HINTS: Citation Time Series Prediction for New Publications via Dynamic Heterogeneous Information Network Embedding

Song Jiang[1], Bernard J. Koch[2], Yizhou Sun[1]
[1]Department of Computer Science, University of California, Los Angeles, Los Angeles, CA
[2] Department of Sociology, University of California, Los Angeles, Los Angeles, CA
[1]{songjiang, yzsun}@cs.ucla.edu [2]bernardkoch@ucla.edu

ABSTRACT

Accurate prediction of scientific impact is important for scientists, academic recommender systems, and granting organizations alike. Existing approaches rely on many years of leading citation values to predict a scientific paper's citations (a proxy for impact), even though most papers make their largest contributions in the first few years after they are published. In this paper, we tackle a new problem: predicting a new paper's citation time series from the date of publication (i.e., without leading values). We propose **HINTS**, a novel end-to-end deep learning framework that converts citation signals from dynamic heterogeneous information networks (DHIN) into citation time series. HINTS imputes pseudo-leading values for a paper in the years before it is published from DHIN embeddings, and then transforms these embeddings into the parameters of a formal model that can predict citation counts immediately after publication. Empirical analysis on two real-world datasets from Computer Science and Physics show that HINTS is competitive with baseline citation prediction models. While we focus on citations, our approach generalizes to other "cold start" time series prediction tasks where relational data is available and accurate prediction in early timestamps is crucial.

CCS CONCEPTS

• **Information systems** → *Data mining*.

KEYWORDS

Citation prediction; time series; dynamic heterogeneous information network; science of science

ACM Reference Format:
Song Jiang[1], Bernard J. Koch[2], Yizhou Sun[1]. 2021. HINTS: Citation Time Series Prediction for New Publications via Dynamic Heterogeneous Information Network Embedding. In *Proceedings of the Web Conference 2021 (WWW '21), April 19–23, 2021, Ljubljana, Slovenia.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3442381.3450107

1 INTRODUCTION

Predicting the "impact" of scientific research is crucial for authors to decide what to study and where to submit their research, for readers or recommender systems to identify important/relevant contributions in a vast scientific literature, and for funding agencies to identify promising young scientists and fields for future support. Because impact is hard to quantify, citation counts of scientific papers are often used as an approximation [11, 29, 44].

To predict future citations, previous works [20, 27, 38, 42] have often relied on observed citations (i.e., leading citations values) in the first few years after publication. These leading value-based solutions have taken both parametric and machine learning approaches. [27, 38, 42] propose formal models to encode assumptions and prior knowledge about how papers are cited (e.g., that citation trajectories follows a log-normal distribution). They then use leading citations for the first several years after a paper is published to infer paper-specific parameters to predict long-term citation counts. Machine learning approaches have used representation learning via recurrent neural networks (RNN) to automatically capture complex citation patterns from leading values, followed by another RNN as decoder to make predictions [1, 47].

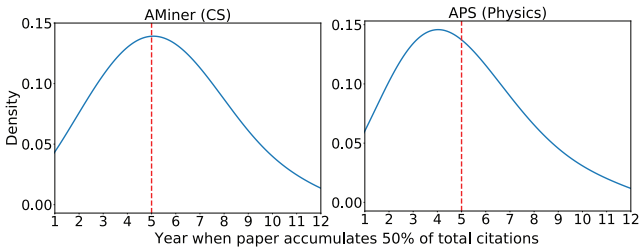

Figure 1: Year from publication when an academic paper has accumulated more than 50% of cumulative citations. Citations only counted up until 12 years after publication. Density of distribution shown in blue. Median shown in red. Zero citation papers removed. Left: 351,926 Computer Science papers (AMiner) published from 2000 to 2005; Right: 71,142 Physics papers (APS) published from 1995 to 2000.

A significant problem for these approaches is that many scientific papers have peak impact in the first few years after publication, when leading values are not yet available. For example in both Computer Science and Physics, we find that half of all cited papers accumulate the majority of their citations within five years of publication (Fig. 1). In fields with very fast publication rates (e.g., machine learning) it is not practical to wait three to five years before predicting impact. In this paper, we therefore tackle a new challenge: generating citation time series for newly-published papers *without any leading values*. To the best of our knowledge, we are the first to focus on predicting citation time series from the time of publication event. While we focus on scientific impact prediction, this "cold start" issue is common to many time series prediction

WWW '21, April 19–23, 2021, Ljubljana, Slovenia

ACM ISBN 978-1-4503-8312-7/21/04.
https://doi.org/10.1145/3442381.3450107

tasks where earlier outcomes are more critical than later ones (e.g., in-links flow of new webpages, revenue streams for start-ups).

One way to avoid relying on leading values for impact prediction is to leverage clues visible to domain experts before they even read the paper. By reading the title, abstract, and bibliography, researchers can identify whether the paper is about a hot topic in their field. Within the author list, they can identify productive labs and reputable researchers. Papers in prestigious venues are also likely to be higher quality. To leverage these "metadata" for long-term citation prediction, previous studies [9, 43, 44, 46] have manually designed complex features. However, feature engineering is time-consuming, non-transferable and rarely complete. Moreover, these approaches rarely use the additional information encoded in the relationships between metadata, or their historical temporal trends. Using historical and relational context, domain experts can identify not just popular topics and reputable authors, but also trending topics and "rising star" researchers.

To leverage all of the predictive "hints" available to domain experts before reading the paper, we encode papers, authors, topics, and venues in a dynamic heterogeneous information network (DHIN). A DHIN captures not just a paper's metadata, but also the relationships between those metadata and their historical trends. This additional information allows us to predict citation counts without leading values, using our proposed end-to-end framework called ***HINTS*** (Heterogeneous Information Network to Time Series).

Composed of three modules, HINTS translates temporal and relational information from a DHIN before publication into a citation time series after publication. In the first module, we use a temporally-aligned GNN which concurrently learns effective embeddings for all nodes in each year's heterogeneous bibliographic network. Because static GNNs [14, 17, 25] do not preserve the underlying evolution of nodes in a bibliographical network, we apply a smooth regularizer to align the positions of nodes in the embedding space across time stamps. This approach allows us to capture the temporal trends of nodes (e.g., the rising star phenomenon). We show that this alignment regularization can be easily integrated into GNN models and improves predictive performance. The second module is a weighted imputation mechanism to estimate a sequence of embeddings for a new paper in the years prior to its publication. By aggregating the dynamics of the metadata, this imputation approximates the temporal trajectory of the new paper in the years before it is published. This learned temporal trajectory serves as "pseudo"-leading values for time series prediction after publication. The third module is a parametric generator based on [38] that encodes prior assumptions about citation processes to predict long-term citation time series. Using an RNN followed by fully-connected layers, we transform the imputed paper embedding trajectory into the parameters of this generator. In conclusion, HINTS combines novel approaches for encoding DHINs (Module 1), synthesizing leading values prior to publication (Module 2), and formal modeling assumptions (Module 3) into an end-to-end framework that can predict citation times series from the time of publication. We note that this framework can be generically adapted to other "cold start" time series problems where pre-event temporal and relational data are available.

