of TGN-TRec models into real-world academic settings opens up exciting opportunities for enhancing the research process. By providing more accurate recommendations and predictions, these models can support researchers in staying abreast of significant developments in their field, discovering cross-disciplinary opportunities, and identifying emerging trends before they become widely recognized.

6. Conclusion and Future Work

6.1. Conclusion

In this work, we introduce a novel approach for scientific paper recommendation using Temporal Graph Neural Networks (TGNNs). This approach stands out by incorporating a time dimension, allowing us to predict the future impact of papers based on how their influence evolves over time. The method is underpinned by a robust temporal embedding strategy using graph neural networks and a memory update mechanism based on recurrent neural networks. Our experiments on a new citation networks from a latest abundant dataset–PaperWithCode demonstrate the effectiveness of our approach compared to traditional static graph methods in scientific documentation recommendation, showing its potential in addressing the dynamic nature of citation networks.

Our model captures the evolution of a paper's impact by integrating its content with the changing views of the community, as evidenced by citation patterns. This dynamic understanding of paper influence provides a more nuanced and potentially more accurate picture of a paper's significance within its field. The use of continuous-time dynamic graph representation learning is particularly significant as it allows for a more granular understanding of how a paper's influence develops and changes over time.

6.2. Future Work

Moving forward, several avenues for improvement and expansion present themselves:

1. **Scalability and Efficiency**: As the volume of scientific publications continues to grow, scalability becomes a crucial factor. Future work could focus on optimizing the model for larger datasets, possibly incorporating more efficient graph neural network architectures or by improving the memory update mechanism.
2. **Knowlegde Graph Based Method**: In our citation networks, we build homogenous graph citation networks. However, in real-world scenario, there are multiple-types of entities/nodes in a heterogeneous graph. This direction indicates us to consider a complex scenario and a more robust system.
3. **Efficient Time Encoding Function**: Our current model employs a standard time encoding function to transform timestamps into features. However, emerging research suggests that more sophisticated time-encoding methods could significantly enhance the expressive power of GNNs dealing with time-series data. Pursuing these advanced encoding techniques represents a promising avenue for further solidifying and refining our model's capabilities in temporal data analysis.
4. **Real-World Implementation and Testing**: Deploying this model in a real-world academic setting and gathering user feedback would be invaluable for iterative improvements, ensuring that the system meets the actual needs of researchers and academics.

In conclusion, while the Temporal Graph Neural Networks based model demonstrates promising results in scientific paper recommendation, its full potential can be further explored and realized through these future research directions.

Acknowledgments

The authors would like to thank...

References

[1] C. K. Kreutz, R. Schenkel, Scientific paper recommendation systems: a literature review of recent publications, 2022. arXiv:2201.00682.

[2] E. Rossi, B. P. Chamberlain, F. Frasca, D. Eynard, F. Monti, M. M. Bronstein, Temporal graph networks for deep learning on dynamic graphs, arXiv preprint arXiv:2006.10637 (2020).

[3] Y. Shi, Z. Huang, W. Wang, H. Zhong, S. Feng, Y. Sun, Masked label prediction: Unified massage passing model for semi-supervised classification, ArXiv abs/2009.03509 (2020). URL: https://api.semanticscholar.org/CorpusID:221534325.

[4] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, P. S. Yu, A comprehensive survey on graph neural networks, IEEE Transactions on Neural Networks and Learning Systems 32 (2021) 4–24. URL: http://dx.doi.org/10.1109/TNNLS.2020.2978386. doi:10.1109/tnnls.2020.2978386.

[5] A. Micheli, Neural network for graphs: A contextual constructive approach, IEEE Transactions on Neural Networks 20 (2009) 498–511. doi:10.1109/TNN.2008.2010350.

[6] T. N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, 2017. arXiv:1609.02907.

[7] W. L. Hamilton, R. Ying, J. Leskovec, Inductive representation learning on large graphs, in: Advances in neural information processing systems, 2017, pp. 1024–1034.

[8] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, Y. Bengio, Graph attention networks, 2018. arXiv:1710.10903.

[9] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, M. Sun, Graph neural networks: A review of methods and applications, 2021. arXiv:1812.08434.

[10] Y. Wu, D. Lian, Y. Xu, L. Wu, E. Chen, Graph convolutional networks with markov random field reasoning for social spammer detection, in: AAAI Conference on Artificial Intelligence, 2020. URL: https://api.semanticscholar.org/CorpusID:213709607.

[11] A. Fout, J. Byrd, B. Shariat, A. Ben-Hur, Protein interface prediction using graph convolutional

networks, in: I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems, volume 30, Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/f507783927f2ec2737ba40afbd17efb5-Paper.pdf.

