Knowledge-Based Systems 238 (2022) 107839

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Heterogeneous dynamical academic network for learning scientific impact propagation

Xovee Xu [a], Ting Zhong [a], Ce Li [a], Goce Trajcevski [b], Fan Zhou [a,*]

[a] University of Electronic Science and Technology of China (UESTC), Chengdu, Sichuan 610054, China
[b] Iowa State University, Ames, IA 50011, USA

ARTICLE INFO

Article history:
Received 29 June 2021
Received in revised form 24 November 2021
Accepted 27 November 2021
Available online 12 December 2021

Keywords:
Scientific impact prediction
Heterogeneous information network
Graph neural network
Information diffusion
Science of science

ABSTRACT

Quantifying and predicting the long-term impact of both scientific papers and individual authors have important implications for many academic policy decisions, from identifying emerging trends to assessing the merits of proposals for potential funding. This paper presents SI-HDGNN, a novel heterogeneous dynamical graph neural network that explicitly models a *heterogeneous*, *weighted*, *directed* and *attributed* academic graph, enabling a prediction of the cumulative scientific impact of papers and authors by a specifically designed aggregation method. Unlike the existing feature-based or homogeneous approaches, SI-HDGNN addresses the problem by capturing the temporal–structural characteristics of the papers and authors as well as their complex interactions and long-term dependencies. Extensive experiments conducted on three large-scale multidisciplinary academic datasets demonstrate its superior performance in predicting the long-term scientific impact of both scientific papers and authors.

1. Introduction

The quantity and rate of scientific research publications have experienced a tremendous increase in recent years as can be witnessed, for example, by the number of records in DBLP[1] which has doubled from 2,486,800 in 2013 to 5,401,295 in 2020. Similarly, according to the AI index report 2021 [1], the number of peer-reviewed AI publications grew by nearly 12 times between 2000 and 2019. Quantifying the impact of articles, journals, conferences, institutions and individual researchers is an important task in many domains pertaining not only to the advancement of science but also to society in a broader sense [2]. For example, funding agencies and research institutes need to not only have a deep understanding of the current research developments but also assess the potential (short- and) long-term impacts. Moreover, identifying frontier ideas and breakthrough topics should go hand-in-hand with identifying productive scholars and improving the search for well-fitted scientists for defined projects, as well as refining the policies for hiring (and awarding) high-quality faculty and also for broader decision-making [3–6].

The availability of various scientific databases, such as Web of Science, Google Scholar, DBLP, ScienceDirect, IEEE *Xplore*, and ACM DL provides an unprecedented opportunity to explore the career of scientists and the dynamics of the evolving process of paper dissemination. However, the scientific impacts of papers and authors can be affected by a variety of factors, e.g., a prolific researcher may publish a number of papers every year, but the impact of their publications can vary significantly over time [7]. Additionally, some scientific findings may receive a burst of attention immediately, while others may take decades to become impactful [8].

Quantifying and foreseeing the impact of scientific diffusion has been of interest to generations of researchers since the pioneering work in [9]. Earlier efforts [2,10–12] primarily focused on extracting indicative features, designing effective stochastic processes, and discovering the latent mechanisms that drive the accumulation of citations. Scholar metrics, such as the number of publications and citations, have been widely used to forecast the future *h*-index [11]. Factors such as topical authority and publication venue that may increase citations were utilized to predict the scientific impact [13]. Temporal and structural features of authors/publications – e.g., growth rate, recency, node degrees, betweenness, community, etc. – have also been used to improve the prediction performance [10,14]. Despite their various merits, the existing works have limitations in predicting the impact of scientific publications due to the confluence of different, and sometimes, controversial factors [8,15] and the difficulty of generalizing the knowledge from one discipline to another. Conversely, some implicit but essential factors have not

* Corresponding author.
E-mail addresses: xovee@ieee.org (X. Xu), zhongting@uestc.edu.cn (T. Zhong), ce.lc@outlook.com (C. Li), gocet25@iastate.edu (G. Trajcevski), fan.zhou@uestc.edu.cn (F. Zhou).

[1] https://dblp.uni-trier.de/statistics/recordsindblp.

https://doi.org/10.1016/j.knosys.2021.107839
0950-7051/

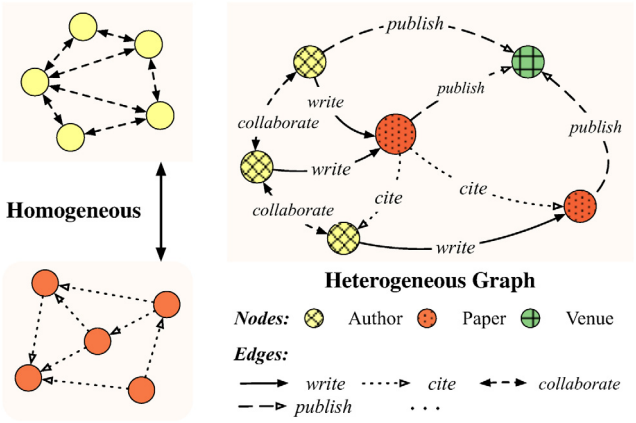

Fig. 1. Illustration of homogeneous and heterogeneous academic networks.

been fully leveraged – e.g., the academic authorities that amplify author/paper exposure and facilitates grant funding.

Another line of research predicts the propagation of scientific impact from the perspective of stochastic dynamics, relying on various pattern recognition-based models [16–18]. These methods are theoretically solid and have demonstrated their innovativeness, particularly for interpretability of the predictions — however, they require longer sequences of observations [15] and may still fail to fully integrate the rich semantic features and explore the complex interactions among authors and papers for scientific impact prediction.

Recent applications of deep neural networks on graph data have inspired numerous models for capturing the temporal and sequential process of information diffusion [19], in which academic graphs (e.g., citation networks) are modeled for scientific impact prediction. An end-to-end graph embedding-based prediction model was proposed in [20], which learns the representation of information cascade graphs with random walk paths, diffusion processes via recurrent neural networks (RNNs) [21], and attention mechanisms. CasCN [22] exploits the structure of each citation cascade through a recurrent graph convolutional network (GCN) [23], and predicts the future size by taking the directionality of the citation graph and time decay effects into account. However, these deep learning approaches deal with the representation learning of homogeneous graphs, limiting their capability to exploit the information associated with multiple node attributes and complex relationships in heterogeneous graph structures. For example, in Fig. 1, a homogeneous graph consists of only one type of node and edge, while the heterogeneous graph can handle multiple types and further encompass semantic information representing the nodes and their interactions. We note that, for compactness, in Fig. 1 (as well as in Fig. 2), we use doubly arrowed edges to substitute a pair of edges between the adjacent nodes in the directed graph.

At the heart of the motivation for this study is the observation that properly incorporating all the meaningful relationships between various "nodes" and "edges" when capturing the diffusion of scientific results and their impact is still not fully addressed. Although the existing methods in scientific impact prediction have gained certain success, they face several important challenges: (1) Feature-based models explore various kinds of features extracted from content, structure, time-series, and metadata from papers, authors, and venues. Nevertheless, they rely on extensive hand-crafted feature engineering that cannot be generalized from one domain to another and are not easy to implement [24] (2) Pattern-based models make strong assumptions about the underlying diffusion mechanisms in the academic graph, which is inapplicable to large-scale data that are full of uncertainties. Also, they lack the guide of future popularity and thus perform poorly [18]. (3) Deep learning-based approaches usually lack effective heterogeneous graph modeling and thus fail to capture the complex dependencies and dynamic relations between different types of entities, limiting their prediction performance [22,25]. On the other hand, traditional HIN-based models are short of learning temporal dynamics between citations [26].

In this paper, we propose **SI-HDGNN**, an end-to-end prediction model that quantifies long-term Scientific Impacts via a Heterogeneous Dynamical Graph Neural Network. It studies the dynamic evolving process of scientific impact while capturing the rich structures and semantics embedded in large-scale heterogeneous academic graphs. SI-HDGNN bridges the gap between dynamical GNNs [27,28] and heterogeneous information network (HIN) embedding [29–32], which has largely been studied independently in prior works. SI-HDGNN learns node representations with a newly designed heterogeneous GNN that aggregates the neighboring features of nodes with a fast weighted contextualized node sampling strategy. In addition, SI-HDGNN is a temporal-attentive representation network, preserving the unevenly distributed scientific impact of nodes. It also captures the dynamic evolution of nodes and the temporal dependencies among heterogeneous entities by encoding temporal cascading information into node representations, which sheds light on the underlying mechanism that accumulates the impact for both papers and authors.

The main contributions of this work are threefold:

- We study the scientific impact prediction problem and present a novel heterogeneous dynamical graph learning framework, which allows us to capture richer and more complex interactions between nodes and edges in a *heterogeneous*, *weighted*, *directed* and *attributed* academic graph, without extensive feature engineering and special designs.
- The proposed SI-HDGNN model extends traditional HIN-based models with a temporal horizon and efficiently acquires knowledge from large-scale academic networks. It learns the temporal aspects of nodes' structural and semantic properties and combines them for scientific impact prediction. Furthermore, we specifically design a temporal aggregation module for effective author prediction by splitting citations in different author publications.
- We conduct extensive evaluations on three large-scale multidisciplinary academic datasets with millions of nodes and edges. The experimental results on two scientific impact prediction tasks show that the proposed model is general across domains and achieves significant improvements over homogeneous graph-based information diffusion models and state-of-the-art HIN approaches.

For reproducibility, the source code and datasets used are publicly available at https://github.com/celi52/si-hdgnn.

The rest of this paper is organized as follows. Section 2 reviews the related literature and positions the contributions of SI-HDGNN in that context. Section 3 introduces the preliminaries of the scientific impact prediction problem and the necessary background of heterogeneous academic graph. In Section 4, we present the details of our solution for scientific impact prediction. We report the experimental results, ablation study and qualitative results, in Section 5. We conclude our work and point out future directions in Section 6.

2. Related work

In this section, we review the up-to-date literature and discuss their relations to our proposed SI-HDGNN model.

2.1. Scientific impact prediction

The goal of scientific impact prediction is to quantify the potential influence of academic publications, institutions and scholars (e.g., the number of citations, *h*-index of authors [33], impact factors of venues [34]). Predicting the long-term impact of individuals can be challenging since many factors/covariates influence the final scientific impact, even when they have similar initial developments [17].

Feature engineering methods explore various kinds of related features with respect to authors/papers/venues. Some examples are:

- *temporal features*, such as the publication date and citation growth speed.
- *structural features*, such as node degree, page-rank [35], betweenness/closeness centralities, importance score and communities [14] in the academic network; content features extracted from both linguistic and visual sources such as paper title, abstract, figures/tables, author biographies.
- *metadata* of the scholars/publications including age, gender, collaborators, number of published papers, number of author citations, *h*-index, research disciplines/interests, etc.

Certain complex features have also been designed to improve the prediction performance such as latent Dirichlet allocation (LDA) for topic modeling [36], as well as popularity, novelty, diversity, links & weights [37,38] and authority of papers [39]. Once the feature set is defined and extracted, they are fed into discriminative machine learning algorithms for training and evaluation by the model. Such models include but are not limited to linear regression, support vector machine (SVM), multi-layer perceptron (MLP), XGBoost [40], etc. Feature-based models – with carefully selected feature groups – generally perform well and are easy to implement and interpret. Nevertheless, selecting and designing good-performing features can be tricky, and heavy feature engineering and expert domain knowledge are often required. In addition, features designed for one scenario may not be transferable to another, as generalization capability is lacking. Our SI-HDGNN is an end-to-end data-driven model that directly learns heterogeneous structures and dynamic impact evolution in a large-scale academic graph for scientific impact prediction.