Empirically, we apply HINTS to two real-world academic datasets in Computer Science and Physics with extensive experiments. The results show that HINTS achieves significant and consistent improvements compared with baseline cold-start prediction approaches. Ablation studies on variants of HINTS also demonstrate the importance of each component of our proposed model.

Our contributions can be summarized as follows:

- We tackle a new, challenging "cold start" time series problem: predicting a new paper's citation time series without leading citation values.
- We propose a novel framework called HINTS that converts signals from a DHIN into signals for citation time series generation.
- We conduct extensive experiments on two real-world large-scale bibliographic datasets from different fields to demonstrate HINTS' effectiveness at impact prediction.

2 PROBLEM STATEMENT

In this section, we first introduce necessary definitions used throughout this paper. Then we present a formal definition of the new paper citation time series prediction problem.

2.1 Preliminaries

Definition 2.1. **Heterogeneous information network.** A *heterogeneous information network (HIN)* [32] is defined as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with a node type mapping function $\varphi: \mathcal{V} \rightarrow \mathcal{T}$ and an edge type mapping function $\phi : \mathcal{E} \rightarrow \mathcal{R}$. $\mathcal{T}$ and $\mathcal{R}$ denote all the predefined types of node and edges, where $|\mathcal{T}| + |\mathcal{R}| > 2$.

A bibliographic network [5, 30] is a type of heterogeneous information network. Scientific papers are the central nodes, with their metadata as neighbors. In our case, a paper's metadata includes referenced papers, authors, keywords, and a venue. Given these typed components, a network schema [31] $\tilde{G} = (\mathcal{T}, \mathcal{R})$ can be utilized to abstract the node types and edge types at the meta level. The schema of our bibliographic network is shown in Fig. 2.

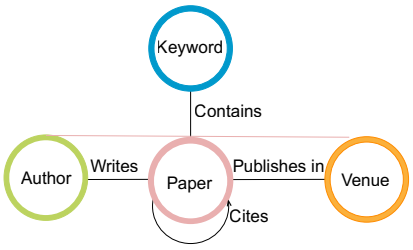

Figure 2: The schema of a bibliographic network. The nodes include paper, author, keyword, and venue, while their four relationships are: paper-cites-paper, author-writes-paper, paper-contains-keywords and paper-publishes in-venue.

In reality, bibliographic networks are constantly evolving. For instance, new papers will be published, new researchers will join the community, and new keywords will be created. These new entities will be added into the network, bringing new edges as well. Formally, given T timestamps, we define a dynamic heterogeneous information network as follows.

Definition 2.2. **Dynamic heterogeneous information network.** A *dynamic heterogeneous information network (DHIN)* is a sequence of HIN snapshots, denoted by $\langle \mathcal{G}_t \rangle_{t=1}^{T} = \{\mathcal{G}^1, \mathcal{G}^2, \ldots, \mathcal{G}^T\}$, where $\mathcal{G}^t = (\mathcal{V}^t, \mathcal{E}^t)$ $(1 \le t \le T)$ represents the heterogeneous graph snapshot and its corresponding node set and edge set at time t.

In our case, a dynamic bibliographic network is a DHIN that consists of T sequential snapshots of the evolving bibliographic network in every calendar year. Thus $\mathcal{G}^t$ is a bibliographic graph snapshot at year t whose node and edge types are described in Fig. 2.

2.2 Problem Definition

We now formalize the new paper citation time series prediction problem using a dynamic bibliographic network with network schema shown in Fig. 2. For each new paper p, we represent its citation counts over next L years after publication as a sequence $c_p = \{c_p^1, \ldots, c_p^l, \ldots, c_p^L\}$, where c_p^l denotes the citation count paper p will receive in the l-th year after publication.

The New Paper Citation Time Series Prediction Problem. Given a dynamic bibliographic network $\langle \mathcal{G}_t \rangle_{t=1}^{T-1}$ and a newly published paper p, our goal is to learn a function $f(\cdot)$ that maps paper p, given its context described by $\langle \mathcal{G}_t \rangle_{t=1}^{T-1}$, to its citation time series in future L years' after the publication year T, which is denoted as follows:

$$(\langle \mathcal{G}_t \rangle_{t=1}^{T-1}, p) \xrightarrow{f(\cdot)} \{c_p^1, \ldots, c_p^l, \ldots, c_p^L\}. \tag{1}$$

Note p is not in $\langle \mathcal{G}_t \rangle_{t=1}^{T-1}$, and no citations are received before T.

3 PROPOSED FRAMEWORK: HINTS

In this section, we introduce our proposed framework, HINTS. We first describe the intuition behind why a DHIN provides key information for paper citation prediction. Then we breakdown the three modules used by HINTS to turn signals from DHIN into citation time series in detail. The overall framework of HINTS is shown in Fig. 3.

3.1 Motivation for HINTS

Although the reasons that some scientific papers achieve high impact are complicated, there are several cues identifiable by domain experts and network scientists that can predict impact. Ideally, representation learning of a DHIN should capture the following factors predictive of citation:

Topic. A paper is more likely to be cited by readers from a similar research area. Papers on hot or trending topics generally attract more attention and thus receive more citations (e.g., artificial intelligence in recent years). *Keywords* can serve as proxies for topic, because they are carefully selected by the authors to describe the new paper.

Author Status. Readers are more likely to search for papers by reputable authors, the advisees of reputable authors, or rising stars because of the high quality of their work.

Venue Status. Because of peer review, readers are more likely to assume that papers published in prestigious venues in their field are higher quality.

Bibliography. Highly influential papers do not start from scratch, instead, they stand on the shoulder of giants. [8] Citing high-impact papers is a baseline signal for relevance and potential impact [35].

Temporal Cues. Domain experts rely not only on the content of metadata when scanning papers, but also on knowledge of the temporal trends of these metadata. For example, a "rising star" might not only have a well-known advisor (relational context), but their network centrality will increase over time as they publish more influential papers (temporal context).

"Fitness". While domain experts can quickly identify the above cues, there are other intangible factors predictive of citation that are not encoded in metadata, such as the rigor of the work or the value of its contributions. For example, the graph convolution network (GCN) paper [17] was a milestone that allowed for new applications of deep learning to graphs. Network scientists have used the term "fitness" to generically capture these latent intangibles. [15, 38]

The three modules of HINTS are designed to automatically detect these six types of information and leverage them for citation prediction. Note that the first five factors can be implicitly encoded in a DHIN connecting keywords, authors, venues and papers (i.e., metadata) as Fig. 2 over time. In the first module, we learn low-dimensional representation vectors of metadata across all time slices concurrently. The learned embeddings naturally capture topic, author status, venue status bibliography, and their trends. Because leading citation values do not exist for new papers, the second module in HINTS uses these node embeddings to impute embeddings in the years before a paper's publication by averaging the embeddings of it's metadata nodes in those years. Because some factors (e.g., author status) are more predictive of citation than others (e.g., bibliography), our framework learns weights to perform this averaging. The imputed embedding trajectory encodes all above factors and serves as the pseudo-leading values before publication. In the third module, we translate our imputed embeddings into the parameters of a parametric citation generator. This model, adapted from [38], encodes prior knowledge about citation processes and captures intangible factors (i.e., "fitness") to predict citation counts in the years immediately following publication. We introduce the details of these three modules in following subsections.