[12] T. Hamaguchi, H. Oiwa, M. Shimbo, Y. Matsumoto, Knowledge transfer for out-of-knowledge-base entities: A graph neural network approach, in: Proceedings of the 26th International Joint Conference on Artificial Intelligence, IJCAI'17, AAAI Press, 2017, p. 1802–1808.

[13] S. M. Kazemi, R. Goel, K. Jain, I. Kobyzev, A. Sethi, P. Forsyth, P. Poupart, Representation learning for dynamic graphs: A survey, Journal of Machine Learning Research 21 (2020) 1–73.

[14] Y. Ma, V. Tresp, E. A. Daxberger, Embedding models for episodic knowledge graphs, Journal of Web Semantics 59 (2019) 100490. URL: https://www.sciencedirect.com/science/article/pii/S1570826818300702. doi:https://doi.org/10.1016/j.websem.2018.12.008.

[15] A. N. Holm, B. Plank, D. Wright, I. Augenstein, Longitudinal citation prediction using temporal graph neural networks, ArXiv abs/2012.05742 (2020). URL: https://api.semanticscholar.org/CorpusID:228083785.

[16] S. Kumar, X. Zhang, J. Leskovec, Predicting dynamic embedding trajectory in temporal interaction networks, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery amp; Data Mining, KDD '19, ACM, 2019. URL: http://dx.doi.org/10.1145/3292500.3330895. doi:10.1145/3292500.3330895.

[17] R. Trivedi, M. Farajtabar, P. Biswal, H. Zha, Know-evolve: Deep temporal reasoning for dynamic knowledge graphs, in: Proceedings of the 34th International Conference on Machine Learning-Volume 70, 2017, pp. 3462–3471.

[18] Z. Cui, Z. Li, S. Wu, X. Zhang, Q. Liu, L. Wang, M. Ai, Dygcn: Dynamic graph embedding with graph convolutional network, 2021. arXiv:2104.02962.

[19] D. Xu, C. Ruan, E. Korpeoglu, S. Kumar, K. Achan, Inductive representation learning on temporal graphs, in: International Conference on Learning Representations (ICLR), 2020.

[20] S. Wu, F. Sun, W. Zhang, X. Xie, B. Cui, Graph neural networks in recommender systems: A survey, 2022. arXiv:2011.02260.

[21] C. Gao, Y. Zheng, N. Li, Y. Li, Y. Qin, J. Piao, Y. Quan, J. Chang, D. Jin, X. He, Y. Li, A survey of graph neural networks for recommender systems: Challenges, methods, and directions, 2023. arXiv:2109.12843.

[22] M. Newman, Clustering and preferential attachment in growing networks, Physical Review E 64 (2001) 025102. URL: http://scholar.google.de/scholar.bib?q=info:3xHu3UP2vTgJ:scholar.google.com/&output=citation&hl=de&as_sdt=0,5&ct=citation&cd=0.

[23] Z. Fan, Z. Liu, J. Zhang, Y. Xiong, L. Zheng, P. S. Yu, Continuous-time sequential recommendation with temporal graph collaborative transformer, in: Proceedings of the 30th ACM International Conference on Information & Knowledge Management, CIKM '21, Association for Computing Machinery, New York, NY, USA, 2021, p. 433–442. URL: https://doi.org/10.1145/3459637.3482242. doi:10.1145/3459637.3482242.

[24] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using rnn encoder-decoder for statistical machine translation, 2014. arXiv:1406.1078.

[25] W. Ammar, D. Groeneveld, C. Bhagavatula, I. Beltagy, M. Crawford, D. Downey, J. Dunkelberger, A. Elgohary, S. Feldman, V. Ha, R. Kinney, S. Kohlmeier, K. Lo, T. Murray, H.-H. Ooi, M. Peters, J. Power, S. Skjonsberg, L. L. Wang, C. Wilhelm, Z. Yuan, M. van Zuylen, O. Etzioni, Construction of the literature graph in semantic scholar, in: S. Bangalore, J. Chu-Carroll, Y. Li (Eds.), Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers), Association for Computational Linguistics, New Orleans - Louisiana, 2018, pp. 84–91. URL: https://aclanthology.org/N18-3011. doi:10.18653/v1/N18-3011.

[26] I. Beltagy, K. Lo, A. Cohan, SciBERT: A pretrained language model for scientific text, in: K. Inui, J. Jiang, V. Ng, X. Wan (Eds.), Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Association for Computational Linguistics, Hong Kong, China, 2019, pp. 3615–3620. URL: https://aclanthology.org/D19-1371. doi:10.18653/v1/D19-1371.

[27] K. Xu, W. Hu, J. Leskovec, S. Jegelka, How powerful are graph neural networks?, 2019. arXiv:1810.00826.