Another set of methods explores the underlying mechanisms that drive scientific publications to disseminate and harvest citations. Statistical methods and stochastic point processes have been introduced to model the arrival process of citations, such as reinforced Poisson processes [16], self-exciting Hawkes processes [41–43], and their combinations [17,18,44–46]. These methods treat the prediction process in a generative way by first observing a small group of early adopters and then simulating the diffusion process using deterministic stochastic models. In addition, this line of studies is built on a range of specified mechanisms governing information propagation, e.g., attractiveness of items, aging effect, second acts [2] and *rich-get-richer* phenomenon [16,19], which have been extensively adopted in modeling general information diffusion such as microblogs [42,47] and scientific publications [17]. Although such methods have gained success in certain contexts, they are often limited to the determined processes and hard to integrate with the internal/external factors that influence the final scientific impact. Most importantly, they lack flexibility and generalizability that are desirable for large-scale scientific data learning. In contrast, SI-HDGNN is a general graph-based framework without a special propagation mechanism assumption. Therefore, various semantic features (e.g., texts and figures) can be easily and simultaneously utilized to learn expressive representations of nodes in the graph with *any* state-of-the-art representation learning models.

Recently, studies have used various deep learning techniques for modeling and forecasting scientific impact. Such models do not require special assumptions regarding paper diffusion mechanisms and can be trained in an end-to-end manner. Existing deep models can be classified into three main categories: (i) Most approaches address the content of scientific articles such as titles, abstracts, keywords, and reviews [48,49] by applying natural language processing (NLP) techniques including LDA [36], word2vec [50], and transformers [51]. These studies rely on the information available at the time of publication and the characteristics of individual articles. (ii) Researchers have also studied the post-publication information, e.g., early citation count, citation sequence and citation graph. Temporal and structural data can be well characterized by deep learning approaches such as recurrent neural networks and graph neural networks [25,52–55]. Notably, these models resort to peeking the early evolving trend of an article's citations for its future impact prediction. (iii) Historical information such as published articles, past success, collaborator, and communities can be vital for forecasting the future. Such information can be used to construct a global academic network. Taking DeepCas [20] as an example, the global network contains citation relationships (edges) between researchers (nodes). Since the global structures imply the influence, reputation and/or preference of nodes in the graph, they are beneficial to scientific impact prediction.

SI-HDGNN can be seen as a novel framework consisting of the aforementioned three main elements: (i) it uses paper texts as content features; (ii) it includes an RNN-based temporal aggregation module for citation sequence modeling; and (iii) the global graph is modeled by specifically tailored heterogeneous dynamical graph neural networks. Although previously proposed graph-based deep models have adopted graph attention networks [56], temporal graph neural networks [57], or graph convolution networks [22] for graph learning and information cascade modeling, most of them (if not all) only consider homogeneous graphs and/or ignore content features.

2.2. Heterogeneous graph representation learning

The homogeneous network (graph) assumes that the type of node or edge is unique, e.g., the author collaboration network and the friendship network contain only one type of nodes and edges. Due to this distinct pattern, for complex network interactions, most homogeneous information networks simply ignore the heterogeneity of nodes and edges or convert different types of nodes and edges to the same type (cf. Fig. 1). To fuse complex relationships/interactions, which are implied in heterogeneous graphs, many heterogeneous information network (HIN) embedding models have been proposed in recent years. Learning expressive graph representations is essential for HIN downstream tasks. Interested readers are referred to comprehensive HIN surveys [58–62].

In general, heterogeneous graph representation learning can be categorized as proximity-preserving and message-passing algorithms. Proximity-preserving methods are largely dependent on network topology engineering, for which paradigms such as random walks [63] and first/second-order proximity [64] have been employed [65,66]. For example, the nodes traversed by meta-path guided random walks are used in metapath2vec [65] to model the context of nodes in the heterogeneous graph. HIN2vec [66] directly considers meta-paths as objects/contexts to learn the embeddings for both nodes and meta-paths. Various HIN studies exploit heterogeneous contexts for learning representations by meta-paths and often work with a two-step strategy. First, they perform random walks on the heterogeneous graph to collect node contexts. Then, the skip-gram algorithm is usually

Table 1
Major symbols and definitions.

Symbol	Definition
P, A, V	Entity class of paper, author and venue
p, a, v	Instance of paper, author and venue
N, M	# of papers and authors in the dataset
t_0, t_{ob}, t_{pd}	Timestamp of first publication, observation, and prediction
$p_{i,C}(t_0, t_k)$, $a_{i,C}(t_0, t_k)$	Sorted citation set of paper p_i and author a_i over time interval $[t_o, t_k]$, the Subscript C denotes Citation set
$c_{p_i}^{t_k}$, $c_{a_i}^{t_k}$	Cardinality of $p_{i,C}(t_0, t_k)$ and $a_{i,C}(t_0, t_k)$ (the number of citations for p_i and a_i from time t_0 to t_k)
$\mathcal{G}_{AP} = (\mathcal{V}, \mathcal{E}, \mathcal{A}, \mathcal{R}, \mathbf{C}_{\mathcal{V}})$	Academic heterogeneous graph $\mathcal{G}_{AP}$ with nodeset $\mathcal{V}$, edgeset $\mathcal{E}$, Node type set $\mathcal{A}$, edge type set $\mathcal{R}$, and node features $\mathbf{C}_{\mathcal{V}}$
$\mathcal{A}_n$, $\mathcal{R}_e$	Type of node for $n \in \mathcal{V}$, and type of edge for $e \in \mathcal{E}$
$\mathcal{N}(n)$	Set of node n's neighbors
$\mathcal{S}_p(n)$, $\mathcal{S}_a(n)$, $\mathcal{S}_v(n)$	Sampled n's neighbors for three types: paper, author, and venue

utilized to generate the node embedding by predicting its context nodes given the meta-paths.

Motivated by the success of graph neural networks (GNNs), many researchers have recently adapted them to facilitate the message-passing process in heterogeneous graphs [29,67–71]. A heterogeneous personalized spacey random walk and a scalable HIN embedding algorithm (SpaceyMetapath) to attain the expected stationary distribution among nodes were presented in [67]. HAN [68] leverages node-level attention and semantic-level attention to discriminate the importance of nodes and meta-paths. HetGNN [29] exploits representation learning from both graph structural heterogeneity and node content heterogeneity. Inspired by transformer [51], node- and edge-type dependent parameters are designed to characterize the heterogeneous attention over each edge in HGT [69]. An investigation of heterogeneous representation by jointly performing graph structure learning and GNN parameter learning was presented in [70]. Unlike the aforementioned HIN models, which are only applicable for static graphs, our proposed SI-HDGNN is a dynamic GNN model, capable of capturing the evolving structures of heterogeneous graphs. In addition, SI-HDGNN includes a fast weighted contextualized node sampling strategy and an attentive representation network, which can learn better heterogeneous knowledge in a large-scale academic network and facilitate subsequent temporal citation sequence learning.

In sum, notwithstanding the existing (works and) results on scientific impact prediction, dynamic graph learning, and heterogeneous graph neural networks — to our knowledge, few studies have addressed the scientific impact prediction problem by modeling heterogeneous dynamic graph neural networks.

3. Preliminaries

We now introduce the basic terminology and formally define the problem(s) addressed in this paper. We note that, for convenience, a list of symbols introduced (and used) throughout the paper, and their concise definitions, are provided in Table 1.

We consider three basic entity classes, an example of which is provided next. We note that here, we first describe the attributes of the instances of each entity class in a broader manner, for the sake of developing intuition. A detailed list of the (sources of the) data items and attributes is provided in Section 5.

- *Publications*: $P = \{p_1, p_2, \ldots, p_{|P|}\}$, where P is the entity class of publication, and $|P|$ represents the number of publication instances. Each instance $p_i \in P$ consists of attributes such as *ID*, *paper title*, *authors*, *venue*, *year*, reference, etc. We note that whenever there is no ambiguity, we will use "publications" and "papers" interchangeably. The attribute *ID* is a unique identifier (e.g., similar to the DOI number used in bibliographic nomenclature).
- *Authors*: $A = \{a_1, a_2, \ldots, a_{|A|}\}$, where A is the entity class of the author, and $|A|$ represents the number of author instances. Each instance $a_i \in A$ also has its collection of attributes such as *ID*, *name*, *affiliation*, *title*, etc. The attribute ID is, once again, a unique identifier for each author, similar to the ORCID (Open Research and Contributor ID[2]) number.
- *Venues*: $V = \{v_1, v_2, \ldots, v_{|V|}\}$, where V is the entity class of venue, and $|V|$ represents the number of venue instances. We assume a unique ID such as ISSN (International Standard Serial Number) or DOI, along with additional attributes such as *type* (e.g., "conference" or "journal"), *name*, *year*, *month*, *volume*, etc. We note that, depending on the *type*, some values may be missing – e.g., when *type* = "conference", the proceedings will not have values for *volume*, *number*, etc.

To assess the value of a particular attribute for a given object, we use the standard "." notation. For example, the publication year of a particular paper is denoted as $p_i.year$. We reiterate that the details of the actual attributes used in the evaluation are provided in Section 5 – and we note that not all the possible sources have the same set of attributes (values). However, we assume, in the rest of this paper, that it is always possible to disambiguate two instances (e.g., if there is no ORCID value, then *(name, affiliation)* can serve for disambiguation among authors).

What is relevant at this point is that, given three such sets (i.e., entity classes), we proceed with constructing the *heterogeneous graph of academic publications* $\mathcal{G}_{AP} = (\mathcal{V}, \mathcal{E})$ (cf. Fig. 1). To capture the heterogeneity, a given graph $\mathcal{G}_{AP}$ is also associated with mappings ($\mathcal{A}$ and $\mathcal{R}$) where:

1. $\mathcal{A}$ denotes the *type of vertex* for the members of $\mathcal{V}$. Specifically, for each $n \in \mathcal{V}$, its type $\mathcal{A}_n$ can be (exclusively) either the `'author'`, `'paper'`, or `'venue'`.
2. $\mathcal{R}$ denotes the *types of directed edges* between vertices, which are dependent on the (types of) adjacent vertices that they are connected. Specifically, we consider seven different kinds of edges:
   – author *writes* paper,
   – author *collaborates* with author,
   – author *publishes in* venue,
   – author *cites* paper,
   – paper *is published in* venue,
   – paper *cites* paper, and
   – paper *cites* author(s)

We can see that each type in $\mathcal{R}$ denotes a particular kind of "social relationship" in the world of academic publications. Similar to actual social networks, there can be different correlation behaviors between (pairs of) members. As a specific example, in

[2] https://orcid.org/.

information cascades [19], a given user may have a tendency to retweet items from another particular user more often. Similar phenomena are possible in our "social network" represented by $\mathcal{G}_{AP}$; a particular author may cite a particular paper multiple times (> 1) in his publications; a particular paper may cite multiple papers (> 1) of the same author; etc. To properly capture this differential affinity, we take the frequency of each behavior into consideration. Specifically, for the directed, weighted, and type-aware edge $e = (n_i, n_j, \mathcal{R}_e)$ between node n_i and n_j where $e \in \mathcal{E}$, and edge type $\mathcal{R}_e \in \{collaborates, publishes, cites\}$ the attribute *weight* of e, denoted $W(e)$, reflects the amount of times that a distinct instance of such an edge has occurred. We implicitly assume that whenever there is a single occurrence of (an instance of) a particular edge, its weight is 1.

Last, given $\mathcal{G}_{AP}$ and its associated $\mathcal{A}$ and $\mathcal{R}$, we use $\mathbf{C}_{\mathcal{V}}$ to denote the vector of *features* of the vertices. In addition to internal features (e.g., title, year), we include (a subset of the) graph topological structure for vertices. For each n of a type `'paper'` (i.e., $\mathcal{A}_n$ = `'paper'`), we use a pre-trained BERT model and bert-as-service [72,73] to obtain the representation of its title. DeepWalk is leveraged to obtain the pre-trained node structural embeddings of `'author'`, `'paper'` and `'venue'` types of vertices in $\mathcal{G}_{AP}$. For compactness, when appropriate, we will denote $\mathcal{G}_{AP}$ as a quintuple $(\mathcal{V}, \mathcal{E}, \mathcal{A}, \mathcal{R}, \mathbf{C}_{\mathcal{V}})$.