3.2 Dynamic Heterogeneous Network Embedding via Temporally-aligned GNN

Given a static heterogeneous bibliographic network, several embedding methods [5, 7, 13, 25] have been proposed. Without loss of generality, we employ a relational graph convolution network (R-GCN) [25] to encode nodes into low-dimensional vectors. R-GCN learns a relation-aware function that updates a node's representation by weighted aggregation of its neighbors according to the corresponding relation types. Formally, given a dynamic bibliographic network $\langle \mathcal{G}_t \rangle_{t=1}^{T}$, each $\mathcal{G}_t$ can be seen as a static network at time t. Let $h_{i,t}^{(k)} \in \mathcal{R}^{d(k)}$ be the embedding vector of node i in the k-th layer at time t, where $d(k)$ denotes the dimension for k-th layer, it will be updated via R-GCN as:

$$h_{i,t}^{(k+1)} = \sigma\Big(\sum_{r \in \mathcal{R}} \sum_{j \in N_{i,t}^r} \frac{1}{|N_{i,t}^r|} W_r^{(k)} h_{j,t}^{(k)} + W_0^{(k)} h_{i,t}^{(k)} \Big), \tag{2}$$

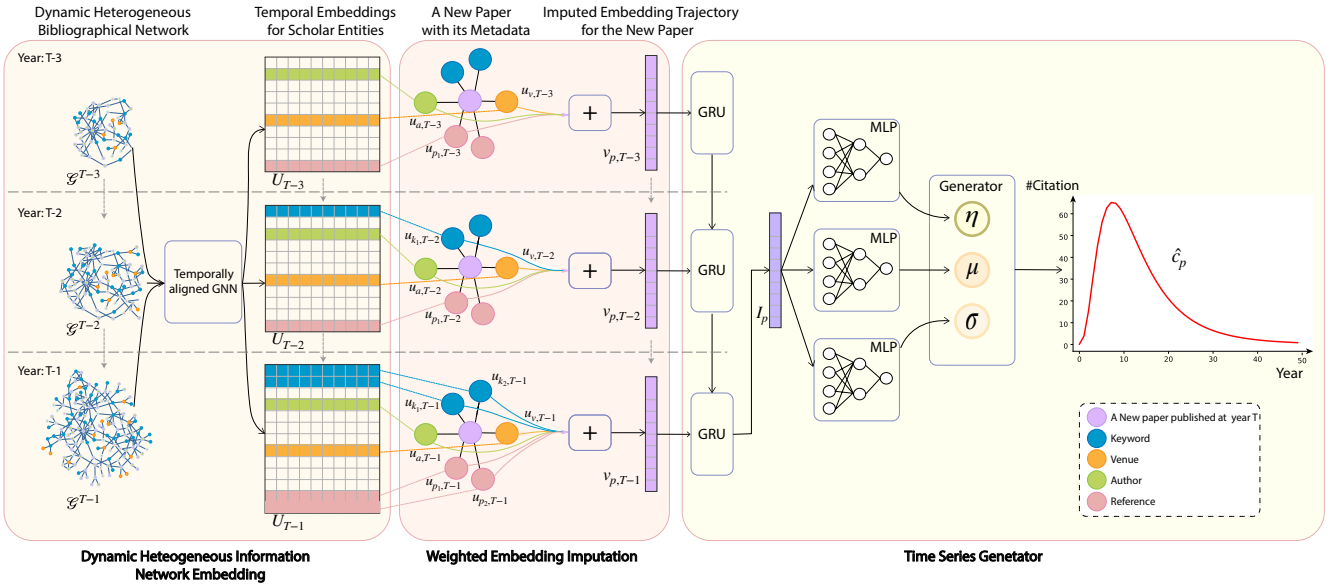

Figure 3: The overall architecture of HINTS. For a new paper published in year T, HINTS first learns temporally-aligned embeddings for each metadata neighbor in the dynamic heterogeneous bibliographical network that exists in the years preceding T (three in this example). An imputed embedding trajectory is built for the new paper (purple node) by computing a weighted average of the neighbors' embeddings. Note that some metadata nodes may not exist across all previous timesteps. (e.g., a new keyword proposed only one year ago). After temporal encoding by an RNN, the imputed embedding trajectory is transformed into three interpretable parameters, based on which HINTS generates the new paper's future citation time series.

where $\mathcal{R}$ denotes the set of predefined types of edges/relations in the bibliographic network, while $N_{i,t}^r$ represents the set of neighbors of node i under relation $r \in \mathcal{R}$ at time t. $W_r^{(k)}$ and $W_0^{(k)}$ are the weight matrices of k-th layer, and σ is a non-linear activity function.

Temporally-aligned Graph Neural Network. R-GCN shows superior performance across many graph-related tasks, but extending it to dynamic settings is still challenging. An important contribution of HINTS is a method for aligning graph neural networks temporally. Because each individual bibliographical network describes a snapshot of the research community in the corresponding year, we first apply R-GCN annually to encode each bibliographical network separately. To ensure that each year's embeddings are in the same space (i.e., they are comparable), we make transformation weight matrices $W_r^{(k)}$ and $W_0^{(k)}$ shared across different timestamps. Second, unlike general dynamic networks where the characteristics of nodes may change rapidly (e.g., online social networks, or protein-protein interaction networks), most entities in dynamic bibliographic information networks do not change too much within a short time frame. For example, a researcher's interests or venue's theme are likely to be similar in adjacent years. Motivated by this observation, we force the embeddings for the same entity in nearby years to be close to each other by introducing a temporal smoothing regularizer $L_{t,t+1}^{\langle time \rangle}$:

$$L_{t,t+1}^{\langle time \rangle} = \frac{1}{|V_t \cap V_{t+1}|} \sum_{i \in V_t \cap V_{t+1}} \|u_{i,t} - u_{i,t+1}\|_2^2, \tag{3}$$

where $u_{i,t}$ denotes entity i's embedding at year t, and V_t is the node set of $\mathcal{G}^t$. After multiple layers of R-GCN operations (we use 2 layers in practice), the final embedding matrix for $\mathcal{G}^t$ is denoted as $U_t \in \mathcal{R}^{N_t \times D}$, where N_t is the number of nodes in $\mathcal{G}^t$, and D is the dimension of the embeddings. Note that although we use R-GCN here, our HINTS framework can accommodate many graph neural networks, e.g., GCN [17], GAT [36], HAN [41].

3.3 Weighted Embedding Imputation

To predict a newly published paper's long-term impact at time of publication, we need a distinct representation for it in the years before publication (i.e., pseudo-leading values). Although a paper is new, the metadata it is linked to may already exist in previous years' bibliographic networks. For example, a paper is usually published in a venue with a long track record, by co-authors with several previous publications, and with keywords that exist for quite a long time.

Using temporally-aligned GNN, we have already learned a vectorized representation for each metadata node which encodes both its historical trend and relationships with other nodes. Suppose one metadata node i appears in t-th year, and we have learned all its embeddings after t, which is a sequence denoted as $\{u_{i,t}, u_{i,t+1}, \ldots, u_{i,T}\}$. This sequence can be considered as an evolutionary trajectory for metadata i across time in the embedding space.

