Citation Forecasting with Multi-Context Attention-Aided Dependency Modeling

TAORAN JI, Department of Computer Science, Texas A&M University, Corpus Christi, USA
NATHAN SELF, Department of Computer Science, Virginia Tech, Arlington, USA
KAIQUN FU, Department of Computer Science, South Dakota State University, Brookings, USA
ZHIQIAN CHEN, Computer Science and Engineering Department, Mississippi State University, Starkville, USA
NAREN RAMAKRISHNAN, Department of Computer Science, Virginia Tech, Arlington, USA
CHANG-TIEN LU, Department of Computer Science, Virginia Tech, Falls Church, USA

Forecasting citations of scientific patents and publications is a crucial task for understanding the evolution and development of technological domains and for foresight into emerging technologies. By construing citations as a time series, the task can be cast into the domain of temporal point processes. Most existing work on forecasting with temporal point processes, both conventional and neural network-based, only performs single-step forecasting. In citation forecasting, however, the more salient goal is n-step forecasting: predicting the arrival of the next n citations. In this article, we propose Dynamic Multi-Context Attention Networks (DMA-Nets), a novel deep learning sequence-to-sequence (Seq2Seq) model with a novel hierarchical dynamic attention mechanism for long-term citation forecasting. Extensive experiments on two real-world datasets demonstrate that the proposed model learns better representations of conditional dependencies over historical sequences compared to state-of-the-art counterparts and thus achieves significant performance for citation predictions.

CCS Concepts: • **Computing methodologies → Machine learning algorithms;**

Additional Key Words and Phrases: Citation analysis, recurrent neural networks, deep learning

ACM Reference Format:
Taoran Ji, Nathan Self, Kaiqun Fu, Zhiqian Chen, Naren Ramakrishnan, and Chang-Tien Lu. 2024. Citation Forecasting with Multi-Context Attention-Aided Dependency Modeling. *ACM Trans. Knowl. Discov. Data.* 18, 6, Article 144 (April 2024), 23 pages. https://doi.org/10.1145/3649140

This work is supported in part by the National Science Foundation via grants Expeditions CCF-1918770, NRT DGE-1545362, OAC-1835660, and IIS-2153369. The US Government is authorized to reproduce and distribute reprints of this work for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of NSF, or the U.S. Government.
Authors' addresses: T. Ji, Department of Computer Science, Texas A&M University, CI 317, 6300 Ocean Dr., Corpus Christi, TX 78412; e-mail: taoran.ji@tamucc.edu; N. Self and N. Ramakrishnan, Department of Computer Science, VTRC-Arlington, 900 North Glebe Road, Arlington, VA 22203; e-mails: nwself@vt.edu, naren@cs.vt.edu; K. Fu, Department of Computer Science, South Dakota State University, Daktronics Eng Hall 123, Electrical Engineering/Computer Science-Box 2222, University Station, Brookings, SD 57007; e-mail: kaiqun.fu@sdstate.edu; Z. Chen, Computer Science and Engineering Department, Mississippi State University, 304 Butler Hall, 75 B. S. Hood Rd, Mississippi State, MS 39762; e-mail: zchen@cse.msstate.edu; C.-T. Lu, Department of Computer Science, Northern Virginia Center, 7054 Haycock Road, Room 312, Falls Church, VA 22043; e-mail: ctlu@vt.edu.
Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM 1556-4681/2024/04-ART144
https://doi.org/10.1145/3649140

1 INTRODUCTION

In academia and industry, innovation and evolution of technology can be thought of as the coupling of prior and new work in either incremental or disruptive fashion. The number and frequency of citations that a paper or patent receives can reflect the nature of that evolution. Indicators of an author's impact, such as g-index [11] and H-index [18], have become well-accepted standard measures which are applied to individuals, high-tech companies, and institutions alike. Patent citation statistics have been widely used for the tasks of technology impact analysis [20], patent quality assessment [3], and identifying emerging technologies at an early stage [28]. Citation forecasting is a field of growing importance due to the ever faster pace of technological change in increasingly competitive industrial and academic environments.

Many previous works [1, 55, 56] regard the citation prediction problems as feature-driven regression tasks, that is, domain-specific handcrafted features (e.g., domain keywords, topics, quality indicators, author information) are collected to formulate a regression model to predict the future citation count after a given time period. Usually, these models require prior domain knowledge and are hard to extend to different research areas. Also, model performance depends on the quality of collected features, while in real-world datasets features such as author or institution information [23] can be noisy, especially when articles from multiple disciplines are involved. Furthermore, this category of models treats features as an accumulated view over a historical window, and thus ignores crucial patterns that evolve over time.

Another group of methods treats the R&D activities as relational objects in a graph connected by different links, such as co-author relations and citations [34, 39]. In real-world scenarios, due to hardware limitations, this group of methods can only be applied to author-level analysis or a specifically tailored citation network where the total number of vertices and edges are manageable. Though some sampling techniques [13, 29] were proposed to support the graph-based methods on the large-scale network, these methods could not effectively preserve the temporal dependency, which is essential to the citation forecasting task.

Furthermore, it is worth noting that certain time series prediction methods show promise for citation forecasting [9, 47, 50]. However, these methods are primarily developed and optimized for regular time series prediction tasks, which present challenges when applied to citation forecasting due to the irregular arrival time of citations. For instance, the use of multiple **convolutional neural networks (CNNs)** to capture periodic patterns, as demonstrated in the works of [47] and [50], may not be suitable for citation forecasting given the presence of irregular time intervals and short observation windows in the context of citation data. These irregularities pose additional complexities that need to be effectively addressed in order to achieve accurate and reliable citation predictions.

To address the above challenges of citation forecasting problem, point-process-based citation prediction models [20, 27, 31] have drawn growing attention in recent years. As shown in Figure 1, the sequence of citations that reference a given paper is naturally a time series. Consequently, it can be modeled as a temporal point process that modulates the temporal pattern in a series of points. In theory, the temporal point process is characterized by a conditional intensity function learned from observing points along the timeline. Conventional methods concentrate on designing a specific parametric form of the intensity function using heuristic assumptions specific to their application [17, 37]. For instance, citation forecasting methods [31, 52] usually follow the paradigm of the general self-exciting process [16] in which intensity spikes whenever a new citation arrives. This feature is used to simulate that a highly cited paper is more likely to receive more citations. These conventional methods have two notable drawbacks: (1) heuristic assumptions may not be able to reflect complicated temporal dependencies in real datasets; and, (2) in practice, the

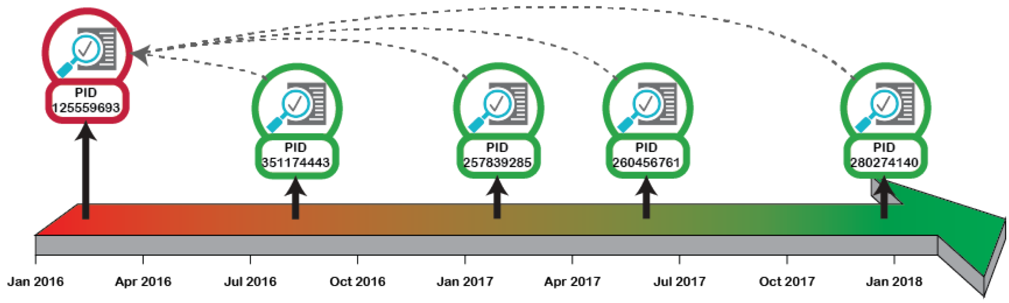

Fig. 1. The citation chain for "The Essence of Wildlife Management" from the MAG dataset. The first 4 citations are shown along the timeline with their MAG ID attached by vertical line.

complexity of the intensity function is limited because maximum likelihood estimation requires integrating intensity over time.

To address the challenges conventional models have in modeling intensity, more recent approaches use **recurrent neural networks (RNNs)** [10, 36] to approximate more complicated conditional intensity functions without heuristic assumptions or prior knowledge of dataset or application. Most existing RNN-based models [48, 51] have shown improved performance over conventional methods on both synthetic data and real-world datasets. RNN-based temporal point process models can be classified into two families: intensity-based models and end-to-end models. Intensity-based models [10, 48] use the neural network to learn the conditional intensity function. Similar to conventional methods, this conditional intensity function is integrated over time to obtain the conditional density function for maximum likelihood estimation and prediction. This group of models is optimized for observation history and suitable for one-step forecasting. However, for the task of long-term prediction in which the model inputs combine both the observation and prediction sequences, the learned intensity function is less reliable due to the lack of ground truth inputs in the prediction phase. Additionally, the integration operation is a computational bottleneck and can cause numerical instability. The family of end-to-end models [22, 53] combines the process of representing the intensity function with the process of inference. The advantage of end-to-end models is that with careful design, the model can be further optimized during the predication phase, instead of only from the observation sequence. The shortcoming of this approach is that without an intermediate intensity, predictions can not be guaranteed to monotonically increase over time.

In this article, we propose an RNN-based end-to-end model for citation forecasting. This model introduces a hierarchical dynamic attention layer which uses two temporal attention mechanisms to enforce the model's ability to represent complicated conditional dependencies in real-world datasets and allow the model to automatically balance the learning process from the observation side and prediction side. Furthermore, the temporal prediction layer guarantees that the predicted citations are monotonically increasing along the time dimension. Specifically, the contributions and highlights of this article are:

— Formulating a Seq2Seq-based framework to provide long-term citation predictions in an end-to-end fashion by integrating the process of learning intensity function representations and the process of predicting future citations.
— Designing two novel temporal attention mechanisms to improve the model's ability to modulate complicated temporal dependencies and to allow the model to dynamically combine the observation and prediction sides during the learning process.

— Conducting extensive experiments on two real-world datasets to demonstrate that our model is capable of capturing the general shape of citation sequences and can consistently outperform other models for the citation forecasting task.
— Curating two large datasets from the **United States Patent and Trademark Office (USPTO)** and **Microsoft Academic Graph (MAG)**, which can be used as citation sequences or citation networks for generalized tasks like temporal point process benchmarks and link prediction or specific tasks such as citation forecasting. The entire datasets, along with source code, are publicly available for download.[1]

The rest of this article is structured as follows: Section 2 presents the background and related work and Section 3 formulates the problem setup. Section 4 details our proposed DMA-Nets models. Experiments on two real-world datasets are presented in Section 5. The summary of the research and future work are concluded in Section 6.

2 RELATED WORK

In this section, we structure the related work into three categories: citation analysis, citation prediction, and neural point process; and discuss related concepts and background.