For each paper (resp. author), let t_0 indicate the time (i.e., year) of its respective first publication. Clearly, for a given set of N papers, during an observation window $[t_0, t_{ob}]$, each paper can be cited by multiple other papers, and at different time instances. Thus, for a given paper p_i, and a time instance t_k ($t_0 \leq t_k$), the sorted set of its citations over $[t_0, t_k]$ can be represented as a collection of pairs $p_{i,C}(t_0, t_k) = \{(p_j, t) | p_j \ cites \ p_i \in \mathcal{E}$, at time t ($t_0 \leq t \leq t_k$)$\}$. Then, we use $c_{p_i}^{t_k}$ to denote the cardinality of $p_{i,C}(t_0, t_k)$ – i.e., the number of citations for p_i by other papers from time t_0 to t_k.

In a similar spirit, we can use $c_{a_i}^{t_k}$ to denote the number of citations that author a_i received by various papers at time instance t_k. We note that this value is further discussed in Section 4.3.

We now define the *scientific impact prediction* for papers and authors, respectively, as follows:

Definition 1 (*Scientific impact prediction for papers (SIPP)*)**.** For a given paper p_i and the corresponding collection of citations between the publication time t_0 and a given observation time t_{ob} (i.e., $p_{i,C}(t_0, t_{ob})$) for p_i, the *scientific impact prediction* problem of p_i aims to provide a (predicted) value of its total number of citations $c_{p_i}^{t_{pd}}$ at some future prediction timestamp $t_{pd} (> t_{ob})$.

Definition 2 (*Scientific impact prediction for authors (SIPA)*)**.** For a given author a_i and the corresponding collection of papers co-authored by a_i between time t_0 (its first publication) and a given observation time t_{ob}, the *scientific impact prediction* problem for a_i aims to provide a (predicted) value of the total number of cited articles $c_{a_i}^{t_{pd}}$ co-authored by a_i at some (future) prediction-timestamp $t_{pd} (> t_{ob})$.

Then, assuming that the exact number of citations at time t_{pd} is $c_{p_i}^{t_{pd}}$, the SIPP problem (cf. Definition 1) can be solved by optimizing the following mean squared logarithmic error (MSLE),

$$\mathcal{L}^{\langle SIPP \rangle} = \frac{1}{N} \sum_{i=1}^{N} \left(\log \hat{c}_{p_i}^{t_{pd}} - \log c_{p_i}^{t_{pd}} \right)^2, \tag{1}$$

where $\hat{c}_{p_i}^{t_{pd}}$ is the predicted number of citations, $c_{p_i}^{t_{pd}}$ is the ground truth, and N is the number of papers. Here, we use the logarithm to make the loss function care only about the relative difference between the true and predicted value.

In a similar spirit, the SIPA problem (cf. Definition 2) can be solved by

$$\mathcal{L}^{\langle SIPA \rangle} = \frac{1}{M} \sum_{i=1}^{M} \left(\log \hat{c}_{a_i}^{t_{pd}} - \log c_{a_i}^{t_{pd}} \right)^2, \tag{2}$$

where $\hat{c}_{a_i}^{t_{pd}}$ is the predicted number of citations for author a_i, $c_{a_i}^{t_{pd}}$ is the ground truth, and M is the number of authors.

4. Model

In this section, we present the details of our SI-HDGNN model. From a global perspective, it consists of two main building blocks: (i) heterogeneous graph representation learning via deep neural networks and (ii) temporal citation sequence modeling and author aggregation via recurrent neural networks.

4.1. Heterogeneous graph building and representation learning

Fig. 2 shows the first part of SI-HDGNN, which is used to learn heterogeneous representations of nodes in academic graph $\mathcal{G}_{AP}$. Specifically, for a node in $\mathcal{G}_{AP}$ – given its heterogeneous neighbors in a non-Euclidean graph structure – we learn a low-dimensional node embedding, in which the learned embedding preserves heterogeneous neighboring proximity in a continuous space. Toward that, we use a random walk with restart [74] and a deep neural network architecture from [29] as the backbone. We design a heterogeneous neighboring node sampling strategy and a multi-head attention based neighboring node aggregation module to enhance heterogeneous node representation learning.

4.1.1. Heterogeneous neighboring node sampling

The main aspects of traditional random walk-based models are that they: (i) depend on homogeneous citation cascade graphs [20,22], which ignores rich interactions among heterogeneous neighbors; (ii) neglect to consider node impact in the context of multiple weighted node/edge relations [29]; or (iii) heavily rely on user-specified meta-paths [6,65]. Given a node n (paper/author/venue) in academic graph $\mathcal{G}_{AP}$, the distribution of its neighboring nodes is often highly skewed, i.e., some nodes connect to a large number of other nodes (those highly cited papers/authors, prestigious venues, etc.). However, most of them have only a few neighbors, intensely following the heavy-tailed distribution of citations [3,10]. We note that modeling of multiple types of nodes/edges and their complex interactions was often underexplored in previous studies.

To accommodate these factors, we design a weighted contextualized node selection strategy, which is more suitable for capturing the scientific impact and imbalanced distribution of nodes in a heterogeneous academic graph. Specifically, for each current step, a given node n either returns to the previous node with probability q, or jumps to the next neighbor node with probability $1 - q$. Let $\mathcal{N}(n)$ be the set of n's neighbors, then node n has a probability $1 - q$ to select one of its neighbors $\mathcal{N}(n)$. The neighboring environment of node n contains multiple node types $\mathcal{A}$, multiple edge types $\mathcal{R}$ and different node/edge characteristics. To take all these factors into consideration and ensure that each type of neighbor for the target node could be chosen, we design a type-based node sampling strategy – the probability of walking to the next node m from n – defined as

$$\Pr(m|\mathcal{N}(n), \mathcal{G}_{AP}) = \begin{cases} (1-q)\alpha D^{\alpha}(n,m), & \text{if } \mathcal{A}_m \text{ is paper} \\ (1-q)\beta D^{\beta}(n,m), & \text{if } \mathcal{A}_m \text{ is author} \\ (1-q)\gamma D^{\gamma}(n,m), & \text{if } \mathcal{A}_m \text{ is venue} \end{cases} \tag{3}$$

where α, β, γ are the weight parameters that define the probability when `'paper'`, `'author'`, and `'venue'` are chosen as

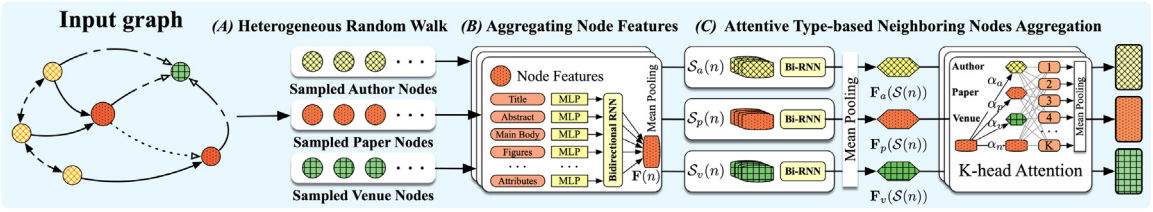

Fig. 2. The architecture of heterogeneous representation learning. ***(A)***: walk heterogeneous nodes using a specifically designed weighted contextualized node selection strategy based on random walk with restart; ***(B)***: aggregate multi-modal node features with Bi-RNNs aggregator; and ***(C)***: aggregating heterogeneous neighbors of notes with type-based multi-head attention mechanism.

the next node. $D^{\alpha}(*)$, $D^{\beta}(*)$, $D^{\gamma}(*)$ are functions measuring node influence from various factors. In this study, we use the node in-degree to represent the importance of a node in its neighborhood and we combine it with the weight of the corresponding edge e. The function $D^{\alpha}(n, m)$ is defined as:

$$D^{\alpha}(n,m) = \frac{\sum_{e_i \in \mathcal{E}_{n,m}} W(e_i) \log(f_{in-degree}(m) + 1)}{\sum_{v_j \in \mathcal{N}^{\alpha}(n)} \sum_{e_j \in \mathcal{E}^{\alpha}_{n,v_j}} W(e_j) \log(f_{in-degree}(v_j) + 1)} \tag{4}$$

where $\mathcal{E}_{n,m}$ is the set of edges from node n to m, and $f_{in-degree}(m)$ is the in-degree value of node m in the graph $\mathcal{G}_{AP}$. We use the value of the logarithm of the in-degree to avoid the weight being strongly dominated by high-in-degree nodes. We note that the factors are not limited to the in-degree and edge weights. Other features, e.g., arrival time, pagerank scores and similarities, can also participate in the function for measuring the neighborhood influence — which is deferred to our future work.

Through running random walks iteratively, we can sample a fixed number of nodes for each node type in $\mathcal{A}$, resulting in three sets for papers, authors, and venues denoted as $\mathcal{S}_p(n)$, $\mathcal{S}_a(n)$, and $\mathcal{S}_v(n)$. Note that we consider edge directions, weights, and node degrees when sampling heterogeneous neighbors. By doing so, the most representative neighbors are sampled simultaneously with respect to multiple important factors designed by the influence function $D(*)$.

4.1.2. Aggregating node features

After sampling the neighbors for each node, we utilize bidirectional gated recurrent units (Bi-GRUs) [21] to capture the dependencies among the nodes' content features. Assuming that there are k content features for one specific type of node, the feature aggregation can be formalized as

$$\mathbf{F}(n) = \frac{1}{k} \sum_{i=1}^{k} \left(\overrightarrow{\text{GRU}}(\mathbf{h}_n^i) \,||\, \overleftarrow{\text{GRU}}(\mathbf{h}_n^i) \right), \tag{5}$$

$$\mathbf{h}_n^i = \text{MLP}(\mathbf{C}_n^i), \text{ for } i = 1, 2, \ldots, k, \tag{6}$$

where $\mathbf{F}(n) \in \mathbb{R}^{d_n}$ is the aggregated embedding of node n computed by mean pooling; $\|$ denotes the concatenation operation; $\mathbf{C}_n^i$ is the ith type of content feature of node n; $\mathbf{h}_n^i \in \mathbb{R}^{d_{\mathbf{h}}}$ is the output of the MLP. In practical applications, various content features can be used here to enhance the model's learning ability – e.g., meta-data and the text of papers (title, abstract, main body), illustrations (figure, table), past publications of authors/venues, metadata of authors/venues (profile, honor, research area, collaborators), etc. The bidirectional recurrent neural networks used here serve as a content feature aggregator and, as shown in the experiments (cf. Table 4 in Section 5), have superior performance compared to other aggregators, such as concatenation and max/sum pooling.

4.1.3. Aggregating heterogeneous neighbors

After aggregating the node content features, for each node n in graph $\mathcal{G}_{AP}$, we have its corresponding aggregated features $\mathbf{F}(n)$. Then we are ready to use a type-based RNN to aggregate embeddings of the neighbors in $\mathcal{S}(n)$. For each node type in $\mathcal{A}$ (in our case, the paper, author, and venue), $\mathcal{S}_{p/a/v}(n)$ is the homogeneous type-specific neighboring set of a given node n, and $\text{RNN}_{p/a/v}$ is a type-specific aggregator. More specifically, SI-HDGNN utilizes another Bi-GRU for modeling n's neighbors. We take the paper neighboring aggregation as an example,

$$\mathbf{F}_p(\mathcal{S}_p(n)) = \frac{\sum_{i=1}^{|\mathcal{S}_p(n)|} \left(\overrightarrow{\text{GRU}}(\mathbf{F}(i)) \,\|\, \overleftarrow{\text{GRU}}(\mathbf{F}(i)) \right)}{|\mathcal{S}_p(n)|}, \tag{7}$$

where $|\mathcal{S}_p(n)|$ is the number of node n's paper-type neighbors, $\mathbf{F}_p(\mathcal{S}_p(n)) \in \mathbb{R}^{d_s}$ is the output embedding from the homogeneous neighboring set $\mathcal{S}_p(n)$, and d_s is the dimension of the aggregated neighboring embeddings of node n.

In SI-HDGNN we use deterministic neural networks, bidirectional RNNs, and mean pooling as aggregators of the node content along with node neighbors. Alternatively, other types of aggregators can also be used [29,75], e.g., the last hidden state of RNNs, CNNs, max or sum pooling.