Given the above two preconditions, we can utilize the embedding sequences of a new paper's metadata neighbors to create an imputed embedding sequence that approximates its trajectory in the years prior to publication. One option to impute such embeddings is to

directly use the same R-GCN operator defined in Eq. 2. However, this operator will result in an embedding in a different space due to additional transformation. Alternatively, inspired by the method in [5], we impute a new paper's embedding sequence by aggregating its metadata's embeddings with type-aware trainable weights to preserve the unequal contribution of different kinds of metadata.

Formally, for a new paper p, given its metadata set $N_{p,t}$ at time t and the metadata type set M, its imputed representation $v_{p,t}$ at time t would be derived from:

$$v_{p,t} = \sum_{m \in M} \sum_{i \in N_{p,t}^m} w_m * u_{i,t} / |N_{p,t}^m|. \tag{4}$$

By applying Eq. 4 in every timestamp, we can construct an imputed embedding sequence $V_p = \{v_{p,t}, v_{p,t+1}, ..., v_{p,T-1}\}$ for the new paper p, where t is the first year that p's metadata is observed. w_m is the weight for metadata type m, which will be learned in training stage. By integrating the temporal trends of its metadata, this imputed embedding sequence could be a good proxy for the hypothetical trend of the paper before publication.

3.4 Time Series Generator

Base on the imputed embedding sequence for paper V_p, we predict future impact by learning a function $g(\cdot) : V_p \rightarrow c_p^l$ that transforms network signals encoded in the imputation V_p to the new paper's long-term citation time series. One straightforward solution is directly "translating" the imputation sequence to a citation sequence with an encoder-decoder schema (e.g., seq2seq [33]). This approach has two major limitations. First it cannot generate flexible long-term citation predictions. Once learned, the decoder can only produce discrete sequences of predefined length. Furthermore, citation time series of scientific publications have been successfully modeled with some minimal assumptions [29, 38]. This solution, however, fails to leverage this important prior knowledge in prediction.

Intuitively, a paper's impact fades gradually since new ideas and new research topics always appear and attract researchers' attention. Therefore, following [38], we model a new paper's citation trajectory as a log-normal survival function along time l, which is formalized as:

$$P_p(l) = \frac{1}{l\sqrt{2\pi}\sigma_p} \exp\left[- \frac{(\ln l - \mu_p)^2}{2\sigma_p^2} \right], \tag{5}$$

where μ_p describes the timestamp when paper p will reach its citation peak, and σ_p indicates the rate of decay of paper p's citations. Moreover, as discussed in 3.1, the "fitness" makes significant contributions to a paper's citations, so another parameter η_p is used to model it. The citation counts are thus positively related to η_p. Integrated across η_p, the predicted cumulative citation counts $\hat{C_p^l}$ of paper p at l-th year after publication can be generated by

$$\hat{C_p^l} = \alpha \big[\exp\left(\eta_p * \Phi(\frac{\ln l - \mu_p}{\sigma_p})\right) - 1 \big] \tag{6}$$

where $\Phi(x)$ is

$$\Phi(x) = (2\pi)^{-1/2} \int_{-\infty}^{x} e^{-y^2/2}\, dy. \tag{7}$$

Following [38], a parameter α is added for the average number of references each new paper contains. Note that α is a global parameter that will be fixed during the model training process.

In Eq. 6, the three parameters η_p, μ_p and σ_p will be estimated for each new paper to generate its citation time series. In contrast with [38] who use the first several years' (e.g., 10 years) of leading citation values to infer parameters, we learn these three parameters by transforming the signals encoded in the DHIN prior to publication. Specifically, the imputed embedding sequence V_p is first temporally encoded into a single vector I_p through a recurrent neural network (RNN) with GRU [6] to model the new paper's temporal trajectory, and then three fully connected layers decode I_p into the three parameters respectively. (See Fig. 3)

It is worth noting that compared to the straightforward encoder-decoder method, our time series generator has three major advantages: (1) It can generate citation time series predictions in a flexible manner, i.e., assigning l a time length, (2) It leverages prior knowledge about citation patterns to achieve better performance, and (3) the three parameters are reasonably interpretable. We show this in detail in Sec. 4.

After accumulating L years citation counts for new paper p with Eq. 6, we transform the cumulative citation counts $\{\hat{C_p^1}, ..., \hat{C_p^l}, ..., \hat{C_p^L}\}$ into citation counts in each individual year and build the predicted citation time series as $\{\hat{c_p^1}, ..., \hat{c_p^l}, ..., \hat{c_p^L}\}$. This predicted sequence is then compared to the ground-truth with a Mean Square Error loss function. The loss function is defined as follows:

$$L^{\langle pred \rangle} = \frac{1}{P} \sum_{p=1}^{P} \frac{1}{L} \sum_{l=1}^{L} (\log(\hat{c_p^l}) - \log(c_p^l))^2, \tag{8}$$

where c_p^l is the ground-truth citation count of paper p at the l-th year after publication, and P is the total number of papers. Following [2, 19], we use log-scale for citation counts to smooth the contribution of each paper to the total loss regardless of its citation level. Note that many paper actually won't receive any citations, so we add 1 count as pseudo citation value before taking the log transformation.

3.5 Objective

By putting the citation time series generator objective and the temporal alignment regularizer aforementioned together, the overall objective function of HINTS $\mathcal{J}$ is defined as :

$$\mathcal{J} = L^{\langle pred \rangle} + \beta \sum_{t=1}^{T-1} L_{t,t+1}^{\langle time \rangle}. \tag{9}$$

For the alignment regularizer, in contrast to the method described in [10] that updates embedding in chronological order, we align embeddings simultaneously across all timestamps such that the final aligned embedding U_t preserves every previous embeddings instead of only U_{t-1}. A hyperparameter β ($\beta > 0$) is used to control the degree of alignment. All parameters in HINTS are updated by optimizing this objective.

4 EXPERIMENTS

In this section, we evaluate HINTS' efficacy in predicting citation time series for new papers. We describe our experimental setting

and then present the results compared with baselines. We also breakdown the framework in ablation studies and interpretation analyses to understand how HINTS works.

4.1 Experimental Setup

Datasets. We use two publicly-available bibliographic datasets in different fields for our analyses: the *AMiner* [34] Computer Science dataset[1] and the *American Physical Society (APS)* Physics dataset[2]. AMiner covers papers in major computer science venues. We use data from 2000-2009 to build the model and data from 2010-2015 for evaluation. The APS dataset covers publications in APS physics journals. Similarly, we use years 1995-2004 for training and years 2005-2010 for testing. The distribution of cumulative citation counts (the number of papers vs. citation counts) for papers in the test set is shown in Fig. 4. We note that a large number of papers are rarely cited after publication, so we take a down-sampling to balance the highly and lowly cited papers in model training.

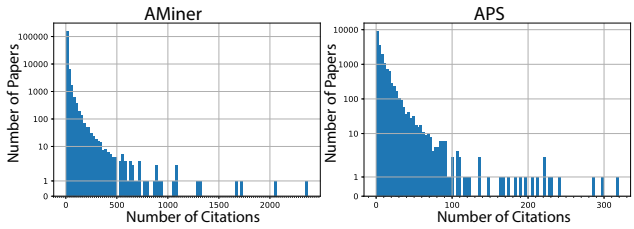

Figure 4: Distribution of cumulative citation counts within five years after publication for papers published in 2010 (AMiner), and 2005 (APS).