Citation analysis. Citation analysis has long been studied for the assessment of the impact of individual researchers [4, 5, 12, 14, 26, 41], publications [2, 6], scientific institutions [33, 40] and the investigation of the evolution of the science and technology field [35]. The typical citation analysis method performs statistical analysis on the existing citations corresponding to a set of papers collected with respect to filters of interest such as publishing journals, year of publication, and research institutions [38, 42]. For example, Braun et al. [6] combine the number of publications and the citation rate in a Hirsch-type way to rank the impact of a target journal. Another group of studies goes beyond traditional bibliometrics analysis and concentrates on the identification of citation polarity and citation purpose by performing natural language processing (NLP) on citation contexts [21]. For example, Jha et al. [21] leverage the citation context around the citing sentence to classify citation purposes into one of six categories which is helpful for applications such as measuring research dynamics and faceted summarization. Both context-based and bibliometrics-based methods usually require the collected papers to be stratified within a specific target research field; that is, they are subject to field variation and are hard to generalize. Furthermore, citation analysis methods are focused on analyzing existing citations received by a specific target paper, patent, scholar, or institution, and thus can only be used as an evaluation of the target's previous achievement. However, in the increasingly competitive industrial and academic environments, the ability to identify the potential value of emerging technologies at an early stage is more and more critical.

Citation Prediction. Citation prediction has drawn increasing attention for its ability to highlight significant areas of research, to assess the potential of emerging technologies, and to unfold promising trends in the industrial and academic environments. From the methodology point of view, the existing citation prediction models can be organized into three categories: (a) graph-based link prediction, (b) feature-driven regression methods, and (c) point-process-based methods.

One body of literature formulates the citation prediction problem as to the link prediction task in a heterogeneous correlation network, that is, to predict the probability of a citation edge between a pair of paper nodes [25, 30, 58]. Typically, this category of models serves as a supplementary paper recommendation algorithm for a keyword-based information retrieval system because the predicted probability can be considered as an indicator showing the non-context correlation between articles. For example, Yu et al. [58] use topic terms, authors' information, and publication

[1] Removed to conform with double-blind submission requirements.

venues to construct a heterogeneous bibliographic network, which is used to rank and recommend possible links between the given query to existing papers in the graph. The constructed graph is a static snapshot of the publication space at a specific timestamp, and therefore capturing evolving dynamics is not trivial; although recently, researchers [39] have started to explore involving temporal dynamics into a graph by leveraging frequent graph pattern mining. Another drawback of this category of models is that the graph usually suffers from scalability problems and is usually only suitable for a relatively small dataset.

Another group of citation prediction models [1, 55, 56] cast the citation prediction problems into the scope of feature-driven regression tasks. This group of models usually adopt domain-specific handcrafted features (e.g., domain keywords, topics, quality indicators, author information) to formulate a regression model to predict the future citation count after a given time period. For example, Yan et al. [56] extract author attributes, paper topic keywords, and venue features to build regression models to predict citation counts after 1 year. This category of methods ignores temporal evolution since the features represent an accumulated view over a historical window. Such handcrafted features require papers to be collected only from similar research areas, and thus these methods suffer from field variations. Furthermore, model performance depends on collected features, but high-quality data for attributes such as author information and paper domain knowledge are not always available. In particular, when publications from multiple disciplines are used, the author name ambiguity problem [23] will severely influence model performance by introducing too much noise.

More recently, researchers seek to overcome the above challenges of the citation forecasting task by focusing on modulating the citation sequence's temporal dynamics, leveraging the temporal point process [20, 27, 31, 43, 52–54, 57, 60]. In this way, the accumulating nature of citations is represented by an integration of the dynamic changing "intensity" of receiving new citations at different times. Mathematically, the intensity function can be formulated into this paradigm:

$$\lambda(t)dt = \Pr(\text{citation arrives in}[t, t + dt]|\mathcal{H}^*),$$

where $\mathcal{H}^*$ carries the information of all historical citations received before time t. For citation forecasting tasks, researchers usually design the intensity function as a self-exciting stochastic process, assuming its intensity jumps up whenever a new citation is received and then decreases back towards the base following a decay function. For example, a classic Hawkes process defines the self-excitation phenomenon among points as:

$$\lambda_h(t) = \alpha_0 + \beta \sum_{t_i<t} k(t, t_i),$$

where $k(t, t_i) \geq 0$ is an exponential decay kernel that reflects the declining "intensity" of occurrence of a new point at time t since the last arrival at time t_i, while the summation of kernels $\sum_{t_i<t} k(t, t_i)$ modulates the "the rich get richer" effect. In practice, the researchers adapt the intensity function to the specific applications. For example, Jang et al. [20] models the patent forward citation sequence as a Hawkes process whose intensity first increases on the arrival of new citation and then decays exponentially back. In contrast to feature-driven methods, point-process-based models can be trained and improved continuously as the arrival of new citations. As a result, this group of methods is more suitable for dynamic systems where the model is expected to work "on the fly" instead of waiting for the newly published papers to accumulate over a time window. Furthermore, these models are more robust to discipline variation since domain prior knowledge is not required. However, this category of methods usually explicitly designate an intensity function to represent the correlation structure among citations received which may not fit the real-world dataset where multiple citation patterns can co-exist [7]. For example, the Hawkes process

assumption can be violated if the combined effects of past citations are not additive or a past citation has a delayed effect on the intensity.

Neural Point Process. Recently, to overcome the challenges faced by traditional point process models, researchers further explore the use of recurrent neural networks (RNNs) [10, 36, 48, 51, 53] to enhance the model's representational ability on complicated conditional intensity functions in real-world patterns. The neural point process does not require heuristic assumptions or prior knowledge of the data to define an intensity function, but instead, it tends to automatically learn the intensity function from the data by capturing the temporal pattern with a neural network. As a result, it is demonstrated [36] to be able to capture effects that a traditional point process model misses. Also, the neural point process is more robust to domain field variation. In various tasks [48, 51, 53], the neural point process demonstrates its ability to capture the general shape of sequential data and shows better performance than the traditional point process. Most recently, researchers started to work on applying the neural point process to the long-term citation prediction task. For example, Ji et al. [22] proposed a neural point process model for patent citation forecasting by jointly modeling patent, inventor, and assignee sequence into a sequence-to-sequence structure.

3 PROBLEM FORMULATION

Let $C = \{C_1, C_2, \ldots, C_{|C|}\}$ be a set of collected citation sequences for scientific documents (e.g., a set of papers or patents). The ith sequence is denoted by $C_i = \{(t_1, m_1), (t_2, m_2), \ldots, (t_{|C_i|}, m_{|C_i|})\}$ where t_k and m_k refer to the published date and the technology class of the kth citation, and the 0th citation is the target document itself. The citation sequence can also be represented in terms of the inter-citation duration between two consecutive citations $C_i = \{(\tau_1, m_1), (\tau_2, m_2), \ldots, (\tau_{|C_i|}, m_{|C_i|})\}$ where $\tau_k = t_k - t_{k-1}$ refers to the time difference between the kth citation and the $(k-1)$th citation. These two representations are equivalent since the set of k inter-citation durations can infer the arrival timestamp of the kth citation:

$$t_k = t_{k-1} + \tau_k = t_0 + \sum_{j=1}^{k} \tau_j.$$

In this article, we use inter-citation duration notation because it makes it easier to constrain the end-to-end model to forecast citations correctly along the time dimension such that $t_{k+1} \geq t_k$.

Given data as described above, our problem is as follows: For a scientific document p, using the first l citations as observations, can we forecast the sequence of the next n citations? The question breaks down into two variants based on the focus of the problem:

(1) n-step forecasting concentrates on predicting the arrival time and technology class of the next $n \geq 1$ citations $\{(\tau_{l+1}, m_{l+1}), (\tau_{l+2}, m_{l+2}), \ldots, (\tau_{l+n}, m_{l+n})\}$, given the first l citations $\{(\tau_1, m_1), (\tau_2, m_2), \ldots, (\tau_l, m_l)\}$ of the target document as observations,
(2) one-step forecasting concentrates on predicting only the arrival time of the next citation (τ_{l+1}, m_{l+1}), given the first l citations $\{(\tau_1, m_1), (\tau_2, m_2), \ldots, (\tau_l, m_l)\}$ of the target document as observations.

The first problem is the most generalized and challenging because there are l citations on the observation side and n citations on the prediction side. There are two challenges for the task of forecasting the next n citations. First, there is a tradeoff of learning from the observation side or from the prediction side. On the one hand, observations are ground truth but there may be too few to provide enough information to modulate the temporal point process. On the other hand, predictions are less trustworthy but can provide extra information to the model for learning the temporal point process. Also, errors that occur early in the predication phase can be propagated into subsequent predictions. All these challenges indeed motivate us to adopt a sequence-to-sequence

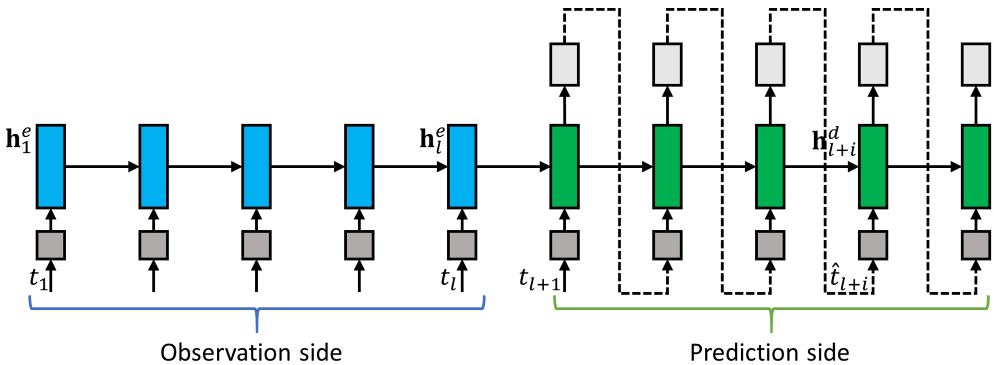

Fig. 2. The motivation of using sequence-to-sequence model for n-step citation prediction task. During training, the sequence-to-sequence model has the advantage of taking into account both the observation side and the prediction side which can improve the accuracy of long-term predictions.