4.1.4. Multi-head attention for type-based neighbors

With each type-based neighboring aggregators in hand, we can combine them using multi-head attention mechanism [76] to learn the importance of each type-based neighbors,

$$\alpha_i = \frac{\exp\left(\text{LeakyReLU}(u^T[\mathbf{F}(n) \,\|\, \mathbf{F}_i^{\mathcal{S}}])\right)}{\sum_{j \in \mathcal{S}'(n)} \exp\left(\text{LeakyReLU}(u^T[\mathbf{F}(n) \,\|\, \mathbf{F}_j^{\mathcal{S}}])\right)}, \tag{8}$$

$$\mathcal{S}'(n) = \mathbf{F}(n) \cup \{\mathbf{F}_j^{\mathcal{S}}\}_{j \in \mathcal{S}(n)}, \tag{9}$$

$$E(n) = \frac{1}{K} \sum_{i=1}^{K} \sum_{\mathbf{F}_i(n) \in \mathcal{S}'(n)} \alpha_i \mathbf{F}_i(n), \tag{10}$$

where $E(n) \in \mathbb{R}^{d_E}$ is the learned embedding of node n, LeakyReLU is the activation function, exp is the exponential function, u is the attention parameter, and K is the number of attention heads. Here, $\mathbf{F}(n)$ and $\mathbf{F}_j^{\mathcal{S}} = \mathbf{F}_{\mathcal{A}_j}(\mathcal{S}_{\mathcal{A}_j})$ are computed by Eqs. (5) and (7), respectively.

4.2. Scientific impact prediction for papers

The second part of SI-HDGNN aims to model the cascading behavior of papers/authors. Here, we consider each paper p_i as an independent entity. Recall that t_0 is the first publication time, t_{ob} is the observation time, and $p_{i,C}(t_0, t_{ob}) = \{(p_j, t) | p_j \textit{ cites } p_i \in \mathcal{E}, \text{at time } t \ (t_0 \le t \le t_{ob})\}$ is the sorted set of p_i's citations published by time during the observation window $[t_0, t_{ob}]$. Since we already obtained the embeddings of papers $E(p)$, authors $E(a)$, and venues $E(v)$ (cf. Eq. (10)), we now separately model authors of a paper and the paper itself by feeding them into RNNs. Fig. 3 shows the process of scientific impact prediction for papers.

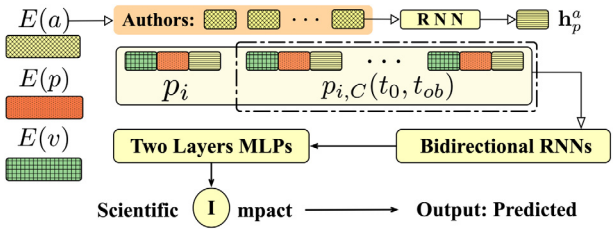

Fig. 3. Temporal prediction module for paper scientific impact. $E(a), E(p), E(v)$ are embeddings of authors, papers, and venues, respectively. We first utilize a vanilla RNN to aggregate the author embeddings and use the last hidden state as the final author embedding $\mathbf{h}_p^a$. Then we concatenate author, paper, venue embeddings as the representation of paper p. The sequence of citation papers as well as the original paper are fed into bidirectional RNNs for final scientific impact prediction.

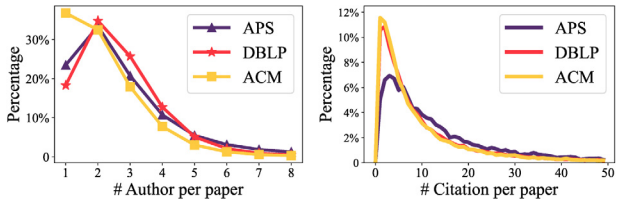

Fig. 4. Percentile of (left) number of authors per paper and (right) number of citations per paper in APS, DBLP, and ACM datasets.

4.2.1. Multi-author aggregation layer

Note that each citation paper as well as the original paper may contain multiple authors. For example, in the APS dataset the mean number of authors per paper is 3.438, and the maximum number is 25 — and the distribution of author and citation quantity per paper is shown in Fig. 4. We sequentially pipeline the author embeddings into a GRU and then use the last hidden state $\mathbf{h}_p^a$ as the representation of p's authors.

4.2.2. Sequential citation aggregation layer

After author aggregation, for each paper p, we have its own embedding $E(p)$, the corresponding venue embedding $E(v)$, and the aggregated author embedding $E(\mathbf{a}) = \mathbf{h}_p^a$. We then use a two-layer Bi-GRU to sequentially aggregate the citing papers ordered by their publishing time t, where each citing paper p_j is modeled as the combination (by concatenation) of paper, authors, and venue embeddings. The rationalè is that we expect to capture temporal dependencies among citing papers in both the forward and backward directions, which, as we will show in Section 5.6, is superior to other aggregators such as sum or max pooling [75]. The overall architecture of the citation aggregation is

$$\mathbf{E}(p_j) = (E(p_j) \| E(\mathbf{a}_j) \| E(v_j)), \tag{11}$$

$$\mathbf{h}_j^1 = (\overrightarrow{\text{GRU}}(\mathbf{E}(p_j)) \parallel \overleftarrow{\text{GRU}}(\mathbf{E}(p_j))), \tag{12}$$

$$\mathbf{h}_j^2 = (\overrightarrow{\text{GRU}}(\mathbf{h}_j^1) \parallel \overleftarrow{\text{GRU}}(\mathbf{h}_j^1)), \tag{13}$$

where $\mathbf{h}_j^2 \in \mathbb{R}^{d_{\mathbf{h}^2}}$ is the jth hidden state of the second layer of Bi-GRU. Here, we concatenate the last hidden states of Bi-GRU as the final output representation of paper p_j and then make use of it to predict its future scientific impact.

4.3. Scientific impact prediction for authors

We now turn to the scenario of predicting the scientific impact for authors. One straightforward way is to rank all citations of one author (i.e., $a_{i,C}(t_0, t_{ob}) = \{(p_j, t) | p_j \text{ cites } a_i \in \mathcal{E}, \text{at time } t \ (t_0 \le t \le t_{ob})\}$) during the observation window into a long sequence ordered by citing time and then directly feed the citation sequence into RNNs, just as we did in the paper prediction. The simple module is expected to learn the temporal dependencies among the citations of all the authors' observed publications. However, this assumption can be problematic since all the citations during the observation window may come from distinct publications (e.g., the author may have several different research interests). These citations may not be highly correlated. Another obstacle is that a single sequence of citations is too long to efficiently capture the long-term temporal dependencies due to the vanishing gradient problem of RNNs [77].

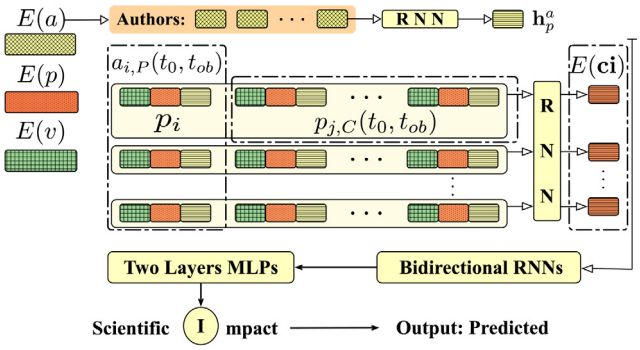

Fig. 5. Temporal prediction module for author scientific impact. Different from paper prediction which only needs one long citation sequence, we designed a new approach for author prediction by splitting the citation sequence into multiple citation paths grouped by author publications $a_{i,P}(t_0, t_{ob})$. $E(a), E(p), E(v)$ are embeddings of authors, papers, and venues, respectively. First, the author sequence of each citing paper are fed into a vanilla RNN to obtain the author embedding h_p^a. Then for each publication p_j of author during observation window, $p_{j,C}(t_0, t_{ob})$ is the citation sequence of p_j. A citation-level RNN is used to aggregate each publication p_j's citations. A career-level Bi-RNN is used to aggregate all of the observed publication impacts of author for final scientific impact prediction.

To address this, we opt to split citations into multiple citation paths grouped by author publications. We note that t_0 is the publication time of author a's first publication, and t_{ob} is the observation time. And let $a_{i,P}(t_0, t_{ob}) = \{(p_j, t) | a_i \text{ writes } p_j \in \mathcal{E}, \text{at time } t \ (t_0 \le t \le t_{ob})\}$ be the set of a_i's publication set ordered by publication time t during the observation window $[t_0, t_{ob}]$. The architecture of the "scientific impact prediction for authors" framework is shown in Fig. 5. We discuss different author prediction approaches and compare their performances in Section 5.7.

4.3.1. Multi-author and sequential citation aggregation

Note that each paper p may contain multiple authors. We sequentially pipeline authors' embeddings into a GRU and then use the last hidden state $\mathbf{h}_p^a$ as the representation of their embedding. Then, for each paper, we have its own paper embedding $E(p)$, the corresponding venue embedding $E(v)$, and the aggregated author embedding $E(\mathbf{a}) = \mathbf{h}_p^a$. We concatenate the three types of embeddings to represent this paper,

$$\mathbf{E}(p) = (E(p) \| E(\mathbf{a}) \| E(v)). \tag{14}$$

For each publication p_j, we can obtain its citation set $p_{j,C}(t_0, t_{ob})$ during a_i's observation window. Then, we use a citation-level RNN to sequentially aggregate the citations of paper p_j ordered by their publishing time t and utilize the last hidden state $\mathbf{h}_{p_j}$ as the representation of p_j's citation papers.

4.3.2. Sequential publication aggregation

After multi-author and citation aggregation, for each author's publication p_j, we have its citation embedding $E(p_j) = \mathbf{h}_{p_j}$. We then use a two-layer Bi-GRU network to sequentially aggregate the publication papers ordered by their publication time

t. We expect to capture temporal dependencies among author's publications,

$$\mathbf{h}_j^1 = (\overrightarrow{\mathrm{GRU}}(E(p_j)) \parallel \overleftarrow{\mathrm{GRU}}(E(p_j))), \tag{15}$$

$$\mathbf{h}_j^2 = (\overrightarrow{\mathrm{GRU}}(h_j^1) \parallel \overleftarrow{\mathrm{GRU}}(h_j^1)), \tag{16}$$

where $\mathbf{h}_j^2 \in \mathbb{R}^{d_{\mathbf{h}^2}}$ is the jth hidden state of the second layer of Bi-GRU. Here we concatenate the last hidden states of Bi-GRU as the final output representation of the author and then use it to make the author scientific impact prediction.

Compared to the paper prediction approach, which aggregates all the citations in a single sequence, author prediction considers individual impacts of the author's publications and then aggregates all the publication impacts using Bi-RNNs.

4.4. Objective

For unsupervised heterogeneous graph representation learning, we optimize the model weights without any node labels through negative sampling [78]:

$$\begin{aligned}\mathcal{L}_1(\Theta_1) = &\sum_{\langle u,v,v'\rangle \in S_{triple}} \log \sigma(E(u)\cdot E(v)) \\ &+ \log \sigma(-E(u)\cdot E(v')),\end{aligned} \tag{17}$$

where S_{triple} is the set of triplets $\langle u, v, v'\rangle$, $\sigma(\cdot)$ is the sigmoid function and $E(\cdot)$ is the learned node embedding. We choose the positive node v for the target node u by its context information through heterogeneous neighboring node sampling. For the negative node v' of each v, we randomly sample it from the whole node set $\mathcal{V}$ with node type $\mathcal{A}(v)$. The objective is to pull the positive node v close to the target node u while pushing negative node v' away from u. Then, the close-far pattern is learned with the discriminator, and the neighboring information of node u is preserved in the embedding space.

Algorithm 1: Heterogeneous Graph representation learning

input : preprocessed nodes features $C_{\mathcal{V}}$,
triplets set S_{triple} sampled from heterogeneous neighboring random walk and negative sampling
output: node embedding $E(n)$,
optimized model Θ_1

for *epoch = 1 ... max_epoch* **do**
  for *batch in* S_{triple} **do**
    Aggregate node content features for all nodes $n \in \mathcal{V}$ with Bi-GRUs via Eq. (5);
    Aggregate the heterogeneous neighbors $S_{p/a/v}$ for (u, v, v') Eq. (7);
    Multiple types aggregation with attention by Eq. (10);
    Obtain $(E(u), E(v), E(v'))$;
    Update model parameters Θ_1 by Eq. (17);
  end
end

Algorithm 1 summarizes the procedure of heterogeneous graph representation learning.