We construct annual snapshots of the heterogeneous bibliographic network for both datasets in each year with the network schema shown in Fig. 2. Because the keywords are not explicitly provided in the original APS dataset, we generate them by combining unigrams and key phrases extracted from the title of each paper using the method proposed in [26].

Baselines. Because the "cold start" citation time series prediction is a novel problem, there are no exact baselines for comparison, to our knowledge. Many state-of-the-art time series models (e.g., [22, 23, 39, 49]) are not applicable for prediction immediately after publication because they require leading values. Instead, we compare HINTS to plausible alternatives. We consider three types of baselines: 1) Models that use manually constructed features 2) Models designed to predict information cascades (citation can be construed as an information cascade), and 3) two variants of HINTS. The specific approaches we consider are:

- **Gradient Boosting Machine (GBM):** We extract scientific features designed by [3, 44] except those that are not available in our problem setting or data, e.g., "first-year citation," h-index. We then use them to predict time series with XGBoost [4].
- **DeepCas [19]:** A state-of-the-art deep learning model for popularity prediction based on random walks across an information cascade graph. Because of the "cold start" setting, we directly use the ego network of a new paper in the snapshot of the DHIN in the publication year as the initial cascade graph.
- **HINTS-GCN:** A variant of HINTS using a homogeneous GCN [17] instead of R-GCN.
- **HINTS-Seq:** A variant of HINTS that replaces the citation generator module with seq2seq [33], directly transforming imputed embedding sequences into discrete citation sequences.
- **HINTS:** Our proposed framework whose three modules are described in Sec. 3.

Evaluation Metrics. Following [2, 19], we use two log-scaled metrics to compare different models:
Mean Absolute Log-scaled Error (MALE)

$$MALE(c^l, \hat{c^l}) = \frac{1}{P}\sum_{i=1}^{P} |\log(\hat{c_p^l}) - \log(c_p^l)|. \tag{10}$$

Root Mean Square Log-scaled Error(RMSLE),

$$RMSLE(c^l, \hat{c^l}) = \sqrt{\frac{1}{P}\sum_{i=1}^{P} (\log(\hat{c_p^l}) - \log(c_p^l))^2}. \tag{11}$$

where c_p^l and $\hat{c_p^L}$ are the ground truth and predicted citation counts of paper p at the l-th year after publication respectively, and P is the total number of papers. As discussed in Sec. 3.4, we use log transformations because citation counts vary widely.

Implementation Details. We implement HINTS using Tensorflow 1.14. For the DHIN embedding module, we use two layers of GNN with 64 and 128 node (for both R-GCN and the variant with GCN). In the GNN layers, node features are randomly initialized in four different ranges according to the node type. The hidden dimension of HINT's RNN temporal encoder is set as 50. Finally, the hidden dimensions of three fully-connected layers are 20, 8, and 1, respectively. For HINTS-Seq, we also use GRU [6] as the RNN decoder. The coefficient of alignment β is set as 0.5.

For training hyperparameters, we set the learning rate to 0.01 for both datasets. We train for 700 epochs with a batch size of 3000 papers for AMiner, and 500 epochs with a batch size of 1200 papers for APS. We randomly initialize all the parameters and optimize them with Adam [16]. We run every experiment three times and report the average. All the experiments are conducted on a desktop machine with a 4-core i7-5860k CPU, 40G memory, and two Nvidia Titan X GPUs. The total running time (not including data prepossessing) is around 24 minutes for AMiner and around 10 minutes on APS with the above settings. Our data and code are available at: https://github.com/songjiang0909/HINTS_code.

4.2 Numerical Comparison Results

In this part, we examine the performance of HINTS comprehensively. We compare HINTS with baselines, conduct ablation studies on components and objective of HINTS, and analyze differences in HINTS ability to predict trajectories for low-citation and high-citation papers.

Comparison with Baselines. Table. 1 shows the first five years of predictions errors for all models. In general, HINTS outperforms our proposed baselines in almost every time step. On AMiner,

[1] https://aminer.org/citation
[2] https://journals.aps.org/datasets

Table 1: Results of effectiveness experiments on AMiner and APS datasets.

Dataset	Model	MALE						RMSLE					
		1st year	2nd year	3rd year	4 year	5 year	overall	1st year	2nd year	3rd year	4 year	5 year	overall
AMiner	GBM	**0.673**	0.971	1.069	1.383	1.332	1.085	**0.753**	**1.108**	1.283	1.624	1.685	1.291
	DeepCas	1.003	1.103	1.068	0.987	1.025	1.037	1.119	1.325	1.366	1.330	1.346	1.321
	HINTS-GCN	0.824	0.904	0.919	0.958	1.019	0.925	0.936	1.119	1.152	1.176	1.196	1.116
	HINTS-Seq	1.139	0.953	0.969	0.980	0.992	1.011	1.375	1.164	1.206	1.216	1.223	1.237
	HINTS	0.783	**0.866**	**0.879**	**0.877**	**0.865**	**0.854**	0.976	1.110	**1.146**	**1.155**	**1.154**	**1.111**
APS	GBM	0.952	0.968	0.972	0.982	1.103	0.995	1.151	1.168	1.189	1.214	1.355	1.215
	DeepCas	0.993	0.998	0.966	0.931	0.886	0.955	1.198	1.221	1.195	1.160	1.114	1.178
	HINTS-GCN	0.949	0.950	0.939	0.917	0.906	0.932	1.153	1.166	1.160	1.133	1.124	1.147
	HINTS-Seq	1.263	0.951	0.959	0.969	0.975	1.023	1.397	1.219	1.199	1.193	1.119	1.225
	HINTS	**0.934**	**0.936**	**0.923**	**0.903**	**0.875**	**0.914**	**1.135**	**1.151**	**1.142**	**1.127**	**1.102**	**1.132**

HINTS outperforms the best baseline, DeepCas, by 17.6% in terms of MALE and 15.8% in terms of RMSLE. And these numbers are 4.3% and 3.9% on APS. We speculate that DeepCas may suffer in "cold start" settings where the initial bibliographic cascade graph is quite small. The strong performance of GBM (particularly on AMiner) in early years is due to overfitting the majority papers that do not receive citations. However, GBM performance degrades sharply over time. In contrast, HINTS actually achieves better scores over time, indicating the importance of leveraging parametric assumptions for long-term citation prediction.

Ablation Study on HINTS Components. We further compare HINTS with two variants in order to evaluate the effectiveness of each module. First, we find that HINTS consistently outperforms the HINTS-Seq variant (Table. 1), again indicating the value of supplementing contextualized embeddings with domain knowledge encoded in formal models. Second, while HINTS-GCN outperforms HINTS-Seq, there is still a non-trivial gap in performance between HINTS-GCN and HINTS. This result underscores the utility of modeling heterogeneous relationships between metadata for citation prediction.