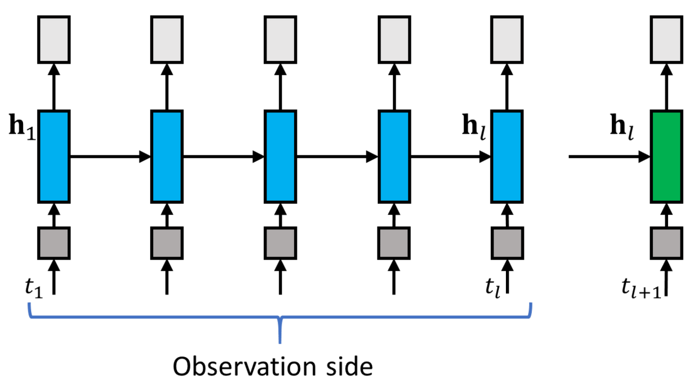

Fig. 3. A typical many-to-many structure for one-step forecasting. The model only learns to capture the temporal dynamics among the observations.

structure which takes into account both the observation side and prediction side during the training, as shown in Figure 2.

When $n = 1$, the first problem is relaxed to the one-step forecasting problem, which is the simpler since learning the temporal point process depends only on the observation side so there will be no error propagation on the prediction side. Typically, a neural point process for this task adopts a many-to-many structure, as shown in Figure 3, where only the temporal dynamics of the observations are captured during training. Since predicting only the next citation does not have much practical value in real-world applications, we focus only on the task of n-step forecasting.

4 MODELS

In this section, we present our proposed model, DMA-Nets. First we show an overview of the design of the proposed framework, as demonstrated in Figure 4. Then we detail the input layer (L1 in Figure 4) and the recurrent representation layer (L2 in Figure 4) used as the base of our model. Next, we propose the hierarchical dynamic attention mechanism (L3 and L4 in Figure 4) which empowers the model to modulate complicated correlations between historical citations and dynamically learn from both the observation side and the prediction side. Finally, we describe the

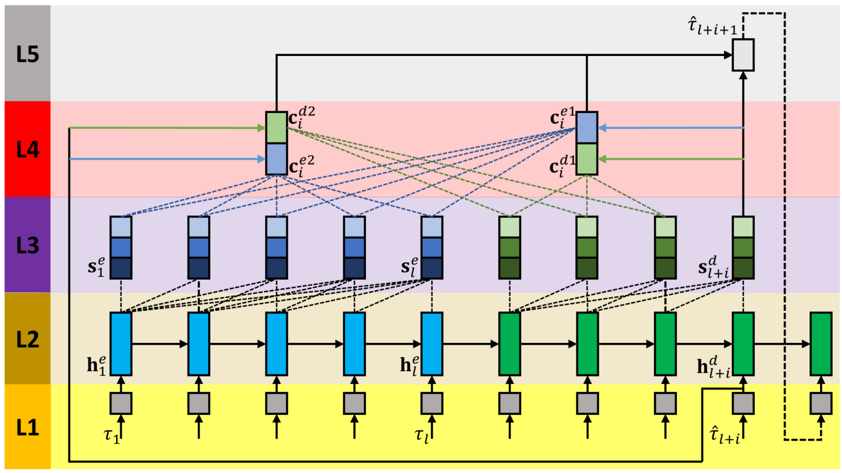

Fig. 4. The architecture of DMA-Nets. L1 is the input layer. L2 is the recurrent representation layer. L3 refers to the local temporal attention (LTA) layer and L4 to the global multi-context temporal attention (GMTA) layer. Together, these comprise the dynamic hierarchical attention layer. L5 is the prediction layer.

temporal prediction layer and the training procedure which unifies the distinct components of the proposed framework.

4.1 Model Architecture

By considering the arrival of a citation as an instant point on the timeline of the citation sequence of a scientific document, we can study the entire citation sequence as a point process that is governed by a hidden conditional intensity function which models the dependencies of arriving citations based on historical citations. The joint density of all citations can be represented as

$$f((\tau_1, m_1), (\tau_2, m_2), \ldots) = \prod_i f\left((\tau_i, m_i) | \ldots, (\tau_{i-2}, m_{i-2}), (\tau_{i-1}, m_{i-1})\right) = \prod_i f\left((\tau_i, m_i) | \mathcal{H}^*\right), \quad (1)$$

where τ_i and m_i, respectively, denote the inter-citation duration and the technology class of ith citation, which are conditioned by the information of all historical citations before the ith citation, denoted by $\mathcal{H}^*$. With this learned joint density, the future arrival of citations can be predicted through a generative process; for example, by estimating the expectation of the conditional density function through numerical integration.

In this article, guided by the Seq2Seq architecture [45], we propose a novel hierarchical dynamic attention neural network in an end-to-end fashion which integrates the task of representing complicated hidden dependencies across historical citations and the task of predicting the technology class and arrival timestamp of the next n citations. Figure 4 presents the overall encoder-decoder architecture of DMA-Nets where the encoder is supplied with the sequence of observed citations $\zeta_e = \{(\tau_1, m_1), (\tau_2, m_2), \ldots, (\tau_l, m_l)\}$ and the decoder aims to recurrently predict the sequence of the next n citations $\zeta_d = \{(\hat{\tau}_{l+1}, \hat{m}_{l+1}), (\hat{\tau}_{l+2}, \hat{m}_{l+2}), \ldots, (\hat{\tau}_{l+n}, \hat{m}_{l+n})\}$.

The model consists of four sublayers: the input layer, the recurrent representation layer, the attention layer, and the prediction layer. The input layer encodes temporal information from the inputs into dense vectors. The recurrent representation layer captures the hidden long/short dependencies of the current citations over all previous citations. The learned representations enter the attention layer which consists of two modules: the local temporal attention layer and the global multi-context temporal attention layer. On top of the recurrent hidden states, the local temporal attention layer compiles the temporal influence between each pair of historical citations and generates intra-encoder states and intra-decoder states. Next, on the decoder side, the global temporal attention layer fuses multiple contexts obtained by attending to different queries on the information embedded by the inner states of both encoder and decoder. Finally, the prediction layer makes the technology category classification and time-aware timestamp prediction for the next n citations. Detailed explanations of each layer follows.

4.2 Seq2Seq Structure for Citation Prediction

In this article, we propose a *year-month-day* embedding method to represent the inter-citation duration τ between subsequent citations into a d_τ-dimension dense vector

$$\mathbf{a}^t = f^t_{emb}(\tau) = \boldsymbol{\tau}^{\mathrm{T}}\mathbf{W}^t, \tag{2}$$

where $\mathbf{W}^t \in \mathbb{R}^{3\times d_\tau}$ is a learnable embedding matrix and $\boldsymbol{\tau} \in \mathbb{R}^3$ is obtained by assembling the discretization of numerical attributes on years, months, and days

$$\text{years} = \left\lfloor \frac{\tau}{365} \right\rfloor, \text{months} = \left\lfloor \frac{\tau - 365 \times \text{years}}{30} \right\rfloor, \text{days} = \tau - 365 \times \text{years} - 30 \times \text{months}.$$

Note here that leap years and the exact number of days in a month are not considered since they do not affect the model performance. For the document class, a look-up table embedding layer is adopted to encode it into a vector in d_m-dimension space:

$$\mathbf{a}^m = f^m_{emb}(\tau). \tag{3}$$

At last, the two embeddings are concatenated together as the input vector to the following recurrent representation layer:

$$\mathbf{a} = f_{emb}(\tau, m) = [\mathbf{a}^t; \mathbf{a}^m] \in \mathbb{R}^{d_\tau + d_m}. \tag{4}$$

Given the observation sequence ζ_e, at each step, the encoder aims to encode and to compile the hidden dependencies across observed historical citations, thus generating a sequence of hidden states $\mathbf{h}^e = \{\mathbf{h}^e_1, \ldots, \mathbf{h}^e_l\}, \mathbf{h}^e_i \in \mathbb{R}^{d_h}$. The calculation of the ith hidden state $\mathbf{h}^e_i$ is defined in Equation (5):

$$\begin{aligned} \mathbf{a}_i &= f_{emb}(\tau_i, m_i), \\ \mathbf{h}^e_i &= g_{rnn}(\mathbf{a}_i, \mathbf{h}^e_{i-1}), \end{aligned} \tag{5}$$

where f_{emb} is the input embedding method defined in Equation (4) which transforms the temporal input τ_i and document class input m_i into a dense vector $\mathbf{a}_i \in \mathbb{R}^{d_\tau + d_m}$, and g_{rnn} is a recurrent unit (e.g., LSTM [19], GRU [8], or vanilla RNN) which captures the dependency structure of the current input over the hidden state at the previous step $\mathbf{h}^e_{i-1}$. Likewise, at each step, the decoder takes as input the prediction from the previous step and predicts the next inter-citation duration as defined in Equation (6):

$$\begin{aligned} \mathbf{a}_{l+i} &= f_{emb}(\hat{\tau}_{l+i}, \hat{m}_{l+i}), \\ \mathbf{h}^d_{l+i} &= g_{rnn}(\mathbf{a}_{l+i}, \mathbf{h}^d_{l+i-1}), \\ \hat{\tau}_{l+i+1} &= p^t(\mathbf{h}^d_{l+i}), \\ \hat{m}_{l+i+1} &= p^m(\mathbf{h}^d_{l+i}), \end{aligned} \tag{6}$$

where p^t and p^m are functions that predict the arrival time and technology category of the next citation based on the current hidden state, respectively. In this work, we use an LSTM recurrent unit because it performs slightly better than GRU and vanillar RNN. And we employ $d_\tau = 32$, $d_m = 32$, and $d_h = 256$.

4.3 Hierarchical Dynamic Attention Layer

Though recurrent neural networks have been successfully used in various time series prediction tasks [10], the fact that the last hidden state holds all the memory of the sequence poses a bottleneck in learning conditional dependencies across a long sequence of temporal points. Furthermore, as a consequence of Equation (1), the model's performance heavily depends on the its ability to approximate the conditional intensity behind the citation sequence. As a result, we propose a hierarchical dynamic attention layer that explores pairwise citation dynamics from both local and global perspectives and from the viewpoint of both observations and predictions.