For *SIPP*, the output of SI-HDGNN is the predicted citation count $c_p^{t_{pd}}$ of paper p at prediction time t_{pd}. We use two-layer of MLPs as the impact predictor. Thus, the impact prediction is defined as

$$\mathcal{L}_2^{\langle SIPP\rangle}(\Theta_2) = \frac{1}{N_T}\sum_{i=1}^{N_T}(\log \hat{c}_{p_i}^{t_{pd}} - \log c_{p_i}^{t_{pd}})^2, \tag{18}$$

where N_T is the number of training samples, and $\hat{c}_{p_i}^{t_{pd}}$ is the predicted number of citations of paper p_i.

For *SIPA*, the general training process is similar, except that Eq. (18) is alternatively defined as: $\mathcal{L}_3^{\langle SIPA\rangle}(\Theta_3) = \frac{1}{M_T}\sum_{i=1}^{M_T}(\log \hat{c}_{a_i}^{t_{pd}} - \log c_{a_i}^{t_{pd}})^2$, where $\hat{c}_{a_i}^{t_{pd}}$ is the predicted number of citations for author a_i and M_T is the number of training samples.

Algorithm 2 outlines impact prediction procedure.

Algorithm 2: Impact prediction

input : Node embedding $E(a), E(p)$ *and* $E(v)$,
observed p_cite_seq $p_{i,C}(t_0, t_{ob})$,
observed a_pub_seq $a_{i,P}(t_0, t_{ob})$,
observer pub_cite_seq $p_{j,C}(t_0, t_{ob})$,
t_{pd} years citation groundtruth $c_{p/a}^{t_{pd}}$
output: Predicted impact $\hat{c}_{p/a}^{t_{pd}}$

==========Paper Prediction==========
while *not converged* **do**
  for p_j *in* $p_{i,C}(t_0, t_{ob})$ **do**
    Calculate $\mathbf{h}_{p_j}^a$ for p_j with GRU;
    Combine a/p/v embedding $E(p_j) = (E(p_j)||E(a_j)||E(v_j))$;
  end
  Pipeline citing papers' embedding sequence into Bi-GRU via Eqs. (12) and (13);
  Feed $\mathbf{h}^2$ into two-layer MLPs for prediction;
  Update the parameters Θ_2
end
==========Author Prediction==========
while *not converged* **do**
  for p_j *in* $a_{i,P}(t_0, t_{ob})$ **do**
    for p_k *in* $p_{j,C}(t_0, t_{ob})$ **do**
      Calculate $\mathbf{h}_{p_k}^a$ for p_k with GRU;
      Combine a/p/v embedding $E(p_k) = (E(p_k)||(E(a_k)||E(v_k)))$;
    end
    Pipeline citing papers into Bi-GRU;
    Obtain publication embedding $E(p_j)$;
  end
  Pipeline publication embedding sequence into Bi-GRU via Eqs. (15) and (16);
  Feed $\mathbf{h}^2$ into two-layer MLPs for prediction;
  Update the parameters Θ_3;
end

4.5. Complexity analysis

We close this section with an analysis of the complexity of SI-HDGNN and a brief quantitative overview.

First, we note that from the perspective of static time complexity analysis, SI-HDGNN consists of two main parts: (a) heterogeneous neighboring node sampling and (b) node representation learning and scientific impact prediction.

• *Complexity for heterogeneous neighboring node sampling.* The first part is the transformation of random walk with restart. Its complexity is polynomial with the number of nodes in the neighborhood of each node, which requires $\mathcal{O}(|\mathcal{V}|^3)$ time complexity – $|\mathcal{V}|$ denotes the number of nodes in the heterogeneous graph. This procedure generates the contextual information of the sampled nodes and is saved for the heterogeneous representation learning.

• *Complexity for node representation learning and scientific impact prediction.* The computational complexity of the bidirectional network is $\mathcal{O}(2*(4IH+4H^2+3H+HK))$, where I is the number of inputs, K is the number of outputs, and H is the number of cells in the hidden layer. The time complexity for the recurrent neural network per weight is $\mathcal{O}(1)$.

Second, from a quantitative perspective: we obtain the total number of weights by summing the number of elements for every parameter group with the supported TensorFlow API. Then SI-HDGNN has ∼604 K parameters for the representation learning

Table 2
Statistics of three multi-disciplinary academic datasets.

Dataset	APS	DBLP	ACM
# nodes	483, 294	1, 920, 499	2, 266, 733
# edges	26, 828, 252	71, 850, 966	39, 480, 147
Heterogeneous nodes in graph:			
# papers	290, 836	1, 161, 820	1, 187, 613
# authors	192, 448	755, 084	887, 974
# venues	10	3, 595	191, 146
Heterogeneous edges in graph:			
author *writes* paper	856, 206	3, 133, 355	2, 602, 722
author *collaborates* with author	1, 875, 081	5, 085, 689	3, 981, 389
author *publishes* in venue	276, 744	2, 092, 398	2, 178, 002
author *cites* paper	5, 796, 482	13, 637, 387	6, 459, 257
paper *cites* paper	2, 482, 448	6, 757, 586	3, 087, 459
paper *cites* author	5, 824, 483	15, 292, 583	6, 646, 654
paper *published* in venue	290, 836	1, 161, 820	1, 187, 613
Selected citation sequence in dataset:			
# paper citation sequence	2, 874	19, 591	7, 754
# author citation sequence	2, 320	12, 093	6, 517

Table 3
Detailed information for one paper entity.

Items	Description	Example
#∗	Paper title	#∗ Information geometry of U-Boost and Bregman divergence
#@	Authors	#@Noboru Murata,Takashi Takenouchi,Takafumi Kanamori,Shinto Eguchi
#*t*	Year	#*t* 2004
#*c*	Publication venue	#*c* Neural Computation
#*index*	Paper index id	#*index* 436405
#%	Paper reference list	[#%94584, #%282290, #%605546, #%620759, #%564877 . . .]

part and costs ∼980s for one epoch training. For paper prediction, the number of parameters Θ_2 is ∼627 K and it requires ∼15s for one epoch of training. Similarly, parameters Θ_3 for author prediction have ∼628 K and the time cost for an epoch is ∼140s.

5. Experiments

In this section, we report the extensive experiments that we conducted to assess the benefits of SI-HDGNN on scientific impact prediction for both papers and authors. We use three large-scale multi-disciplinary academic datasets – APS, DBLP, and ACM – for evaluation between our SI-HDGNN and several state-of-the-art baselines. We reiterate that the source code and datasets are publicly available at https://github.com/celi52/si-hdgnn

5.1. Dataset

The evaluations were performed on three large-scale publicly available datasets of academic publications/networks: APS, DBLP, and ACM. Their detailed statistics are shown in Table 2 and Fig. 4. It is worth pointing out that our approach is not constrained to any particular academic domains (e.g., APS for physics, DBLP and ACM for computer science) – i.e., we did not design domain-specific mechanisms in SI-HDGNN which, in turn, enables it to be easily extended to other fields such as medicine or biology.

- **APS** (American Physical Society) dataset is accessed at Jan 19, 2017.[3] The APS academic network contains over 616 K academic papers on 17 APS journals between 1893 and 2017.
- **DBLP** citation dataset [79]. The DBLP academic network (released at October 27, 2017 by AMiner[4]) contains over 3.6M academic papers on 3 K venues published between 1936 and 2017.
- **ACM** (Association for Computing Machinery) dataset is released at Jan 20, 2017 by AMiner. It contains over 2.3M academic papers on 3 K venues between 1936 and 2017.

For all three datasets, each paper is associated with the information for the index, title, authorship, reference and publication venue/conference and year. We present the detailed information for the paper entity in Table 3. The paper index can uniquely identify a paper item while authors and venues are authenticated by their names. This could yield ambiguity, as a name may be potentially shared among several other researchers. This could be resolved by relying on the ORCID number; however, as mentioned in Section 3, not all authors have an ORCID number and, to our knowledge, no other public dataset with such a property (unique ID for each other) is available.

We set the observation window length $|t_{ob} - t_0|$ to 2 years. We define 20, 10 and 10 as the prediction times t_{pd} for APS, DBLP and ACM, respectively. To avoid data leakage, we use APS data from 1886–1999 to build the model and obtain nodes embeddings for three types of nodes. Then, we collect citation relationships during the observation window $[t_0, t_{ob}]$ = 1997–1999 and leverage the node representation we learned to predict the citation number after 20 years in 2017. The observation time is 2006–2008 for DBLP and 2004–2006 for ACM.

We note that the papers/authors having fewer than 4 citations during the observation window are filtered out. The settings of predictions for authors are the same as those for papers. We used 50% of them for training, 25% for validation, and the remaining 25% for testing.

Dataset Analysis. We now show some additional statistics of APS papers/authors. Fig. 6 ***(A)***: For papers with more than 200 citations after 30 years since publication, most yield citations linearly with the years (the dashed line denotes mean values), with a tendency to have citations during early years. A few papers had a long hibernation period (∼10-20 years) and then enjoyed a citation burst, which indicates a "sleeping beauty" phenomenon [8]. Fig. 6 ***(B)*** and ***(C)*** shows the complementary cumulative distribution function (CCDF) of paper citations and

[3] https://journals.aps.org/datasets.
[4] https://www.aminer.org/citation.

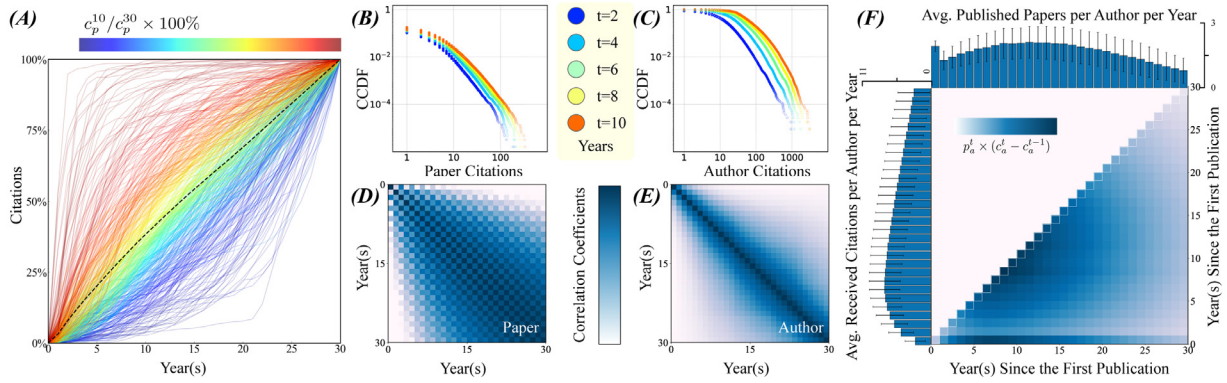

Fig. 6. APS dataset statistics: ***(A)***: we select 509 papers which citations more than 200 after 30 years since publication before 1987. Lines represent normalized citation growth trends, line colors indicate citation rank of papers at the tenth year, i.e., $c_a^{10}/c_a^{30} \times 100\%$; dashed line denotes the mean values. ***(B)*** and ***(C)***: Complementary cumulative distribution function (CCDF) of paper citations and author citations, respectively. ***(D)*** and ***(E)***: Pearson correlation coefficients of paper citations and author citations over 30 years, respectively. The (i, j) block in heatmap represents the correlation between ith year's and jth year's cumulative citations for all papers/authors. ***(F)***: The value of the ith diagonal block of the heatmap is $p_a^t \times (c_a^t - c_a^{t-1})$, i.e., the average number of papers each author published at the ith year multiplied by the average number of citations each author received at the ith year; Top histogram: average published papers per author per year, Left histogram: average citations authors received per year (errorbars are standard deviations).

author citations, respectively. We can see that lines go right over time (citations grow large). The paper citation generally follows the power-law distribution, while most of the author citations are larger than those of papers. Fig. 6 ***(D)*** and ***(E)*** show the Pearson correlation coefficients of paper/author citations over time. It is easy to see that the relationships between author citations in different years are less correlated compared to the paper citation-year relationship. This also explains why author scientific impact prediction is more difficult than paper prediction (cf. experimental results in Table 4). Fig. 6 ***(F)*** presents a 30 year history of average author publications and citations in the APS dataset. Two key factors are analyzed: (i) the number of papers published per year and (ii) the number of citations the author received per year. We can see that except for the first year, the average number of published papers increases until the 12th year and then decreases. The peak of citations received each year is around the 7th year. Taking the two factors together, the most productive career of researchers is around the 7th year to the 14th year.