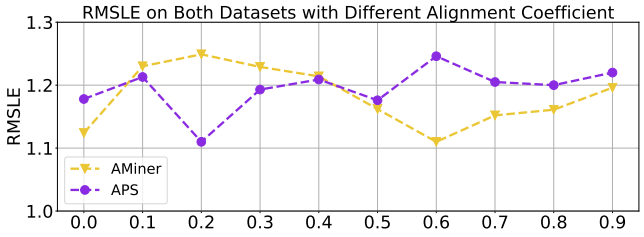

Figure 5: The average RMSLE across five years on AMiner and APS datasets with 10 different alignment coefficients.

Ablation Study on Alignment. The learning objective of HINTS balances citation prediction and the temporal alignment of embeddings. To measure the impact of the alignment regularizer ($\sum_{t=1}^{T-1} L_{t,t+1}^{\langle time \rangle}$), we conduct another ablation experiments with 10 different alignment coefficient β ranging from 0 to 0.9.

We compare the average RMSLE across five years on testing set of both AMiner and APS datasets. We also run HINTS three times for every β and report the mean results in Fig. 5. The optimal βs for AMiner and APS datasets are 0.6 and 0.2, respectively. Although the two datasets rely on the alignment regularization to varying degrees, these results show regularization improves performance for both datasets (especially APS) by learning more accurate embeddings for the DHIN. We also note that performance begins to degrade when β>0.7. This is because a larger β forces embeddings across years to be too similar. Actually, an extreme case is when β is large enough, the embeddings over time would be restricted almost the same, which makes the embeddings no longer reasonable.

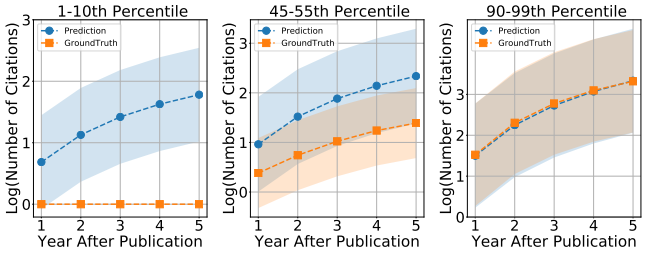

Figure 6: Predicted versus ground truth citations for AMiner papers published in 2010, stratified by log-scale cumulative citation count. Solid shapes indicate mean, ribbons cover 95% of data. From left: papers in 1-10th percentile, papers in 45-55th percentile, papers in the top 90th-99th percentile.

Sensitivity to Paper Impact. The number of citations received by scientific papers varies widely. To evaluate HINTS' ability to accurately predict citations for papers of varying impact, we stratify AMiner papers into three groups based on their ground truth log-scale cumulative citation count five years after publications: lowly cited papers (bottom decile), moderately cited papers (45-55th percentiles), and highly-cited papers (90-99th percentiles). Fig. 6 shows the average predicted time series for all papers within each of these groups compared with their corresponding average ground truth trajectory. HINTS seems to over-predict low-citation papers over time. This is likely because our parametric generator is designed to model the trajectories of cited papers, while many of these papers receive no citations. In future work, this could be addressed with

a zero-inflation parameter. However, HINTS performs remarkably well for highly-cited papers and shows tolerable error for medium-cited papers. The strong performance on very high impact papers suggests HINTS could be useful for both scientists and funders to spot "hidden gems" in the scientific literature.

4.3 How HINTS Works

In this part, we perform a series of detailed analyses to better understand the performance of HINTS. We first compare how the algorithm uses metadata differently across fields. Next, we explore the imputed embeddings and learned citation time series parameters through visualization.

Importance of Metadata Types in Imputation. Not all metadata contain equal information for citation prediction. To understand how HINTS uses different types of metadata, we normalize the learned imputation weights using a softmax function (Table. 2). Unsurprisingly, we find that each reference contributes comparatively little information, while author, venue and keywords are more important predictors of citation time series in both Computer Science and Physics. However, these three factors play different roles between CS than Physics. The distinction possibly reflects differences in how these two communities operate (e.g., perhaps Physics communities converge more strongly on papers in top journals).

Table 2: Learned weights of metadata for imputation.

Field	Reference	Author	Venue	Keywords
AMiner (CS)	0.181	0.243	0.281	0.295
APS (Physics)	0.191	0.260	0.312	0.237

Visualization of Imputed Embeddings. To confirm that our imputed, temporally-encoded embeddings contribute to prediction, we randomly sampled 1000 papers from each strata described in 4.2 from AMiner, i.e., 1-10th percentile, 45th-55th percentile and 90-99th percentile (3000 papers in total). We use t-SNE [21] to project embeddings into a two-dimensional space (Fig. 7). Each point represents a paper, which is colored by its log-scale 5-year cumulative citation count after publication.

The embeddings clearly capture information about cumulative citation counts, as evidenced by the gradient from blue points (left-top) to red points (right-bottom). However, the gradient is not perfect; consistent with Fig. 6, the bottom 10% of papers is widely scattered, and some are mixed with the top 10%. This overlap shows that two papers with the same metadata can still have vastly different outcomes due to differences in quality or chance preferential attachment. Although metadata are strongly predictive of citation, they can't capture everything that contributes to citation for new papers.

Interpretation of Parameters in Time Series Generator. Modeled after the parameters in [38], we expect our three citation parameters "fitness" η, "peak time" μ and "rate of decay" σ described in Sec. 3.4 to capture different aspects of the citation process. Notably Fig. 8 shows a strong correlation between "fitness" η and the cumulative citation counts. Furthermore, highly-cited papers have

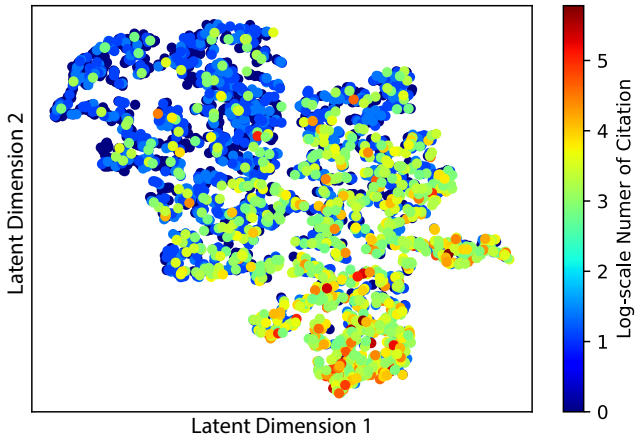

Figure 7: 2D t-SNE projections of imputed embeddings from AMiner sample. Embeddings color is coded by log-scale five-year cumulative citation count: blue means lower citations while red means higher.

a larger σ, indicating their longer survival time due to preferential attachment. The interpretability of these parameters reinforces our conclusion from the ablation analysis with HINTS-Seq: leveraging domain-specific knowledge is crucial for accurate time series prediction.

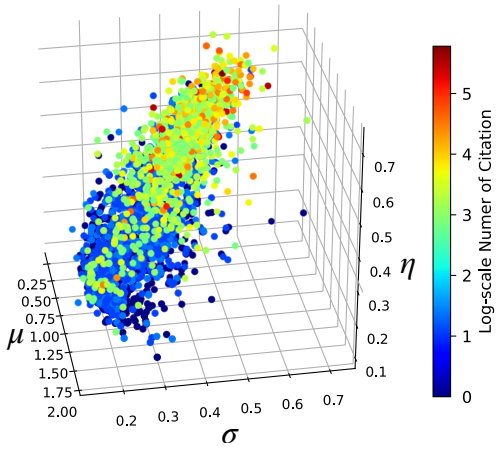

Figure 8: AMiner papers with respect to citation function parameters η, μ, σ. Papers are colored by log-scale five-year cumulative citation count: blue means lower citations while red means higher.