4.3.1 Local Temporal Attention (LTA) Layer. In this layer, we propose a local temporal attention mechanism to enhance the modulation of conditional dependencies by allowing the model to access and directly attend to previous hidden states. We illustrate the local temporal attention mechanism on the encoder. The decoder follows a similar process. Let $\mathbf{h}_i^e$ be the current hidden state of the encoder and let $\mathcal{H}_i^e = [\mathbf{h}_1^e; \ldots; \mathbf{h}_i^e] \in \mathbb{R}^{d_h \times i}$ be the i previous hidden states available along the time dimension. The local temporal attention mechanism aims to generate a corresponding intra-encoder attentional hidden state $\mathbf{s}_i^e$ for hidden state $\mathbf{h}_i^e$

$$\mathbf{s}_i^e = \mathrm{LTA}(\mathbf{h}_i^e, \mathcal{H}_i^e) + \mathbf{h}_i^e,$$

where the second term is a residual connection to improve the stability of the model. To further enhance the model's flexibility in representing conditional temporal dependencies, we use multiple *heads* [46] to calculate attentional hidden states in different semantic subspaces and concatenate all the results together as the final $\mathbf{s}_i^e$. The calculation for the kth head is defined as

$$\begin{aligned}\mathbf{s}_{i,k}^e &= \mathrm{LTA}_k(\mathbf{W}_k^1\mathbf{h}_i^e, \mathbf{W}_k^2\mathcal{H}_i^e, \mathbf{W}_k^3\mathcal{H}_i^e) \\ &= \mathrm{LTA}_k(\boldsymbol{\hbar}_i^e, \bar{\mathcal{H}}_i^e, \tilde{\mathcal{H}}_i^e) \\ &= \sum_j^i w_{ij}\tilde{\mathcal{H}}_{i,j}^e = \sum_j^i \frac{\exp(e_{ij})}{\sum_k^i \exp(e_{ik})}\tilde{\mathcal{H}}_{i,j}^e,\end{aligned} \tag{7}$$

where $\mathbf{s}_{i,k}^e \in \mathbb{R}^{d_q}$ is an attentional hidden state for head k, and $\mathbf{W}_k^1$, $\mathbf{W}_k^2$, $\mathbf{W}_k^3$ are three learnable $d_q \times d_h$ matrices which project $\mathbf{h}_i^e$, $\mathcal{H}_i^e$ into three different subspaces $\boldsymbol{\hbar}_i^e \in \mathbb{R}^{d_q}$, $\bar{\mathcal{H}}_i^e \in \mathbb{R}^{d_q \times i}$, and $\tilde{\mathcal{H}}_i^e \in \mathbb{R}^{d_q \times i}$, w_{ij} is normalized e_{ij} measuring the amount of attention $\boldsymbol{\hbar}_i^e$ should pay to $\tilde{\mathcal{H}}_{i,j}^e$ (the jth column of $\tilde{\mathcal{H}}_i^e$), and e_{ij} is calculated by the following score function

$$e_{ij} = \frac{\left(\mathrm{Sigmoid}(\mathbf{W}^e\boldsymbol{\hbar}_i^e)\right)^{\mathrm{T}}\tilde{\mathcal{H}}_{i,j}^e}{\sqrt{d_q}},$$

where $\mathbf{W}^e \in \mathbb{R}^{d_q \times d_q}$ is a learnable square matrix. Different from the vanilla dot-product score function, a non-linear projection of $\boldsymbol{\hbar}_i^e$ is used to avoid biased attention towards its neighbor hidden states (e.g., $\boldsymbol{\hbar}_i^e, \boldsymbol{\hbar}_{i-1}^e$). Also, the score is scaled to avoid values of large magnitude [46]. Finally, $\mathbf{s}_i^e$ is obtained by concatenating $\mathbf{s}_{i,k}^e$ for each head:

$$\mathbf{s}_i^e = \mathrm{LTA}(\mathbf{h}_i^e, \mathcal{H}_i^e) + \mathbf{h}_i^e = \mathrm{concat}(\mathbf{s}_{i,1}^e, \ldots, \mathbf{s}_{i,h}^e)\mathbf{W}^o + \mathbf{h}_i^e, \tag{8}$$

where h is the number of heads used, $\mathbf{W}^o \in \mathbb{R}^{hd_q \times d_h}$ transforms the hd_q-dimension result back to d_h-dimension space. Likewise, the intra-decoder attentional hidden state $\mathbf{s}_d^{\langle i+l \rangle}$ can be obtained by

$$\mathbf{s}_{l+i}^d = \mathrm{LTA}(\mathbf{h}_{l+i}^d, \mathcal{H}_{l+i}^d) + \mathbf{h}_{l+i}^d, \tag{9}$$

where $\mathcal{H}^d_{l+i} = [\mathbf{h}^d_{l+1}; \ldots; \mathbf{h}^d_{l+i}]$ refers to all currently available hidden states on the decoder. For this article, we employ $h = 4$ and $d_q = 64$.

4.3.2 Global Multi-Context Temporal Attention (GMTA) Layer. On top of the local temporal attention layer, we propose a global multi-context temporal attention mechanism with the following considerations in mind. First, the approach should allow the model to continue examining conditional temporal dependencies in the decoder phase. That is, the proposed approach should consider the attentional hidden states on both sides, unlike the traditional attention strategy [32] which attends only to encoder states. Second, the approach should let the model dynamically determine the combination of information from the encoder and decoder sides. Third, instead of learning attention weights based only on the state value, we argue that the temporal pattern of the temporal point process in the input should also be a decisive factor.

Here, we illustrate the computation process of attentional contexts on the encoder side. At the ith step of the decoder, let $\mathbf{s}^d_{l+i}$ be the decoder's current attentional hidden state. Let $\mathrm{S}^e = [\mathbf{s}^e_1; \ldots; \mathbf{s}^e_l] \in \mathbb{R}^{d_h \times l}$ be the encoder's l attentional hidden states and $\mathcal{A}^e = [\mathbf{a}^e_1; \ldots; \mathbf{a}^e_l]$ be the encoder's l inputs. Two contexts, $\mathbf{c}^{e1}_i$ and $\mathbf{c}^{e2}_i$, are calculated. Again, we employ the multi-head strategy to calculate $\mathbf{c}^{e1}_i$ and $\mathbf{c}^{e2}_i$ in different semantic subspaces and use the concatenation for the final context. For the kth head, both contexts are a weighted sum of projected S^e. The difference is that for $\mathbf{c}^{e1}_{i,k}$ the attention weight e^{e1}_{ij} depends on the value of the attentional hidden states $\mathbf{s}^d_{l+i}$ and $\mathbf{s}^e_j$ while for $\mathbf{c}^{e2}_{i,k}$ the temporal pattern inputs $\mathbf{a}^d_{l+i}$ and $\mathbf{a}^e_j$ determine the attention weights e^{e2}_{ij}, that is

$$e^{e1}_{ij} = \frac{\left(\mathbf{Q}^{e1}_k \mathbf{s}^d_{l+i}\right)^{\mathrm{T}} \mathbf{V}^{e1}_k \mathbf{s}^e_j}{\sqrt{d_{c1}}}, \quad e^{e2}_{ij} = \frac{\left(\mathbf{Q}^{e2}_k \mathbf{a}^d_{l+i}\right)^{\mathrm{T}} \mathbf{V}^{e2}_k \mathbf{a}^e_j}{\sqrt{d_{c2}}}, \tag{10}$$

where $\mathbf{Q}^{e1}_k, \mathbf{V}^{e1}_k \in \mathbb{R}^{d_{c1} \times d_h}$ and $\mathbf{Q}^{e2}_k, \mathbf{V}^{e2}_k \in \mathbb{R}^{d_{c2} \times d_h}$ are learnable matrices for linear projection. Then we have $\mathbf{c}^{e1}_{i,k}$ and $\mathbf{c}^{e2}_{i,k}$ calculated as:

$$\mathbf{c}^{e1}_{i,k} = \sum_j^l \frac{\exp(e^{e1}_{ij})}{\sum_k^l \exp(e^{e1}_{ik})} \mathbf{U}^{e1}_k \mathbf{s}^e_j, \quad \mathbf{c}^{e2}_{i,k} = \sum_j^l \frac{\exp(e^{e2}_{ij})}{\sum_k^l \exp(e^{e2}_{ik})} \mathbf{U}^{e2}_k \mathbf{s}^e_j, \tag{11}$$

where $\mathbf{U}^{e1}_k \in \mathbb{R}^{d_{c1} \times d_h}$ and $\mathbf{U}^{e2}_k \in \mathbb{R}^{d_{c2} \times d_h}$. Finally, contexts $\mathbf{c}^{e1}_i$ and $\mathbf{c}^{e2}_i$ are obtained by concatenating the results of all heads:

$$\begin{aligned}
\mathbf{c}^{e1}_i &= \mathrm{GMTA}^{s}(\mathbf{s}^d_{l+i}, \mathrm{S}^e) = \mathrm{concat}(\mathbf{c}^{e1}_{i,1}, \ldots, \mathbf{c}^{e2}_{i,m1}) \mathbf{W}^{e1}, \\
\mathbf{c}^{e2}_i &= \mathrm{GMTA}^{\tau}(\mathbf{a}^d_{l+i}, \mathcal{A}^e, \mathrm{S}^e) = \mathrm{concat}(\mathbf{c}^{e2}_{i,1}, \ldots, \mathbf{c}^{e2}_{i,m2}) \mathbf{W}^{e2},
\end{aligned}$$

where m_1 and m_2 are the number of heads to use for $\mathbf{c}^{e1}_i$ and $\mathbf{c}^{e2}_i$, respectively, and both $\mathbf{W}^{e1} \in \mathbb{R}^{hd_{c1} \times d_h}$ and $\mathbf{W}^{e2} \in \mathbb{R}^{hd_{c2} \times d_h}$ are learnable projection matrices. Likewise, let S^d_{l+i} represent all of the decoder's previous attentional hidden states and $\mathcal{A}^d_{l+i}$ all the decoder's previous inputs. At the ith step, the decoder context $\mathbf{c}^{d1}_i$ and $\mathbf{c}^{d2}_i$ can be calculated by $\mathrm{GMTA}^{s}(\mathbf{s}^d_{l+i}, \mathrm{S}^d_{l+i})$ and $\mathrm{GMTA}^{\tau}(\mathbf{a}^d_{l+i}, \mathcal{A}^d_{l+i}, \mathrm{S}^d_{l+i})$, respectively. In this work we employ $m_1 = 4$, $d_{c1} = 64$, $m_2 = 2$ and $d_{c2} = 32$.