5.2. Baselines

We consider several state-of-the-art scientific impact prediction baselines to demonstrate the effectiveness of our proposed SI-HDGNN model. Three feature engineering-based models, two point-process-based statistical models, three homogeneous graph-based deep learning models, and three heterogeneous graph representation learning based models are selected for comparison.

The following five baselines can be used directly for scientific impact prediction:

- **Uniform** – for all papers/authors, we always predict their impact as a fixed number, uniformly searched from the minimum $\log c_{p/a}^{t_{pd}}$ to maximum $\log c_{p/a}^{t_{pd}}$ with a step of 0.001.
- **Feature engineering based** – these models are popular for scientific impact prediction. Their performances are demonstrated as powerful or comparative to others when the selected features are quality. We carefully select structural and temporal features and feed them into two dense layers for prediction. Selected features include:
  - **Feature**-$c^{t_{ob}}$: we use observed citation $c^{t_{ob}}$ to predict the final citation $c^{t_{pd}}$ as a simple baseline.
  - **Feature-Structural**: features are extracted from the heterogeneous academic graph including original node in-degree, out-degree, number of neighbors, and page-rank, mean citation node in-degree and out-degree, 10th and 90th percentiles of citation node (both in- and out-) degree distribution, mean number of citation node neighbors, and mean page-rank of citation nodes.
  - **Feature-Temporal**: features are extracted from citation sequence including mean citing time, mean citing time for first half of citations, and mean citing time for last half of citations.
  - **Feature-All**: all temporal and structural features are combined for prediction.
- **DeepCas** [20] – is an end-to-end deep learning based prediction model, which utilizes multiple random walk processes for cascade path sampling and then uses bi-directional GRU and attention mechanism for predictions.
- **DeepHawkes** [18] – aims to bridge the gap between prediction and understanding of citation cascade prediction. Specifically, DeepHawkes proposes a deep information cascade learning model that combines Hawkes self-exciting point process by the following three components: user embedding, path encoding & pooling, and a non-parametric time decay function.
- **VaCas** [25] – is a deep probabilistic citation graph learning model which learns hierarchical citation-sequence representation and leverages variational auto-encoder for scientific knowledge diffusion learning.
- **CasFlow** [24] – is a hierarchical structure learning framework that learns citation-sequence uncertainties via variational inference and normalizing flows. It combines the local structure of citation graph with global author collaboration network (homogeneous) for performance improvement.

The following five baselines are used for homogeneous- or heterogeneous-aware node representation learning. We incorporate the learned representations into SI-HDGNN's specifically designed temporal aggregation module to predict the final scientific impact for both papers and authors. These baselines are indicated by a "+ T" suffix.

- **DeepWalk** [63]: which is a classic node representation learning method inspired by language model SkipGram. It employs truncated random walk to learn local information of nodes in graph.

Table 4
Performance comparison between SI-HDGNN and the baselines on three citation datasets. Note that bold values indicate better results than other configurations – the lower the MSLE and/or the higher the ACC, the better the performance.

Model	APS Paper		APS Author		DBLP Paper		DBLP Author		ACM Paper		ACM Author	
	MSLE	ACC	MSLE	ACC	MSLE	ACC	MSLE	ACC	MSLE	ACC	MSLE	ACC
Uniform	0.638	54.2	1.401	23.9	0.761	47.7	1.270	24.2	0.812	48.7	1.083	24.3
Feature-c^{t_r}	0.402	59.5	0.991	36.2	0.296	62.6	0.828	40.8	0.323	64.3	0.622	47.5
Feature-Structural	0.409	55.7	0.938	43.9	0.308	62.9	0.618	50.7	0.297	66.0	0.519	51.5
Feature-Temporal	0.398	59.5	0.995	37.0	0.285	62.5	0.789	42.1	0.275	65.7	0.581	48.6
Feature-All	0.381	55.4	0.881	42.9	0.267	65.2	0.562	51.1	0.255	65.3	0.456	55.3
DeepCas [20]	0.501	51.2	1.421	22.3	0.392	41.5	1.329	30.8	0.301	53.9	1.174	35.8
DeepHawkes [18]	0.442	55.3	1.382	36.7	0.325	45.7	1.269	35.7	0.280	58.8	1.085	40.4
VaCas [25]	0.406	60.0	1.402	30.5	0.281	64.6	1.274	32.4	0.259	67.5	1.084	37.3
CasFlow [24]	0.383	60.2	1.401	30.7	0.283	64.4	1.276	32.3	0.269	66.8	1.083	37.2
DeepWalk [63] + T	0.407	59.8	0.892	44.1	0.267	66.1	0.544	53.6	0.264	65.8	0.423	57.3
GraphSAGE [75] + T	0.402	58.6	0.896	41.5	0.269	65.7	0.539	53.3	0.259	66.1	0.405	57.7
ProNE [80] + T	0.427	57.4	0.914	43.2	0.604	50.7	0.604	50.7	0.285	64.7	0.453	57.6
metapath2vec [65] + T	0.409	60.9	0.880	**47.0**	0.261	66.6	0.552	52.1	0.267	67.1	0.434	56.3
HetGNN [29] + T	0.394	58.2	0.892	43.4	0.260	66.4	0.526	52.2	0.256	67.4	0.404	58.2
SI-HDGNN MLP	0.607	50.0	1.006	41.5	0.556	48.3	0.758	44.9	0.571	50.1	0.657	49.8
SI-HDGNN w/o Author	0.370	60.9	0.906	43.4	0.251	**67.8**	0.522	53.5	0.242	69.4	0.398	59.0
SI-HDGNN w/o Venue	0.389	60.8	0.897	42.4	0.250	67.4	0.526	53.3	0.257	67.5	0.407	59.0
SI-HDGNN-MaxPooling	0.501	57.7	1.021	43.7	0.437	58.2	0.601	52.1	0.438	60.1	0.456	59.9
SI-HDGNN-SumPooling	0.383	60.2	0.915	41.7	0.248	67.3	0.518	**54.4**	0.245	67.6	**0.383**	**61.1**
SI-HDGNN-Concat	0.391	59.2	0.889	41.0	0.253	66.4	**0.506**	53.4	0.239	67.7	0.396	60.1
SI-HDGNN	**0.366**	**63.0**	**0.860**	45.8	**0.245**	67.4	0.515	54.3	**0.239**	**70.3**	0.390	59.4

- **GraphSAGE** [75]: is an inductive node representation learning framework which leverages node features for generating node embeddings by neighbor sampling and aggregating algorithms.
- **ProNE** [80]: is a fast and scalable graph representation model. It adopts sparse matrix factorization for learning acceleration and spectral propagation for embedding enhancement.
- **metapath2vec** [65]: is a heterogeneous graph learning model, which designs a meta-path-guided random walk algorithm to sample heterogeneous neighbors.
- **HetGNN** [29]: learns heterogeneous node embeddings by aggregating type-based node features and neighboring nodes. It considers both structure heterogeneity and node content heterogeneity by using MLP and Bi-RNN aggregators.

In addition, we defined five variants of SI-HDGNN for the ablation study. To evaluate the quality of the graph representation, we directly feed the learned node embedding into two-layer MLPs as **SI-HDGNN MLP**. To evaluate the impact of author/venue embeddings, we separately remove the author part or venue part in Eq. (11) as two variants **SI-HDGNN w/o Author** and **SI-HDGNN w/o Venue**. To evaluate the effectiveness of citation aggregators, we use *max pooling*, *sum pooling*, or *concatenation* to substitute for the *RNN aggregator*. The resulting three variants are denoted as **SI-HDGNN-MaxPooling**, **SI-HDGNN-SumPooling**, and **SI-HDGNN-Concat**.

5.3. Metrics

We use two widely adopted evaluation metrics following previous work [16,18,42] – mean square logarithmic error (MSLE) and accuracy (ACC) – defined as

$$\text{MSLE} = \frac{1}{N_t}\sum_{i=1}^{N_t}(\log \hat{c}_{p_i}^{t_{pd}} - \log c_{p_i}^{t_{pd}})^2, \tag{19}$$

$$\text{ACC} = \frac{1}{N_t}\sum_{i=1}^{N_t}\mathbb{1}(0.5 * c_{p_i}^{t_{pd}} \le \hat{c}_{p_i}^{t_{pd}} \le 1.5 * c_{p_i}^{t_{pd}}), \tag{20}$$

where $\mathbb{1}(\cdot)$ is the indicator function, N_t is test sample size for the *SIPP* task, and $\hat{c}_{p_i}^{t_{pd}}$ is the predicted scientific impact of paper/author at prediction time t_{pd}.

5.4. Experimental settings

For all baseline models, whenever applicable, we set the embedding dimension to 128. The random walk restart probability q is 0.5, the walk length is 30, and the number of walks for each node is 10. For type-specific parameters $D^{\alpha}(*)$, $D^{\beta}(*)$, $D^{\gamma}(*)$, we use node in-degree and edge weights as a proxy of node influence.

For SI-HDGNN and its variants, the learning rate is chosen from $\{10^0, 10^{-1}, \ldots, 10^{-5}\}$. The length of the paper citation sequence of all methods is set to 100 – i.e., the maximum number of citation sequences. For papers with citation sequence lengths exceeding 100, we select their first 100 citations (for author aggregation, the length of RNN is set to 6). For author scientific impact prediction, we select at most 15 papers (and each of its first 100 citations) during the observation window. The numbers of units in the two-layer Bi-RNNs are set to 128 and 64, respectively. The number of units in two-layer MLPs are set to 64 and 32, respectively. For multi-head attention, we set the head number as 8. For content aggregation RNNs, we obtain 128 dimensional paper title embeddings pre-trained via BERT and bert-as-service [72,73], and node embeddings pre-trained via DeepWalk [63]. All the other hyper-parameters of baselines are set to their default values. Performance results are reported with early stopping on validation loss with 10 epochs of patience.

5.5. Prediction performance

We show the performance of all the models on three citation datasets in Table 4, and we discuss the main observations next.

Feature-based models obviously outperform the uniform model, which predicts the impact as a fixed number. This indicates that the features, e.g., citation number, academic graph structure and citation sequential information, are useful and play an important role in paper/author prediction. One interesting observation is that Feature-Temporal outdoes Feature-Structural on paper prediction, while Feature-Structural performs better in author prediction. This can be attributed to the fact that sequential

information reveals more citation pattern information that benefits paper predictions. Structural features, e.g., node in-degree, out-degree and number of neighbors, reflect the popularity of the author in the academic network, which plays a leading role in improving the author impact forecasting performance. Feature-All is the combination of temporal and structural features. It performs well and even surpasses the deep cascade method.

SI-HDGNN and its variants outperform all the other methods in both paper and author scientific impact prediction in all three datasets. This result demonstrates the effectiveness of learning interactions among heterogeneous nodes with the proposed heterogeneous information aggregation, which can be further verified by the fact that both feature-based models and homogeneous cascade prediction methods do not show a comparable performance. Previous deep learning-based prediction methods, e.g., DeepCas, DeepHawkes, VaCas and CasFlow, do not distinguish the type of nodes and therefore fail to model their complex and meaningful interactions. Specifically, compared with HetGNN+T, our proposed SI-HDGNN model decreases the paper prediction error MSLE by up to 7.1% and increases the prediction accuracy by up to 4.8%.