5 RELATED WORK

We review three lines of related work: citation time series prediction, academic recommender systems, and heterogeneous information network embedding.

5.1 Citation Time Series Prediction

Citation time series capture a scientific publication's impact or popularity over time. Most existing approaches focus on extracting citation patterns from a paper's early leading citation values after publication. Methods fall into two categories. Parametric approaches make explicit assumptions about a paper's citation pattern [15, 20, 27, 38, 42]. For example, we build on the log-normal intensity function proposed by Wang *et al.* to model each individual paper's citation pattern. In a subsequent work, Liu *et al.* capture the "recency effect" in citation time series by proposing a time-aware term. The second group of papers do machine learning on leading citation values with limited domain-specific assumptions [1, 47]. Unlike HINTS, all of these approaches depend on leading citation values, which are not available for new papers.

To model a new paper, several feature engineering techniques [3, 9, 43, 44, 46] have been proposed. For example, Dong *et al.* propose to represent a paper using the following features: author, topic, references, venue, social network, and temporal attributes. However, feature engineering takes significant human labor and may discard useful information. Compared with these approaches, HINTS automatically encodes temporal and relational cues from a paper's metadata in the context of a DHIN for time series prediction. To our knowledge, this is the first work to convert dynamic network signals into citation time series.

5.2 Academic Recommender Systems

Another line of research closely related to our work is graph-based paper recommendation, which aims to retrieve the most relevant literature with respect to a reader's query. Similar to citation prediction, these approaches seek to assess the importance and trending popularity of papers to make useful recommendations. Inspired by webpage search, [24, 37, 45] conduct PageRank on citation networks and introduce time-aware regularizers to reduce the recency bias that original PageRank has against recent publications. To model the temporal order of edges, [12, 18] propose time-aware centrality metrics to capture the dynamic nature of citation networks. However, these works fail to consider the heterogeneous interactions between metadata. Similar to us, Wang *et al.* [40] use dynamic heterogeneous information networks to learn the "saliency" of a paper using a reinforcement learning paradigm. They find that citation momentum (i.e., preferential attachment) contributes significantly more to citation than static metadata.

HINTS differs from these works in two aspects. First, we are not only concerned with identifying similar and/or high impact papers, but also predicting their citations into the future. Second, post-publication temporal dynamics available to recommender systems (e.g., citation momentum [40]), are not available for "cold-start" citation prediction. Instead, we propose to predict temporal dynamics using imputed pre-publication embedding trajectories.

5.3 Heterogeneous Network Embedding

Our work is also related to heterogeneous information network (HIN) embedding, which learns low-dimension vectors for nodes in HINs. To capture the multiple types of nodes and relations, path-based HIN embedding methods preserve the network structure through pre-defined meta-paths [32]. For example, [7] designs meta-paths on bibliographic networks and then use skip-gram to learn embeddings for nodes based on meta-paths. Another direction is based on matrix factorization [28], which decomposes a HIN into several simple sub-networks. These sub-networks are then individually processed and fused together. Recently, graph neural networks (GNNs) [17] have been used to learn embeddings for nodes through end-to-end frameworks. Relational-GCN [25] is an extension of GNN for heterogeneous graphs. Other relational GNN approaches have been proposed in recent years. [41, 48]. We extend R-GCN with a simple temporal alignment technique to learn embeddings for a DHIN.

6 DISCUSSION

Although HINTS is designed for citation time series prediction, it is noteworthy that HINTS can be generally applied to many other time series prediction tasks with limited modification. For example, the future in-links of a newly created website could be predicted with a web connected graph and a seasonal time series function. In addition, HINTS can also flexibly incorporate leading values if they exist. One potential way is that leading values can be another source used to learn the parameters η_p, μ_p and σ_p beyond the imputed embeddings.

Despite its effectiveness for predicting "cold start" time series, we note that HINTS also has limitations. By design, our model is not equipped to identify "sleeping beauties" that become impactful many years later (e.g., neural networks papers from the 1980s) [15]). We leave this "sleeping beauty" problem to future work.

7 CONCLUSION

In this paper, we tackle a new problem: citation time series prediction from the time of publication, *without leading citation values*. We propose a novel framework, HINTS, which transforms the historical and relational signals encoded in pre-publication dynamic bibliographical networks into predicted citation trajectories for new papers. More generally, HINTS demonstrates how relational and temporal information in DHINs can be combined with interpretable, domain-specific statistical models for effective citation prediction. The novelty of HINTS comes from the framework, and future work could substitute each of the three modules with other algorithms (e.g., GAT [36] for node embedding, point processes for parametric modeling) as needed to tackle other cold-start prediction problems. In extensive experiments, we demonstrate that HINTS is effective for citation prediction. We believe HINTS could be useful for scientists, granting organizations, and academic recommender systems seeking to identify "hidden gems" with high potential for future impact.

ACKNOWLEDGMENTS

This work is partially supported by NSF III-1705169, NSF CAREER Award 1741634, NSF # 1937599, DARPA HR00112090027, Okawa Foundation Grant, and Amazon Research Award. Bernard Koch is funded by an NSF Graduate Research Fellowship.

REFERENCES

[1] Ali Abrishami and Sadegh Aliakbary. 2019. Predicting citation counts based on deep neural network learning techniques. *Journal of Informetrics* 13, 2 (2019), 485–499.

[2] Qi Cao, Huawei Shen, Keting Cen, Wentao Ouyang, and Xueqi Cheng. 2017. Deephawkes: Bridging the gap between prediction and understanding of information cascades. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 1149–1158.

[3] Carlos Castillo, Debora Donato, and Aristides Gionis. 2007. Estimating number of citations using author reputation. In *International Symposium on String Processing and Information Retrieval*. Springer, 107–117.

[4] Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. 785–794.

[5] Ting Chen and Yizhou Sun. 2017. Task-guided and path-augmented heterogeneous network embedding for author identification. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. ACM, 295–304.

[6] Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *CoRR* abs/1412.3555 (2014). arXiv:1412.3555 http://arxiv.org/abs/1412.3555

[7] Yuxiao Dong, Nitesh V Chawla, and Ananthram Swami. 2017. metapath2vec: Scalable representation learning for heterogeneous networks. In *SIGKDD*. ACM.

[8] Yuxiao Dong, Reid A Johnson, and Nitesh V Chawla. 2015. Will this paper increase your h-index?: Scientific impact prediction. In *Proceedings of the eighth ACM international conference on web search and data mining*. ACM, 149–158.

[9] Yuxiao Dong, Reid A Johnson, and Nitesh V Chawla. 2016. Can scientific impact be predicted? *IEEE Transactions on Big Data* 2, 1 (2016), 18–30.

[10] Lun Du, Yun Wang, Guojie Song, Zhicong Lu, and Junshan Wang. 2018. Dynamic Network Embedding: An Extended Approach for Skip-gram based Network Embedding.. In *IJCAI*. 2086–2092.