While the encoder context and the decoder context dynamics integrate the historical representations unfolding along each side's timeline separately, it's challenging to fuse the contexts from both sides since the importance of the observation and prediction side could be very dynamic and depending on the dataset. Thus, we propose a gate mechanism that is able to automatically determine the balance between the context of the two sides based on the difference of two sides:

$$\begin{aligned} gate_1 &= \sigma\left(\mathbf{W}_{g1}\left|(\mathbf{c}_i^{d1} - \mathbf{c}_i^{e1})\right|\right), \\ gate_2 &= \sigma\left(\mathbf{W}_{g2}\left|(\mathbf{c}_i^{d2} - \mathbf{c}_i^{e2})\right|\right), \end{aligned} \tag{12}$$

where W_{g1} and W_{g2} are both learnable parameters, and σ is the Sigmoid function which is adopted to enable model to better fuse the contexts via the feature dimension. At last, the context calculated on the observation and prediction side could be dynamically fused as:

$$\begin{aligned} \mathbf{c}_i^1 &= \mathbf{c}_i^{e1} \odot gate_1 + (1 - gate_1) \odot \mathbf{c}_i^{d1}, \\ \mathbf{c}_i^2 &= \mathbf{c}_i^{e2} \odot gate_2 + (1 - gate_2) \odot \mathbf{c}_i^{d2}. \end{aligned} \tag{13}$$

4.4 Prediction Layer

The citation sequence has an implicit constraint that future citations always come after the most recent citation, i.e., $t_{i+1} \geq t_i$. Some previous work [52] ignores this constraint. Considering that $t_{i+1} = t_i + \tau_{i+1}$, we have $\tau_{i+1} \geq 0$; that is, the predicted inter-citation duration should always be non-negative. With this in mind, we design the temporal prediction layer in which a blend function combines encoder contexts, decoder contexts, and the current hidden attentional state of the decoder to generate the fused context $\mathbf{c}_i \in \mathbb{R}^{d_h}$, which is then used for prediction:

$$\begin{aligned} \mathbf{c}_i &= \text{concat}\left(\mathbf{c}_i^1, \mathbf{c}_i^2, \mathbf{s}_i^d, \mathbf{h}_i^d\right)\mathbf{W}^c, \\ \hat{\tau}_{l+i} &= \text{Softplus}(\mathbf{W}^\tau \mathbf{c}_i), \\ \hat{m}_{l+i} &= \text{Softmax}(\mathbf{W}^m \mathbf{c}_i), \end{aligned} \tag{14}$$

where $\mathbf{W}^c$, $\mathbf{W}^\tau$, and $\mathbf{W}^m$ are all learnable parameters and $\odot$ is the Hadamard product. Note that the prediction is constrained by the Softplus function to enforce the non-negative requirement.

4.5 Parameter Learning

The loss for the timestamp forecasting consists of two parts: (1) alignment between the predicted and the ground truth arrival time and (2) the alignment between the predicted and the ground truth interval. First, intuitively, the most straightforward loss function to optimize the model is to calculate the difference of alignment of the predicted and the ground truth timestamp:

$$\text{loss}_{t1} = \sum_{i=l+1}^{l+n} d\left(\hat{t}_i, t_i\right) = \sum_{i=l+1}^{l+n} \left|\hat{t}_i - t_i\right|, \tag{15}$$

where the $(l+1)$th citation is the first citation that arrives after the observation window, the $(l+n)$th citation is the last citation received, and d is a function calculating the difference between $\hat{t}_i$ and t_i. In this article, we use absolute difference. For inter-citation sequence, the loss function can be derived from Equation (15):

$$\begin{aligned} \text{loss}_{t1} &= \sum_{i=l+1}^{l+n} d\left(t_l + \sum_{j=l+1}^{i} \hat{\tau}_j, t_l + \sum_{j=l+1}^{i} \tau_j\right) \\ &= \sum_{i=l+1}^{l+n} \left|\sum_{j=l+1}^{i} \left(\hat{\tau}_j - \tau_j\right)\right|. \end{aligned} \tag{16}$$

We argue that, for time sequence predictions, the error of earlier predictions will be propagated to later predictions. To alleviate this problem, we further assign weights which decay along the decoder's sequence of predictions:

$$\text{loss}_{t1}^{\text{weighted}} = \sum_{i=l+1}^{n+l} w_i \left| \sum_{j=l+1}^{i} \left(\hat{\tau}_j - \tau_j \right) \right|, \tag{17}$$

where w_i follows an exponential decay function [15]:

$$w_i = \exp \left\{ -\theta(i - l - 1) \right\}, \theta > 0.$$

Optimizing based only on the alignment of the citation-wise arrival time would lead the model to ignore the time window with a high density of arriving citations when the inter-arrival period is relatively small. As a result, we propose another loss function to encourage the model to align the inter-arrival time directly:

$$\text{loss}_{t2} = \sum_{i=l+1}^{l+n} dist\left(\hat{\tau}_i, \tau_i\right), \tag{18}$$

where $dist(\cdot, \cdot)$ is the distance function and we adopt the Euclidean distance.

At last, the total loss is the sum of the timestamp loss (Equations (17) and (18)) and the cross-entropy loss for document category prediction:

$$\text{loss} = \text{loss}_{t1}^{\text{weighted}} + \text{loss}_{t2} - \sum_{i=l+1}^{l+n} \log(m_i). \tag{19}$$

For regularization, we use dropout to the output of each sublayer with a dropout rate of 0.1. For optimization, we adopted the ADAM [24] optimizer for training with learning rate set to 0.0001 and weight decay of 0.0001.

5 EXPERIMENTS

We compare our DMA-Nets experimentally to state-of-the-art methods for modeling temporal point processes on two large real-world datasets collected from the United States Patent and Trademark Office (USPTO) and the Microsoft Academic Graph (MAG).

5.1 Dataset Description and Experiment Setup

Dataset. USPTO is a patent database documenting 6,819,362 U.S. patents. According to patent law, new inventions must cite prior arts and differentiate their innovations from them. For each patent, we construct a citation chain using timestamps of related patents from the database. In order to make fair comparisons, we use the dataset available in [22] but trim citation chains longer than 100 to conform with limited video memory. Also, long citation chains are relatively rare in practice and unbalanced sequence lengths lead to unnecessary computation. In summary, the dataset consists of 15,000 sequences of which 3,000 sequences are the test set and the remaining 12,000 sequences are split 80/20 for training/validation. MAG [44] is a paper database maintained by Microsoft containing information on around 166,192,182 publications including conference papers, journal papers, and books. For each paper, we construct a citation chain using its publish date from the database. We likewise remove papers with chains shorter than 20 and trim chains longer than 100, and then sample 15,000 sequences with 3,000 for the test set and the rest split 80/20 for training/validation. The entire MAG database is publicly accessible on Zenodo.[2]

[2]https://zenodo.org/record/2593154#.XJmKTaQpBhG

Metrics. Following similar procedures in [10], [22], [48], and [53], we use **mean absolute error (MAE)**, **root mean squared error (RMSE)**, and **accuracy (ACC)** as evaluation metrics for citation time predictions:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=l+1}^{l+n}\left|\hat{t}_i - t_i\right|,$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=l+1}^{l+n}\left(\hat{t}_i - t_i\right)^2}.$$

Compared Baselines. We compare DMA-Nets with state-of-the-art neural point process baselines, including two intensity-based models and four end-to-end based models. For all baselines, we adopted their official implementations and utilized the default hyperparameter settings provided in the implementations. This ensures a fair and consistent comparison among the different models.

— *RMTPP* [10]. RMTPP uses recurrent units to learn the intensity function for general point process analysis and is able to predict the next point in an event sequence. RMTPP is a one-step model, by directing the ith output to the $(i + 1)$th input, this model becomes a citation prediction generator.
— *CYAN-RNN* **[48]**. CYAN-RNN uses GRU-based recurrent units and attention mechanisms to learn the intensity function for a general information resharing process and can forecast the arrival of next resharing behavior. Similar to RMTPP, CYAN-RNN is a one-step model, by directing the ith output to the $(i + 1)$th input, this model becomes a citation prediction generator.
— *RPP* [53]. RPP is similar to RMTPP, but it uses a fully connected layer to map the embedded hidden state directly to time predictions. Also, it uses Gaussian penalty to calculate time prediction loss. RPP is a one-step model. By directing the ith prediction to the $(i+1)$th input, we use this model as a citation prediction generator.
— *LT-CCP* [59]. LT-CCP uses LSTM-based recurrent units to modulate the temporal dependency across citations. It's worth noting that the LT-CCP is an n-step model.
— *GRU-CPM* [49]. Like LT-CCP, GRU-CPM is also an n-step forecasting model based on a recurrent neural network structure but with GRU units. GRU-CPM is focused on a small set of papers on Computer Science and is originally designed to accept hand-crafted features extracted from author and literature content. We utilize the temporal and document class instead since the literature and author information is unavailable in our dataset, which covers many different disciplines.
— $\mathrm{S2S}_d$ [32]. $\mathrm{S2S}_d$ represents a Seq2Seq model with the traditional static attention mechanism. We used a dot-product score function in the experiments. Seq2seq is an n-step model.
— *PC-RNN* [22]. PC-RNN is an end-to-end neural point process model for patent citation forecasting which is able to integrate multiple observation sequences and have a static attention mechanism equipped on the prediction side. On the USPTO dataset, we used three sequences of patent citations, assignee citations, and inventor citations. On the MAG dataset, the observation side has only paper citation sequences available. PC-RNN is an n-step model.

5.2 Performance Comparison

By restricting the length of the observation window to a different degree, we evaluate all models' performance on citation forecasting. In the first setting, we adopted a dynamic observation window by separately using 10%, 30%, 50%, and 80% of the citation sequence as observations. We call this setting the ratio observation setting. In the second setting, we tested all models with a relatively

Table 1. Performance Evaluation of Our Method and Peer Methods on USPTO Dataset under the Settings of 10% and 30% as Observations

Model	10% as observations					
	Main-category			Sub-category		
	MAE	RMSE	ACC	MAE	RMSE	ACC
RMTPP‡	349.95	477.19	0.576	344.34	466.87	0.147
CYAN-RNN‡	288.83	391.91	0.832	342.55	494.87	**0.623**
RPP‡	318.13	457.98	**0.838**	464.88	673.89	0.514
LT-CCP	344.86	469.32	0.458	344.94	469.53	0.146
GRU-CPM	344.06	467.60	0.458	344.74	469.31	0.146
S2S	222.42	309.58	0.511	221.43	308.30	0.382
$S2S_d$	210.49	289.60	0.753	208.53	287.85	0.450
PC-RNN	**197.37**	**278.10**	0.779	**195.17**	**281.59**	0.462
DMA-Nets	202.01	282.52	0.823	201.08	278.75	0.537
Model	30% as observations					
	Main-category			Sub-category		
	MAE	RMSE	ACC	MAE	RMSE	ACC
RMTPP‡	258.96	370.65	0.586	236.15	332.64	0.247
CYAN-RNN‡	191.96	283.46	0.843	212.62	301.52	0.622
RPP‡	309.07	429.71	0.684	393.05	603.28	0.525
LT-CCP	191.42	288.61	0.457	207.32	311.66	0.154
GRU-CPM	236.19	335.32	0.457	235.80	334.23	0.145
S2S	148.83	223.14	0.587	145.42	219.11	0.481
$S2S_d$	143.98	215.89	0.756	139.33	206.42	0.560
PC-RNN	132.27	199.28	0.774	129.81	198.39	**0.643**
DMA-Nets	**129.67**	**194.23**	**0.847**	**128.14**	**192.35**	0.620

Timestamp predictions are evaluated using MAE and RMSE. Patent category predictions are evaluated using ACC. ‡ means this model is a one-step forecasting approach.

static observation window by adopting the first 10, 20, 30, and 40 citations in the sequence as the observations. We call this setting the fixed-length observation setting. In the experiment, S2S and $S2S_d$ use the same hyperparameter setting as our proposed model, including hidden state size, learning rate, and learning steps. The other models used their official settings and implementations. All results are reported in Tables 1–6.