Author impact prediction is much harder than that of paper. As shown in ***(B-E)*** in Fig. 6, the citation number of authors is higher than that of papers by orders of magnitude, as well as the coefficients of correlation between observed and future citations. In fact, in settings of two-year observations, the proportion of average observed citations c_p^2 to c_p^{20} is approximately 9.1% for authors. In contrast, the proportion for papers is 34.6% (cf. ***(A)*** in Fig. 6), which explains why prediction for authors' impact is more difficult – i.e., largely due to insufficient observations and enormous variability in the authors' productivity [15] (cf. ***(F)*** in Fig. 6). In addition, the paper citation is strongly correlated to factors such as the citations a paper has gained and the importance of the publication venue (e.g., journal impact factor), which can be easily modeled in the graph with node attributes. Also, author impact is far more unstable due to implicit factors such as funding scheme, tenure, gender issues — all of which need to be quantified with external high-resolution data repositories. There are also discrepancies between paper prediction and author prediction in baselines. The performances of deep learning-based models drop severely on author prediction, and SI-HDGNN 's improvements over baselines are also larger on author prediction.

5.6. Ablation study

We now investigate the effect of essential modules in SI-HDGNN. First, as shown in Table 4, SI-HDGNN MLP exhibits the worst performance among all the variants, which emphasizes the importance of SI-HDGNN's specifically designed temporal aggregation module. The information aggregation mechanism used in SI-HDGNN is better than other graph embedding models, including two heterogeneous embedding methods, i.e., metapath2vec and HetGNN, because of the more complex relations considered in our model and the benefit of considering temporal dependencies between citation sequences and/or author sequences. For example, SI-HDGNN models 7 types of relations among nodes, whereas HetGNN only considers 3 edge types. Additionally, the venue plays a vital role in predicting the impact of an author or a paper. This is demonstrated by the significant performance degradation after removing venue embeddings in Eq. (11). Authorship, surprisingly, is less important than the journal that a paper published in, although masking the authorship information may slightly degrade the performance. For aggregation choices, max pooling and sum pooling are sometimes superior to the RNN aggregator used in SI-HDGNN due to their lack of sequential dependencies.

Table 5
MSLE of two author prediction approaches on two datasets.

Dataset	APS Author	DBLP Author	ACM Author
Cite-Seq	1.145	0.762	0.569
Cite-Pub-Seq	**0.860**	**0.515**	**0.390**

5.7. Discussion on author prediction

Although the observation (cf. Fig. 6) that the proportion of average observed citation c_p^2 to c_p^{20} for paper (34.6%) is higher than that of author (9.1%) could explain why the author prediction task is harder, we conjecture that this deficiency is caused by the mix of paper citations from different research disciplines. To better understand the inner drivers of the author's scientific impact and try to alleviate this deficiency, we designed two approaches for author scientific impact prediction and compared them in Table 5.

- **Cite-Seq.** Given an author a, we denote $\{(p_j, t_j)\}_j$ as the set of a's citations within observation window $[t_{pb}, t_{ob}](t_j \leq t_{ob})$. Then an RNN is used to sequentially aggregate the citation paper embeddings ordered by the publication time t_j, and use the last hidden state $\mathbf{h}_a^{ci}$ as the representation of author a's potential scientific impact.
- **Cite-Pub-Seq.** Different from **Cite-Seq**, which pipelines all author citations into one flat RNN, we adopt a hierarchical RNN network which predicts the impact in two steps: first predict individual publication impact via a citation-level RNN and then aggregate all the predicted publication impacts by utilizing a career-level Bi-RNN (cf. Fig. 5). We use this approach as default in this paper.

The prediction results on the two datasets are shown in Table 5. We can see that Cite-Pub-Seq approach effectively decreases the prediction errors on the two datasets (see Fig. 13).

5.8. Qualitative analysis

To make an intuitive evaluation of SI-HDGNN prediction performance. We show the comparison of the predicted value and the ground truth for *SIPP* and *SIPA* in Fig. 7. Black solid points are the true value, while empty circles in the gray color come to denote the predicted value. It is clear that major parts of the circles are located around the solid points for the results of DBLP and ACM, which indicates that SI-HDGNN performs better in DBLP and ACM.

Fig. 8 shows the prediction results on 6 representative journals — the lower the MSLE and/or the higher the ACC, the better the result. The performance of SI-HDGNN varies significantly on different publication venues — this is natural since the venue is a strong indicator for future impact accumulation. In addition, we found that the prediction accuracy is affected by the citation distribution of papers in a journal. For example, the standard deviation of 20 years of citations of papers (i.e., c_p^{20}) on Phys. Rev. Lett. is higher (61.64), whereas the value on Phys. Rev. A is a little less (45.30). This discrepancy also reveals why the prediction of papers on Phys. Rev. Lett. is more difficult.

In the same way, we collect 10 representative AI conferences in the DBLP dataset and 7 famous journals and conferences in the ACM dataset and plot the performance in terms of MSLE and ACC in Fig. 9 and Fig. 10, respectively.

Fig. 11 plots the latent space learned by SI-HDGNN in the APS dataset, where we can observe a clear clustering phenomena of author/paper embeddings from ***(a)*** and ***(c)***. It appears that papers published in the same journal tend to cluster together, which also indicates that publication venue is an important indicator

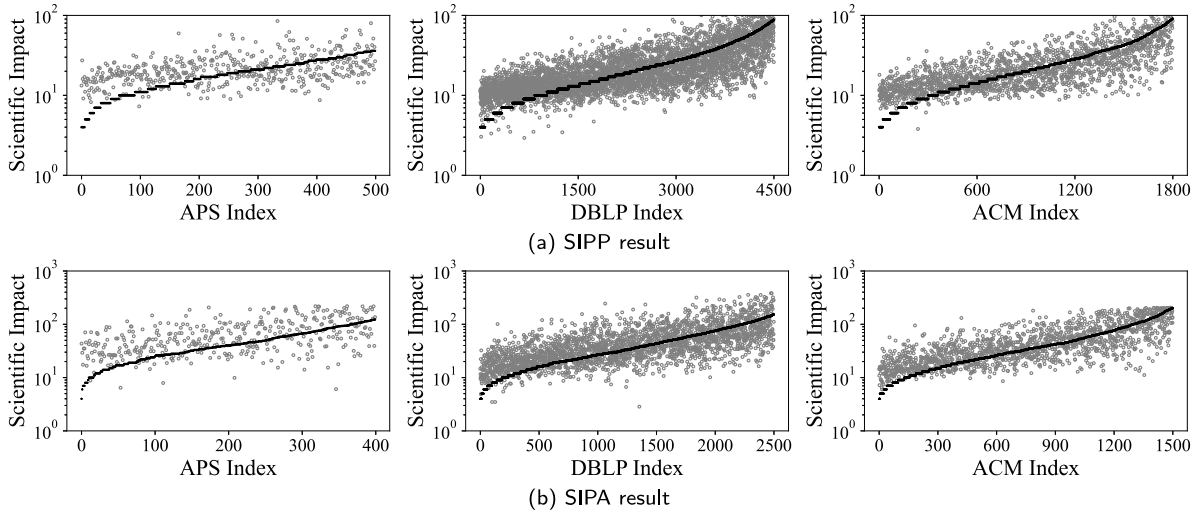

Fig. 7. The comparison of predicted value and the ground truth for SIPP and SIPA. Black solid points in all figures are the true value, while empty circles in gray color come to denote the predicted value.

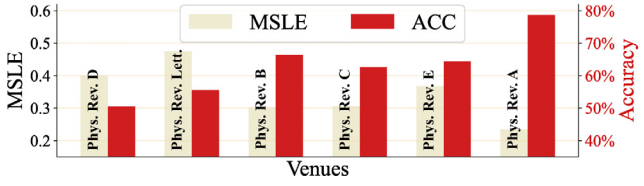

Fig. 8. Paper Prediction Performance on 6 representative APS venues.

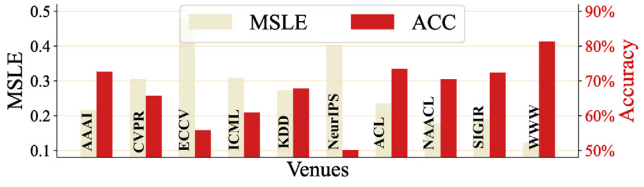

Fig. 9. Paper Prediction Performance on 10 representative AI conferences in DBLP dataset.

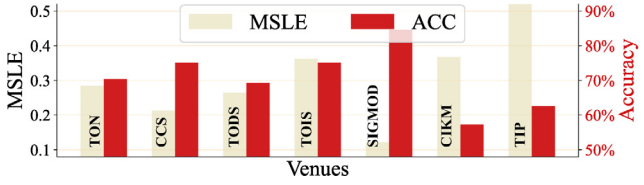

Fig. 10. Paper Prediction Performance on 7 representative ACM journals and conferences.

for scientific impact prediction. In addition, we also visualize a "crowd effect" of high impact papers/authors, as shown in ***(b)*** and ***(d)***. This also implies strong correlations among the latent representations of high impact scholars and papers. Another interesting result that can be visualized is the gradually decaying color of the paper/author citations, implying that a heavy-tailed distribution of scientific impact is successfully (to some extent at least) encoded in our model.

For DBLP, we choose 13 representative conferences in AI areas and classify them into 5 domains according to CSRanking (http://csrankings.org/). The conferences selected from the Artificial intelligence field are AAAI and IJCAI, while the conferences selected from the Computer vision field include CVPR, ECCV and ICCV. Conferences ICML, KDD and NeurIPS represent the Machine learning & data mining field. Another three conferences ACL, EMNLP and NAACL point to the natural language processing field. Two conferences (SIGIR and WWW) are selected for the Web & information retrieval field. Different with the APS author t-SNE (cf. Fig. 11(b)) with clear clustering, the DBLP authors of four domains, i.e., AI, ML&DM, NLP and W&I, together (cf. Fig. 12(b)), except with the authors from the CV field. We speculate that this is the cause of the difference between disciplines. The themes of venues in APS differ significantly from each other. For example, the Phys. Rev. B (Physical Review B) focuses on condensed matter physics while the journal Phys. Rev. C (Physical Review C) concentrates on the Nuclear physics. However, crossover studies can be widely found in the AI domain. Some conferences, such as IJCAI, AAAI, NeurIPS and ICML, receive papers from the CV, NLP or DM fields. Moreover, we can see that although some authors (green colored points) who published more papers at the comprehensive fields of conference, e.g., the ML&DM with ICML, KDD and NeurIPS, can also be discriminated that they are closer to the CV or NLP field. That is our heterogeneous academic network, which extracts complex cooperation relations and citing-cited behavior, contributes to this meaningful node representation. Similar results can also been found in Fig. 13 for ACM dataset.

5.9. Case study

In this section, we present a series of analyses to help us better understand the performance of SI-HDGNN author prediction. We first explore when our model works or fails from a statistical perspective. Next, we study the publication and citation trends of the individual scholars.

5.9.1. Interpreting the performance differences

To gain deeper insight into SI-HDGNN author prediction from a statistical perspective, we select the 500 best and 500 worst author predictions produced by SI-HDGNN in the DBLP dataset. Then, we count and compute the average quantity for the publication and citation of new authors in 2006, i.e., their first publications were received in 2006, with each passing year. The yearly trend for the worst 500, best 500, and all authors are shown in Fig. 15.

Moreover, Figs. 14 and 16 plot the trends for scholars in the APS and ACM datasets, respectively. Figs. 14(a) and 15 (a) delineate that the number of publications for both CS and Physics

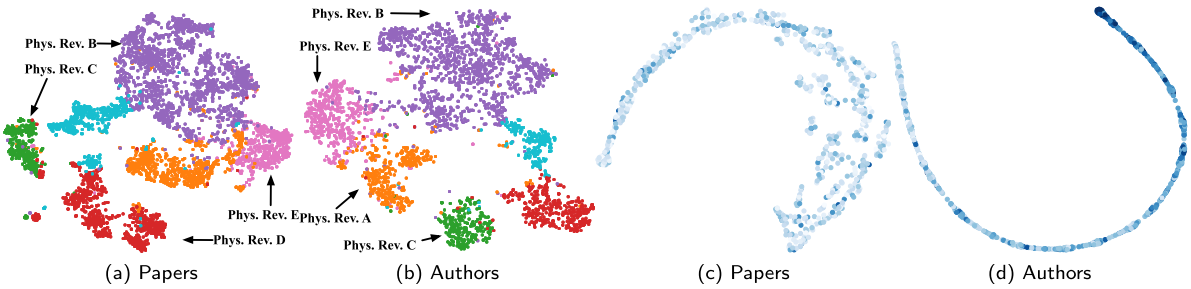

Fig. 11. Plot APS dataset latent space of 6 venues after mapping to 2D with t-SNE (best viewed in color). (***a***) and (***c***) show paper/author embeddings retrieved from the heterogeneous graph representation (i.e., $E(p)$ or $E(a)$, cf. Section 4.1) – colors are specified by venues; (***b***) and (***d***) plot paper/author citation embeddings from the prediction layer (i.e., $\mathbf{h}^2$, cf. Section 4.2) – colors are specified by magnitude of citations.