[11] James A Evans and Jacob Reimer. 2009. Open access and global participation in science. *Science* 323, 5917 (2009), 1025–1025.

[12] Rumi Ghosh, Tsung-Ting Kuo, Chun-Nan Hsu, Shou-De Lin, and Kristina Lerman. 2011. Time-aware ranking in dynamic citation networks. In *2011 IEEE 11th International Conference on Data Mining Workshops*. IEEE, 373–380.

[13] Huan Gui, Qi Zhu, Liyuan Liu, Aston Zhang, and Jiawei Han. 2018. Expert finding in heterogeneous bibliographic networks with locally-trained embeddings. *arXiv preprint arXiv:1803.03370* (2018).

[14] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*. 1024–1034.

[15] Qing Ke, Emilio Ferrara, Filippo Radicchi, and Alessandro Flammini. 2015. Defining and identifying sleeping beauties in science. *Proceedings of the National Academy of Sciences* 112, 24 (2015), 7426–7431.

[16] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.).

[17] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

[18] Kristina Lerman, Rumi Ghosh, and Jeon Hyung Kang. 2010. Centrality metric for dynamic networks. In *Proceedings of the Eighth Workshop on Mining and Learning with Graphs*. 70–77.

[19] Cheng Li, Jiaqi Ma, Xiaoxiao Guo, and Qiaozhu Mei. 2017. Deepcas: An end-to-end predictor of information cascades. In *Proceedings of the 26th international conference on World Wide Web*. 577–586.

[20] 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 *AAAI*. 1438–1444.

[21] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, Nov (2008), 2579–2605.

[22] JN Manjunatha, KR Sivaramakrishnan, Raghavendra Kumar Pandey, and M Narasimha Murthy. 2003. Citation prediction using time series approach kdd cup 2003 (task 1). *ACM SIGKDD Explorations Newsletter* 5, 2 (2003), 152–153.

[23] Yao Qin, Dongjin Song, Haifeng Cheng, Wei Cheng, Guofei Jiang, and Garrison W. Cottrell. 2017. A Dual-Stage Attention-Based Recurrent Neural Network for Time Series Prediction. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence* (Melbourne, Australia) *(IJCAI'17)*. AAAI Press, 2627–2633.

[24] Hassan Sayyadi and Lise Getoor. 2009. Futurerank: Ranking scientific articles by predicting their future pagerank. In *Proceedings of the 2009 SIAM International Conference on Data Mining*. SIAM, 533–544.

[25] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*. Springer, 593–607.

[26] Jingbo Shang, Jialu Liu, Meng Jiang, Xiang Ren, Clare R Voss, and Jiawei Han. 2018. Automated phrase mining from massive text corpora. *IEEE Transactions on Knowledge and Data Engineering* 30, 10 (2018), 1825–1837.

[27] Huawei Shen, Dashun Wang, Chaoming Song, and Albert-László Barabási. 2014. Modeling and predicting popularity dynamics via reinforced poisson processes. In *Twenty-eighth AAAI conference on artificial intelligence*.

[28] Chuan Shi, Binbin Hu, Xin Zhao, and Philip Yu. 2018. Heterogeneous Information Network Embedding for Recommendation. *IEEE Transactions on Knowledge and Data Engineering* (2018).

[29] Roberta Sinatra, Dashun Wang, Pierre Deville, Chaoming Song, and Albert-László Barabási. 2016. Quantifying the evolution of individual scientific impact. *Science* 354, 6312 (2016), aaf5239.

[30] Yizhou Sun, Rick Barber, Manish Gupta, Charu C Aggarwal, and Jiawei Han. 2011. Co-author relationship prediction in heterogeneous bibliographic networks. In *2011 International Conference on Advances in Social Networks Analysis and Mining*. IEEE, 121–128.

[31] Yizhou Sun and Jiawei Han. 2013. Mining heterogeneous information networks: a structural analysis approach. *Acm Sigkdd Explorations Newsletter* 14, 2 (2013).

[32] Yizhou Sun, Jiawei Han, Xifeng Yan, Philip S Yu, and Tianyi Wu. 2011. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. *Proceedings of the VLDB Endowment* 4, 11 (2011), 992–1003.

[33] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*.

[34] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. 2008. Arnetminer: Extraction and Mining of Academic Social Networks. In *KDD'08*. 990–998.

[35] Brian Uzzi, Satyam Mukherjee, Michael Stringer, and Ben Jones. 2013. Atypical combinations and scientific impact. *Science* 342, 6157 (2013), 468–472.

[36] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. *International Conference on Learning Representations*.

[37] Dylan Walker, Huafeng Xie, Koon-Kiu Yan, and Sergei Maslov. 2007. Ranking scientific publications using a model of network traffic. *Journal of Statistical Mechanics: Theory and Experiment* 2007, 06 (2007), P06010.

[38] Dashun Wang, Chaoming Song, and Albert-László Barabási. 2013. Quantifying long-term scientific impact. *Science* 342, 6154 (2013), 127–132.

[39] Jingyuan Wang, Ze Wang, Jianfeng Li, and Junjie Wu. 2018. Multilevel wavelet decomposition network for interpretable time series analysis. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2437–2446.

[40] Kuansan Wang, Zhihong Shen, Chi-Yuan Huang, Chieh-Han Wu, Darrin Eide, Yuxiao Dong, Junjie Qian, Anshul Kanakia, Alvin Chen, and Richard Rogahn. 2019. A review of Microsoft academic services for science of science studies. *Frontiers in Big Data* 2 (2019), 45.

[41] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. 2019. Heterogeneous graph attention network. In *The World Wide Web Conference*. 2022–2032.

[42] Shuai Xiao, Junchi Yan, Changsheng Li, Bo Jin, Xiangfeng Wang, Xiaokang Yang, Stephen M. Chu, and Hongyuan Zhu. 2016. On Modeling and Predicting Individual Paper Citation Count over Time. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence* (New York, New York, USA) *(IJCAI'16)*.

[43] 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*. ACM, 51–60.

[44] 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*. ACM, 1247–1252.

[45] Philip S Yu, Xin Li, and Bing Liu. 2004. On the temporal dimension of search. In *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*. 448–449.

[46] Tian Yu, Guang Yu, Peng-Yu Li, and Liang Wang. 2014. Citation impact prediction for scientific papers using stepwise regression analysis. *Scientometrics* 101, 2 (2014), 1233–1252.

[47] Sha Yuan, Jie Tang, Yu Zhang, Yifan Wang, and Tong Xiao. 2018. Modeling and predicting citation count via recurrent neural network with long short-term memory. *arXiv preprint arXiv:1811.02129* (2018).

[48] Chuxu Zhang, Dongjin Song, Chao Huang, Ananthram Swami, and Nitesh V Chawla. 2019. Heterogeneous graph neural network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 793–803.

[49] Ling Zhao, Yujiao Song, Chao Zhang, Yu Liu, Pu Wang, Tao Lin, Min Deng, and Haifeng Li. 2019. T-gcn: A temporal graph convolutional network for traffic prediction. *IEEE Transactions on Intelligent Transportation Systems* 21, 9 (2019), 3848–3858.