5.2.1 DMA-Nets Versus One-Step Forecasting Models. Our model consistently outperforms RMTPP, CYAN-RNN, and RPP for timestamp prediction in all experiments. For instance, under the ratio observation setting, on the patent dataset, for example, against the best of these three models, DMA-Nets can obtain 22.36%, 30.45%, 31.10%, and 30.06% gains in MAE on 80%, 50%, 30%, and 10% observation windows, respectively. The performance of one-step forecasting models drops significantly as the observation window shrinks. This observation demonstrates our assumption stated in Section 3, that is, the model should consider prediction side information and errors to improve overall accuracy. Our proposed model has a significantly better performance by considering both the citation dynamics on the observation and prediction sides. In the experiments, we also observed intensity-based methods such as RPP and RMTPP suffer from high variance during training; that is, the model performance on the validation dataset decreases even though training

Table 2. Performance Evaluation of Our Method and Peer Methods on USPTO Dataset under the Settings of 50% and 80% as Observations

Model	50% as observations					
	Main-category			Sub-category		
	MAE	RMSE	ACC	MAE	RMSE	ACC
RMTPP‡	182.57	271.25	0.587	174.15	252.67	0.243
CYAN-RNN‡	142.61	221.63	0.842	150.84	229.41	0.634
RPP‡	234.23	351.27	0.723	195.57	280.44	0.575
LT-CCP	127.65	206.88	0.540	125.21	205.12	0.265
GRU-CPM	169.66	251.95	0.455	167.96	249.60	0.148
S2S	110.45	176.99	0.682	108.01	173.52	0.579
$S2S_d$	106.93	165.70	0.737	102.27	161.10	0.601
PC-RNN	**98.83**	**156.60**	0.758	**97.31**	**153.38**	**0.659**
DMA-Nets	99.19	156.93	**0.853**	97.97	153.57	0.637
Model	80% as observations					
	Main-category			Sub-category		
	MAE	RMSE	ACC	MAE	RMSE	ACC
RMTPP‡	83.02	138.22	0.454	84.15	135.31	0.246
CYAN-RNN‡	71.86	124.23	0.844	79.95	131.63	0.639
RPP‡	109.36	175.82	0.821	118.94	183.15	0.664
LT-CCP	66.64	119.48	0.792	66.28	119.73	0.456
GRU-CPM	81.82	135.42	0.455	81.55	134.58	0.145
S2S	62.00	108.96	0.742	61.45	109.20	0.631
$S2S_d$	60.81	105.75	0.819	58.84	102.73	**0.669**
PC-RNN	56.89	100.38	0.812	56.31	97.14	**0.669**
DMA-Nets	**55.79**	**97.72**	**0.857**	**55.23**	**96.44**	0.668

Timestamp predictions are evaluated using MAE and RMSE. Patent category predictions are evaluated using ACC. ‡ means this model is a one-step forecasting approach.

loss is decreasing. We argue this is because intensity-based methods use maximum likelihood estimation, which is more susceptible to overfitting by using only information from the observation side. In terms of document class prediction, we observed that our model significantly outperforms RMTPP and RPP and is competitive with CYAN-RNN. We argue this is because both our model and CYAN-RNN are empowered by the attention mechanism, which allows the model to look back at previous document categories during the prediction process.

5.2.2 DMA-Nets Versus n-Step Forecasting Methods. Our model generally performs better than S2S and $S2S_d$. For instance, on the patent dataset, against the best of these two models, our model can improve performance on MAE at least by 3.57% in all tests. Also, on the patent dataset, our model's performance is competitive with PC-RNN on the citation prediction task, though PC-RNN uses three information sequences on the observation side. We argue that this boost is attributed to both the local temporal and the global multi-context temporal attention layers. Our proposed local temporal attention layer allows both the encoder and the decoder to look back along the temporal dimension at each step and automatically attend to important states in each sequence. First, this alleviates the burden on the recurrent unit and significantly improves the model's flexibility to modulate complicated dependency structures in the observation and prediction sequences. Second,

Table 3. Performance Evaluation of Our Method and Peer Methods on USPTO Dataset under the Settings of First 10 and 30 as Observations

Model	First 10 as observations					
	Main-category			Sub-category		
	MAE	RMSE	ACC	MAE	RMSE	ACC
RMTPP‡	285.30	394.41	0.587	241.52	338.00	0.145
CYAN-RNN‡	328.68	463.56	0.845	328.68	508.54	0.662
RPP‡	241.58	338.13	0.453	241.43	337.88	0.138
LT-CCP	236.19	330.94	0.453	202.93	295.31	0.178
GRU-CPM	236.51	331.48	0.453	236.86	331.93	0.145
S2S	177.54	256.84	0.750	169.42	247.45	0.449
$S2S_d$	175.52	254.72	0.773	168.31	241.65	0.501
PC-RNN	**166.90**	**242.42**	**0.833**	**164.69**	**238.32**	0.548
DMA-Nets	173.59	251.70	0.813	165.67	239.66	**0.624**
Model	First 30 as observations					
	Main-category			Sub-category		
	MAE	RMSE	ACC	MAE	RMSE	ACC
RMTPP‡	172.00	253.19	0.573	168.75	245.93	0.256
CYAN-RNN‡	268.41	386.39	0.825	283.26	431.61	0.637
RPP‡	163.69	237.87	0.441	163.50	237.66	0.141
LT-CCP	130.46	201.40	0.632	126.57	198.00	0.274
GRU-CPM	163.54	237.97	0.441	163.28	237.69	0.152
S2S	124.88	194.62	0.765	120.00	186.48	0.479
$S2S_d$	121.66	189.82	0.829	117.42	179.98	0.563
PC-RNN	118.29	**180.84**	**0.843**	115.55	179.39	0.645
DMA-Nets	**118.00**	181.17	0.821	**111.32**	**175.47**	**0.649**

Timestamp predictions are evaluated using MAE and RMSE. Patent category predictions are evaluated using ACC. ‡ means this model is a one-step forecasting approach.

the global multi-context temporal attention layer allows the model to improve the learned dependency structure even in the prediction phase. Furthermore, it empowers the model to automatically combine learned information from both encoder and decoder for better prediction results. In contrast, S2S relies solely on the last hidden state to carry the entire sequence's information while $S2S_d$ and PC-RNN only consider static attention on the hidden states of the encoder side.

5.3 Ablation Study

Global Multi-Context Temporal Attention (GMTA) Layer Analysis. We first analyze the contributions of the global multi-context temporal attention layer (GMTA). In this ablation test, we remove the GMTA layer from DMA-Nets and create one variant, named DMA-Nets$_g$. For DMA-Nets$_g$, at each step of the decoder, we drop the calculation of the encoder's contexts $\mathbf{c}_i^{e1}$ and $\mathbf{c}_i^{e2}$ and the decoder's contexts $\mathbf{c}_i^{d1}$ and $\mathbf{c}_i^{d2}$ (L4 in Figure 4) and instead use only the current attentional hidden state $\mathbf{s}_i^d$ as the input for the prediction layer. Consequently, the calculation of the encoder's attentional hidden states $[\mathbf{s}_1^e; \ldots; \mathbf{s}_l^e]$ is also removed. In this variant, decoder's states $\mathbf{h}_i^d$ and $\mathbf{s}_i^d$ carry the burden of holding information of previous states. The performance of DMA-Nets$_g$ is reported in Table 7. As expected, DMA-Nets outperforms DMA-Nets$_g$. Intuitively, the GMTA layer

Table 4. Performance Evaluation of Our Method and Peer Methods on USPTO Dataset under the Settings of First 50 and 80 as Observations

Model	First 50 as observations					
	Main-category			Sub-category		
	MAE	RMSE	ACC	MAE	RMSE	ACC
RMTPP‡	146.46	221.23	0.677	129.91	245.93	0.256
CYAN-RNN‡	225.41	322.21	**0.849**	240.36	357.52	0.653
RPP‡	185.60	272.70	0.423	193.29	283.87	0.240
LT-CCP	101.26	161.81	0.559	98.86	158.59	0.270
GRU-CPM	129.72	191.85	0.423	129.76	192.03	0.168
S2S	100.57	158.79	0.777	98.53	154.83	0.502
$S2S_d$	97.38	155.22	0.836	93.80	148.09	0.589
PC-RNN	**93.28**	**148.22**	0.845	92.77	147.89	0.666
DMA-Nets	94.15	150.34	0.818	**90.90**	**145.84**	**0.667**
Model	**First 80 as observations**					
	Main-category			Sub-category		
	MAE	RMSE	ACC	MAE	RMSE	ACC
RMTPP‡	129.88	200.79	0.560	112.77	167.29	0.274
CYAN-RNN‡	200.06	287.11	0.833	217.73	333.80	0.661
RPP‡	180.09	261.93	0.404	183.46	266.61	0.337
LT-CPP	82.52	135.47	0.724	81.98	134.59	0.344
GRU-CPM	106.97	162.82	0.404	106.54	162.48	0.176
S2S	83.76	135.14	0.798	80.14	129.61	0.523
$S2S_d$	80.33	131.38	0.836	79.13	129.10	0.570
PC-RNN	76.98	123.46	**0.847**	75.70	124.91	0.665
DMA-Nets	**75.72**	**123.27**	0.815	**72.56**	**120.30**	**0.669**

Timestamp predictions are evaluated using MAE and RMSE. Patent category predictions are evaluated using ACC. ‡ means this model is a one-step forecasting approach.

is most beneficial in cases where the model relies more on observations. This is because the GMTA layer provides a global view of citation sequences and captures the dynamics on both the observation side and the prediction side. When the GMTA layer is missing, the prediction side dynamics can be carried by both the recurrent unit and the local attentional state but the historical observations can be encoded only by the RNN backbone. On the USPTO dataset, we observed that the performance gain brought by the GMTA layer is most significant when 80% of sequence used as observations. On the MAG dataset, GMTA layer is most beneficial when 50% of sequence used as observations.

Local Temporal Attention (LTA) Layer Analysis. Next, we analyze the contributions of the local temporal attention layer. We created an ablation named DMA-Nets$_l$ by removing the local temporal attention layer from DMA-Nets. As a result, instead of the attentional hidden states $\{\mathbf{s}_i^e\}$ and $\{\mathbf{s}_i^d\}$, we used their corresponding RNN hidden states $\{\mathbf{h}_i^e\}$ and $\{\mathbf{h}_i^d\}$ as the input for the subsequent **global multi-context temporal attention (GMTA)** layer. The performance of DMA-Nets$_l$ is also reported in Table 7. The fully fledged DMA-Nets outperforms DMA-Nets$_l$ on both datasets, indicating that the LTA layer improves model performance. Also, DMA-Nets$_l$ outperforms both DMA-Nets$_{g1}$ and DMA-Nets$_{g2}$ on both datasets. This further demonstrates that the GTMA layer can modulate temporal dynamics globally and therefore achieve better model accuracy.

Table 5. Performance Evaluation of our Method (DMA-Nets) and Peer Methods on MAG Dataset under the Ratio Observation Setting

Model	10% As Observations			30% As Observations		
	MAE	RMSE	ACC	MAE	RMSE	ACC
RMTPP‡	182.59	238.66	0.494	85.83	118.57	0.327
CYAN-RNN‡	101.06	228.74	0.633	71.85	98.50	0.639
RPP‡	287.61	387.44	0.563	155.23	234.09	0.603
LT-CCP	57.12	73.95	0.212	48.47	65.20	0.473
GRU-CPM	115.51	154.81	0.205	85.60	117.01	0.206
S2S	72.31	94.65	0.242	59.81	78.71	**0.649**
S2S$_d$	55.82	72.55	0.499	47.01	63.16	0.547
PC-RNN	55.45	73.09	0.600	56.67	73.72	0.617
DMA-Nets	**54.32**	**71.43**	**0.619**	**41.37**	**56.40**	0.640
Model	50% As Observations			80% As Observations		
	MAE	RMSE	ACC	MAE	RMSE	ACC
RMTPP‡	56.18	85.05	0.207	26.23	38.53	0.292
CYAN-RNN‡	39.83	56.73	**0.651**	20.34	30.73	0.652
RPP‡	76.55	107.75	0.609	33.14	48.29	0.666
LT-CCP	32.39	46.84	0.612	**18.00**	26.65	0.664
GRU-CPM	52.38	74.19	0.300	23.84	33.99	0.416
S2S	41.55	57.48	0.640	21.67	30.49	0.667
S2S$_d$	32.41	46.07	0.590	**17.59**	25.97	0.632
PC-RNN	38.40	54.28	0.622	20.99	30.45	**0.672**
DMA-Nets	**28.23**	**40.58**	**0.651**	18.00	**24.19**	0.669

Timestamp predictions in days are evaluated using MAE and RMSE and document category predictions are evaluated using accuracy. ‡ means this model is a one-step forecasting approach.

5.4 Hyper-Parameter Analysis

We investigate the sensitivity of d_h, d_{emb}, h, m_1, m_2, and dropout rate and report the performance of DMA-Nets on the 80% observation setting. The results are shown in Table 8. We observe that, in general, DMA-Nets is robust to different hyper-parameter settings. Reducing the model size (d_h, d_{emb}) will slightly decrease the model's performance on the patent dataset. We further observe that using too many or too few heads (h, m_1, and m_2) will have a negative impact on the model's quality.

5.5 Time Complexity Analysis

In comparison to the vanilla recurrent neural network-based baselines, our proposed method exhibits higher computational demands. However, it still maintains a quadratic time complexity, ensuring efficient data processing. Specifically, the time complexity of our method can be expressed as:

$$O\left(N * d_h^2 + N * d_h^2 + N * d_h^2\right) \approx O\left(N * d_h^2\right),$$

where N is the length of the citation sequence. The first term corresponds to the computational complexity of the LSTM layer, when the hidden state size, denoted as d_h, surpasses the combined input size of $d_m + d_\tau$. The second and third terms represent the computational complexity of the local and global attention layers, respectively.

Table 6. Performance Evaluation of Our Method (DMA-Nets) and Peer Methods on MAG Dataset under the Fix-Length Observation Setting

Model	First 10 As Observations			First 20 As Observations		
	MAE	RMSE	ACC	MAE	RMSE	ACC
RMTPP‡	147.51	206.66	0.590	106.17	147.23	0.611
CYAN-RNN‡	101.66	159.28	**0.660**	85.16	135.16	0.664
RPP‡	224.17	315.02	0.326	163.69	250.56	0.374
LT-CCP	58.87	78.49	0.562	41.80	59.67	0.572
GRU-CPM	83.80	115.01	0.206	54.08	77.66	0.205
S2S	59.20	78.39	0.597	43.16	61.84	0.629
$S2S_d$	58.65	78.06	0.620	42.54	61.04	0.649
PC-RNN	58.58	78.61	0.628	42.37	62.65	0.664
DMA-Nets	**57.33**	**77.30**	0.652	**41.30**	**59.01**	**0.665**
Model	**First 30 As Observations**			**First 40 As Observations**		
	MAE	RMSE	ACC	MAE	RMSE	ACC
RMTPP‡	84.26	117.32	0.604	75.11	103.77	0.623
CYAN-RNN‡	75.85	120.34	0.666	56.81	88.94	0.673
RPP‡	108.85	157.54	0.487	88.44	135.71	0.537
LT-CCP	35.09	51.34	0.627	30.04	44.07	0.632
GRU-CPM	44.58	63.62	0.291	35.82	50.65	0.432
S2S	36.81	54.72	0.629	31.95	47.04	0.656
$S2S_d$	35.94	52.71	0.658	30.01	44.03	0.670
PC-RNN	36.17	52.99	0.667	29.54	43.49	0.668
DMA-Nets	**34.43**	**49.61**	**0.675**	**28.31**	**41.16**	**0.679**

Timestamp predictions in days are evaluated using MAE and RMSE and document category predictions are evaluated using accuracy. ‡ means this model is a one-step forecasting approach.

Table 7. Performance Evaluation of Variants of DMA-Nets

USPTO									
Model	80% As Observations			50% As Observations			30% As Observations		
	MAE	RMSE	ACC	MAE	RMSE	ACC	MAE	RMSE	ACC
DMA-Nets_g	58.91	100.31	0.850	103.19	162.47	0.843	133.76	198.67	0.838
DMA-Nets_l	57.54	99.18	0.857	102.52	160.49	0.850	134.07	198.49	0.847
DMA-Nets	55.79	97.72	0.857	99.19	156.93	0.853	129.67	194.23	0.847
MAG									
Model	80% As Observations			50% As Observations			30% As Observations		
	MAE	RMSE	ACC	MAE	RMSE	ACC	MAE	RMSE	ACC
DMA-Nets_g	20.73	25.41	0.657	31.67	42.89	0.644	43.62	60.93	0.630
DMA-Nets_l	19.97	26.01	0.671	30.01	41.93	0.645	43.84	58.41	0.629
DMA-Nets	18.00	24.19	0.669	28.23	40.58	0.651	41.37	56.40	0.640

6 CONCLUSION

In this article, we present a neural network model for forecasting citations of scientific publications. On top of a traditional Seq2Seq architecture, this model constructs a hierarchical dynamic attention layer considering both observation and prediction sequences. To enable the model to represent

Table 8. Hyper-parameter Analysis for DMA-Nets

	Hyper-parameters									USPTO		MAG	
	d_{emb}	d_h	h	d_q	m_1	d_{c1}	m_2	d_{c2}	dropout	MAE	ACC	MAE	ACC
base	64	256	4	64	4	64	2	32	0.1	**55.79**	**0.857**	18.00	**0.669**
d_h	64	128	4	32	4	32	2	32	0.1	56.72	0.854	18.77	0.652
	64	64	4	16	4	16	2	32	0.1	57.50	0.843	19.97	0.632
d_{emb}	32	256	4	64	4	64	2	16	0.1	56.24	0.849	18.77	0.667
	16	256	4	64	4	64	2	8	0.1	56.99	0.848	18.70	0.666
h, m_1	64	256	8	32	8	32	2	32	0.1	56.15	**0.857**	17.69	0.656
	64	256	2	128	2	128	2	32	0.1	56.22	0.856	19.03	**0.669**
m_2	64	256	4	64	4	64	4	16	0.1	56.97	0.853	**17.15**	0.665
dropout	64	256	4	64	4	64	2	32	0.3	57.52	0.856	17.97	0.665

interconnected dependencies across observation sequences and prediction sequences, we employ a local temporal attention mechanism to allow the model to look back along the temporal dimension and fuse more complicated intra-encoder and intra-decoder hidden attentional states. Additionally, the global multi-context attention layer encourages the model to learn the temporal point process from a global viewpoint by considering not only observations but also the predictions that have already been made. We demonstrate the performance improvement of our model on two real-world datasets collected from USPTO and MAG. Experimental results demonstrate that our model can consistently outperform state-of-the-art temporal point process modeling methods for the task of citation forecasting.

REFERENCES

[1] Daniel E. Acuna, Stefano Allesina, and Konrad P. Kording. 2012. Future impact: Predicting scientific success. *Nature* 489, 7415 (2012), 201.

[2] Carl T. Bergstrom, Jevin D. West, and Marc A. Wiseman. 2008. The eigenfactor™ metrics. *Journal of Neuroscience* 28, 45 (2008), 11433–11434.

[3] James Bessen. 2008. The value of US patents by owner and patent characteristics. *Research Policy* 37, 5 (2008), 932–945.

[4] Aggelos Bletsas and John N. Sahalos. 2009. Hirsch index rankings require scaling and higher moment. *Journal of the American Society for Information Science and Technology* 60, 12 (2009), 2577–2586.

[5] Lutz Bornmann and Werner Marx. 2013. *Standards for the Application of Bibliometrics in the Evaluation of Individual Researchers Working in the Natural Sciences.* Technical Report.

[6] Tibor Braun, Wolfgang Glänzel, and András Schubert. 2006. A Hirsch-type index for journals. *Scientometrics* 69, 1 (2006), 169–173.

[7] Tanmoy Chakraborty, Suhansanu Kumar, Pawan Goyal, Niloy Ganguly, and Animesh Mukherjee. 2015. On the categorization of scientific citation profiles in computer science. *Commun. ACM* 58, 9 (2015), 82–90.

[8] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259* (2014).

[9] Daizong Ding, Mi Zhang, Xudong Pan, Min Yang, and Xiangnan He. 2019. Modeling extreme events in time series prediction. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1114–1122.

[10] Nan Du, Hanjun Dai, Rakshit Trivedi, Utkarsh Upadhyay, Manuel Gomez-Rodriguez, and Le Song. 2016. Recurrent marked temporal point processes: Embedding event history to vector. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1555–1564.

[11] Leo Egghe. 2006. Theory and practise of the g-index. *Scientometrics* 69, 1 (2006), 131–152.

[12] Leo Egghe. 2014. A good normalized impact and concentration measure. *Journal of the Association for Information Science and Technology* 65, 10 (2014), 2152–2154.

[13] Oleksandr Ferludin, Arno Eigenwillig, Martin Blais, Dustin Zelle, Jan Pfeifer, Alvaro Sanchez-Gonzalez, Sibon Li, Sami Abu-El-Haija, Peter Battaglia, Neslihan Bulut, et al. 2022. TF-GNN: Graph neural networks in TensorFlow. *arXiv preprint arXiv:2207.03522* (2022).

[14] Emilio Ferrara and Alfonso E. Romero. 2013. Scientific impact evaluation and the effect of self-citations: Mitigating the bias by discounting the h-index. *Journal of the American Society for Information Science and Technology* 64, 11 (2013), 2332–2339.

[15] Vladimir Filimonov and Didier Sornette. 2015. Apparent criticality and calibration issues in the Hawkes self-excited point process model: Application to high-frequency financial data. *Quantitative Finance* 15, 8 (2015), 1293–1314.

[16] Alan G. Hawkes. 1971. Spectra of some self-exciting and mutually exciting point processes. *Biometrika* 58, 1 (1971), 83–90.

[17] Agnès Helmstetter and Didier Sornette. 2002. Subcritical and supercritical regimes in epidemic models of earthquake aftershocks. *Journal of Geophysical Research: Solid Earth* 107, B10 (2002), ESE–10.

[18] Jorge E. Hirsch. 2005. An index to quantify an individual's scientific research output. *Proceedings of the National academy of Sciences* 102, 46 (2005), 16569–16572.

[19] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation* 9, 8 (1997), 1735–1780.

[20] Hyun Jin Jang, Han-Gyun Woo, and Changyong Lee. 2017. Hawkes process-based technology impact analysis. *Journal of Informetrics* 11, 2 (2017), 511–529.

[21] Rahul Jha, Amjad-Abu Jbara, Vahed Qazvinian, and Dragomir R. Radev. 2017. NLP-driven citation analysis for scientometrics. *Natural Language Engineering* 23, 1 (2017), 93–130.

[22] Taoran Ji, Zhiqian Chen, Nathan Self, Kaiqun Fu, Chang-Tien Lu, and Naren Ramakrishnan. 2019. Patent citation dynamics modeling via multi-attention recurrent networks. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI 2019)* (Macao, China, August 10–16, 2019). 2621–2627. https://doi.org/10.24963/ijcai.2019/364

[23] Jinseok Kim, Jana Diesner, Heejun Kim, Amirhossein Aleyasen, and Hwan-Min Kim. 2014. Why name ambiguity resolution matters for scholarly big data research. In *Proceedings of the 2014 IEEE International Conference on Big Data (Big Data '14)*. IEEE, 1–6.

[24] Diederik P. Kingma and Jimmy Ba. 2014. ADAM: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[25] Thomas N. Kipf and Max Welling. 2016. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308* (2016).

[26] David F. Klosik and Stefan Bornholdt. 2014. The citation wake of publications detects nobel laureates' papers. *PloS One* 9, 12 (2014).

[27] Changyong Lee, Yangrae Cho, Hyeonju Seol, and Yongtae Park. 2012. A stochastic patent citation analysis approach to assessing future technological impacts. *Technological Forecasting and Social Change* 79, 1 (2012), 16–29.

[28] Changyong Lee, Ohjin Kwon, Myeongjung Kim, and Daeil Kwon. 2018. Early identification of emerging technologies: A machine learning approach using multiple patent indicators. *Technological Forecasting and Social Change* 127 (2018), 291–303.

[29] Yunjae Lee, Jinha Chung, and Minsoo Rhu. 2022. SmartSAGE: Training large-scale graph neural networks using in-storage processing architectures. *arXiv preprint arXiv:2205.04711* (2022).

[30] Hanwen Liu, Huaizhen Kou, Chao Yan, and Lianyong Qi. 2019. Link prediction in paper citation network to construct paper correlation graph. *EURASIP Journal on Wireless Communications and Networking* 2019, 1 (2019), 1–12.

[31] Xin Liu, Junchi Yan, Shuai Xiao, Xiangfeng Wang, Hongyuan Zha, and Stephen M. Chu. 2017. On predictive patent valuation: Forecasting patent citations and their types. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence.*

[32] Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025* (2015).

[33] Francesco Alessandro Massucci and Domingo Docampo. 2019. Measuring the academic reputation through citation networks via PageRank. *Journal of Informetrics* 13, 1 (2019), 185–201.

[34] Amy McGovern, Lisa Friedland, Michael Hay, Brian Gallagher, Andrew Fast, Jennifer Neville, and David Jensen. 2003. Exploiting relational structure to understand publication patterns in high-energy physics. *ACM SIGKDD Explorations Newsletter* 5, 2 (2003), 165–172.

[35] Lokman I. Meho. 2007. The rise and rise of citation analysis. *Physics World* 20, 1 (2007), 32.

[36] Hongyuan Mei and Jason M. Eisner. 2017. The neural Hawkes process: A neurally self-modulating multivariate point process. In *Advances in Neural Information Processing Systems.* 6754–6764.

[37] Swapnil Mishra, Marian-Andrei Rizoiu, and Lexing Xie. 2016. Feature driven and point process approaches for popularity prediction. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management.* ACM, 1069–1078.

[38] Bluma C. Peritz. 1992. On the objectives of citation analysis: Problems of theory and method. *Journal of the American Society for Information Science* 43, 6 (1992), 448–451.

[39] Nataliia Pobiedina and Ryutaro Ichise. 2016. Citation count prediction as a link prediction problem. *Applied Intelligence* 44, 2 (2016), 252–268.

[40] Gangan Prathap. 2014. A three-class, three-dimensional bibliometric performance indicator. *Journal of the Association for Information Science and Technology* 65, 7 (2014), 1506–1508.
[41] Filippo Radicchi and Claudio Castellano. 2013. Analysis of bibliometric indicators for individual scholars in a large data set. *Scientometrics* 97, 3 (2013), 627–637.
[42] Filippo Radicchi, Santo Fortunato, and Claudio Castellano. 2008. Universality of citation distributions: Toward an objective measure of scientific impact. *Proceedings of the National Academy of Sciences* 105, 45 (2008), 17268–17272.
[43] Huawei Shen, Dashun Wang, Chaoming Song, and Albert-László Barabási. 2014. Modeling and predicting popularity dynamics via reinforced poisson processes. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence.*
[44] Arnab Sinha, Zhihong Shen, Yang Song, Hao Ma, Darrin Eide, Bo-june Paul Hsu, and Kuansan Wang. 2015. An overview of Microsoft academic service (MAS) and applications. In *Proceedings of the 24th International Conference on World Wide Web.* ACM, 243–246.
[45] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems.* 3104–3112.
[46] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems.* 5998–6008.
[47] Kang Wang, Kenli Li, Liqian Zhou, Yikun Hu, Zhongyao Cheng, Jing Liu, and Cen Chen. 2019. Multiple convolutional neural networks for multivariate time series prediction. *Neurocomputing* 360 (2019), 107–119.
[48] Yongqing Wang, Huawei Shen, Shenghua Liu, Jinhua Gao, and Xueqi Cheng. 2017. Cascade dynamics modeling with attention-based recurrent neural network. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence.* AAAI Press, 2985–2991.
[49] Jiaqi Wen, Liyun Wu, and Jianping Chai. 2020. Paper citation count prediction based on recurrent neural network with gated recurrent unit. In *Proceedings of the 2020 IEEE 10th International Conference on Electronics Information and Emergency Communication (ICEIEC '20).* IEEE, 303–306.
[50] Xian Wu, Baoxu Shi, Yuxiao Dong, Chao Huang, Louis Faust, and Nitesh V. Chawla. 2018. Restful: Resolution-aware forecasting of behavioral time series data. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management.* 1073–1082.
[51] Shuai Xiao, Junchi Yan, Mehrdad Farajtabar, Le Song, Xiaokang Yang, and Hongyuan Zha. 2019. Learning time series associated event sequences with recurrent point process networks. *IEEE Transactions on Neural Networks and Learning Systems* (2019).
[52] Shuai Xiao, Junchi Yan, Changsheng Li, Bo Jin, Xiangfeng Wang, Xiaokang Yang, Stephen M. Chu, and Hongyuan Zha. 2016. On modeling and predicting individual paper citation count over time. In *Proceedings of IJCAI.* 2676–2682.
[53] Shuai Xiao, Junchi Yan, Xiaokang Yang, Hongyuan Zha, and Stephen M. Chu. 2017. Modeling the intensity function of point process via recurrent neural networks. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence.*
[54] Junchi Yan, Shuai Xiao, Changsheng Li, Bo Jin, Xiangfeng Wang, Bin Ke, Xiaokang Yang, and Hongyuan Zha. 2016. Modeling contagious merger and acquisition via point processes with a profile regression prior. In *Proceedings of IJCAI.* 2690–2696.
[55] Rui Yan, Congrui Huang, Jie Tang, Yan Zhang, and Xiaoming Li. 2012. To better stand on the shoulder of giants. In *Proceedings of the 12th ACM/IEEE-CS Joint Conference on Digital Libraries.* 51–60.
[56] Rui Yan, Jie Tang, Xiaobing Liu, Dongdong Shan, and Xiaoming Li. 2011. Citation count prediction: Learning to estimate future citations for literature. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management.* 1247–1252.
[57] Shuang-Hong Yang and Hongyuan Zha. 2013. Mixture of mutually exciting processes for viral diffusion. In *Proceedings of the International Conference on Machine Learning.* 1–9.
[58] Xiao Yu, Quanquan Gu, Mianwei Zhou, and Jiawei Han. 2012. Citation prediction in heterogeneous bibliographic networks. In *Proceedings of the 2012 SIAM International Conference on Data Mining.* SIAM, 1119–1130.
[59] Sha Yuan, Jie Tang, Yu Zhang, Yifan Wang, and Tong Xiao. 2022. Modeling and predicting citation count via recurrent neural network with long short-term memory. *arXiv preprint arXiv:1811.02129* (2022).
[60] Ke Zhou, Hongyuan Zha, and Le Song. 2013. Learning triggering kernels for multi-dimensional hawkes processes. In *Proceedings of the International Conference on Machine Learning.* 1301–1309.

Received 17 January 2022; revised 14 June 2023; accepted 12 February 2024