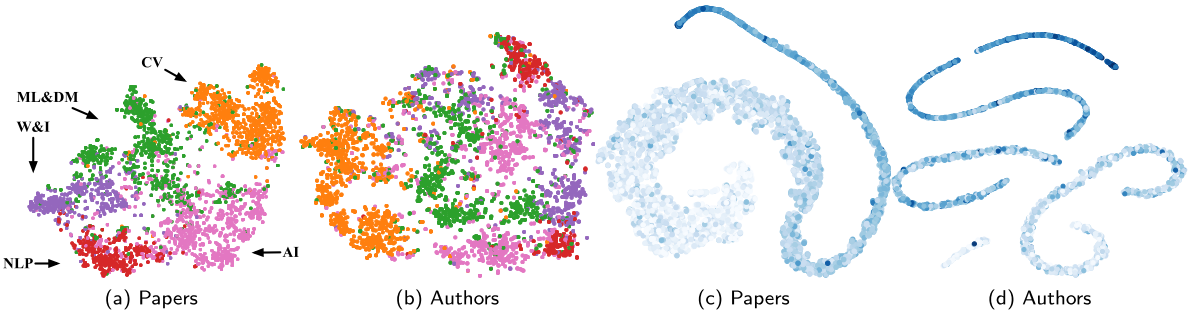

Fig. 12. Plot DBLP dataset latent space of 5 domains of AI paper after mapping to 2D with t-SNE. The abbreviations AI, CV, ML&DM, NLP and W&I represent the Artificial intelligence, Computer vision, Machine learning & data mining, Natural language processing and Web & information retrieval, respectively. (**a**) and (**c**) show paper/author embeddings retrieved from the heterogeneous graph representation — colors are specified by venues; (**b**) and (**c**) plot paper/author citation embeddings from the prediction layer — colors are specified by magnitude of citations.

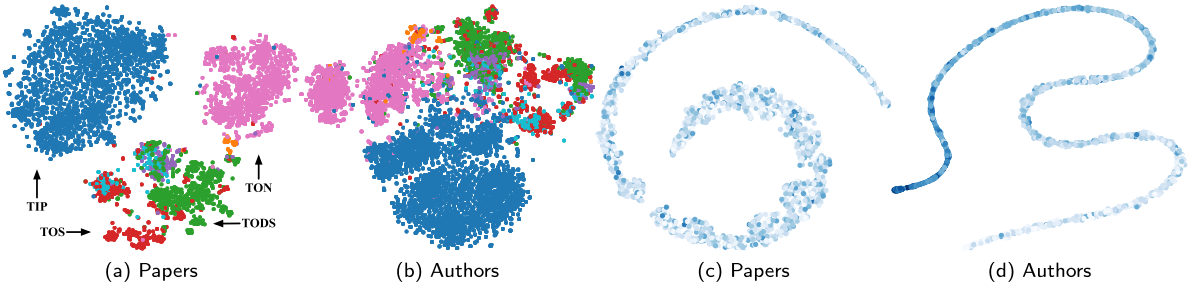

Fig. 13. Plot ACM dataset latent space of 7 venues of Computer Science paper after mapping to 2D with t-SNE. (**a**) and (**c**) show paper/author embeddings retrieved from the heterogeneous graph representation — colors are specified by venues; (**b**) and (**c**) plot paper/author citation embeddings from the prediction layer — colors are specified by magnitude of citations.

authors may drop in the early stage of his or her academical career. We can see that the citation quantity per year for these authors increases linearly in the first 10 years in Figs. 14(b) and 15(b).

We speculate that authors with steady publications and increasing citations are harder to model by SI-HDGNN. One reason is that the useful information utilized for impact prediction is constrained by the length of the observation window. We can see that the citations are the same for the worst 500 and best 500 in the first two years (cf. Fig. 15(b)). Therefore, the predictor is trained with indifferent knowledge as two years of observation is not enough for authors to receive diverse citations. Another reason is the unbalanced data distribution. The author impact actually follows a heavy-tailed distribution. Most researchers are not so popular, while only a few scholars have a high reputation. This phenomenon contributes to the unbalanced trained data distribution. The classifier receives many supervision signals from general scholars so that the impact predictions of popular authors are hard to derive in such situations. One possible solution to alleviate this inherent unbalanced data distribution problem is *decoupling representation* [81], which can be seen as future work.

5.9.2. Performance of individual scholars

We collect six anonymous authors with good and bad performances in three datasets, i.e., each dataset has one good author prediction and one bad. We plot the yearly publication and citation for them in Figs. 17, 18 and 19, respectively. For APS, we choose two new authors in 1997, i.e., their first publication years are 1997, denoted as APS_{good} and APS_{bad}. After 20 years in 2017, APS_{good} received 264 citations while APS_{bad} received 280 citations. The predicted values are 263 and 117, respectively. During the observation time 1997–1999, APS_{good} received approximately 30 citations while the number for APS_{bad} was only 5. From Fig. 17, we can see that authors with fewer publications and citations in the observations are more likely to be predicted with a fewer citations.

For DBLP, we choose one good author prediction with 361 for the prediction number and 397 for ground-truth, and one bad

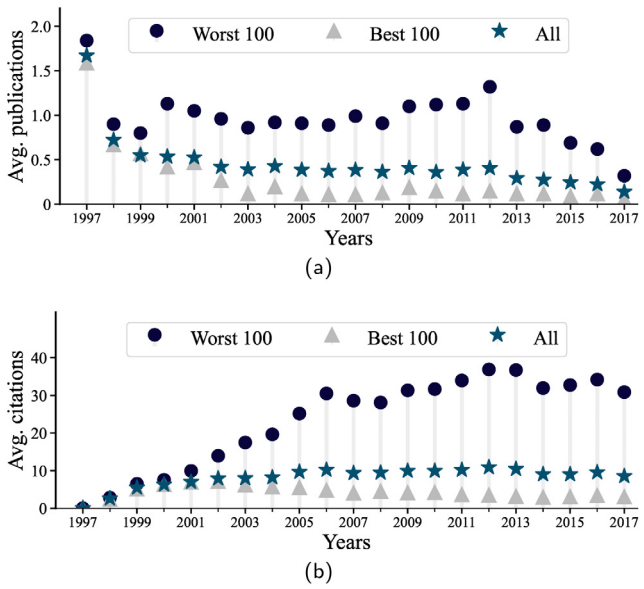

Fig. 14. Average publication (a) and citation (b) per year for the worst 100, best 100 and all authors of the segmentation test from 1997 to 2017 in APS dataset.

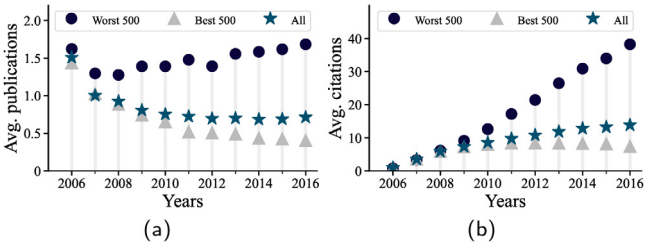

Fig. 15. Average publication (a) and citation (b) per year for the worst 500, best 500 and all authors of the segmentation test from 2006 to 2016 in DBLP dataset.

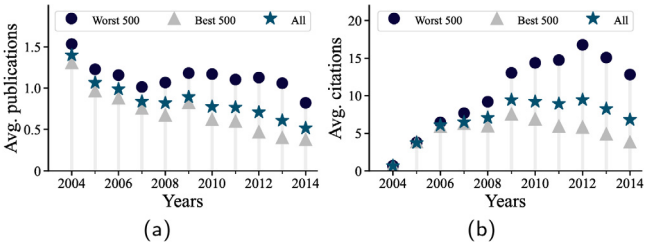

Fig. 16. Average publication (a) and citation (b) per year for the worst 500, best 500 and all authors of the segmentation test from 2004 to 2014 in ACM dataset.

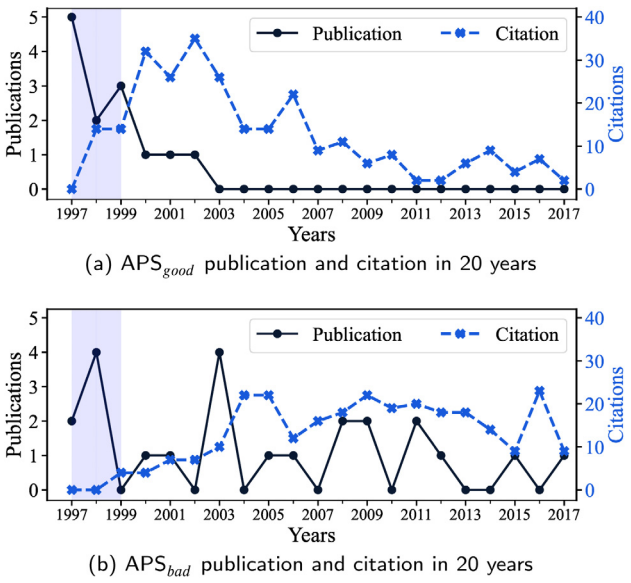

Fig. 17. Yearly publication and citation for two authors from 1997 to 2017 in APS dataset. Shaded area highlights the author's publication and citation during the observation window $[t_0, t_{ob}] = 1997 - 1999$.

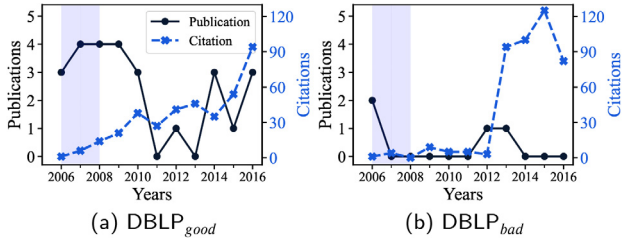

Fig. 18. Yearly publication and citation for two authors from 2006 to 2016 in DBLP dataset.

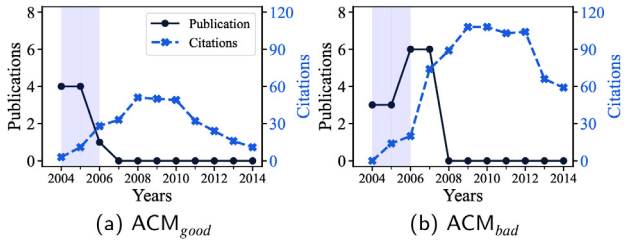

Fig. 19. Yearly publication and citation for two authors from 2004 to 2014 in ACM dataset.

author prediction with 40 for the prediction number and 451 for the ground-truth from prediction results.

Our model failed when one's flashpoint came up in her later academic career (cf. Fig. 18(b)). Another observation is that some researchers may be active for the first five years and will not publish paper or cooperate with others after that (cf. Figs. 17(a), 18(b), 19(a) and (b)). This is normal in real life when one starts her research as she is a PhD candidate, and drops her academic career after graduation. However, this actually contributes to the difficulty of long-term author prediction, as whether she will continue her research career or not can hardly be inferred with limited observations.

6. Conclusion

We presented SI-HDGNN, a novel methodology for effectively quantifying and predicting the scientific impact of scholars and research publications by bridging the dynamic processes of impact evolution and complex heterogeneous node interactions. We devised an efficient network sampling method considering rich node relations, along with a temporally attentive neighbor aggregation network to model the complex and accumulating dynamic processes of scientific impact. We further developed a temporal aggregation module specifically for author scientific impact prediction for performance improvement. Evaluations on three large-scale real-world academic datasets demonstrated the superior performance of SI-HDGNN in comparison to several state-of-the-art baselines. An immediate extension of our work is to

incorporate other kinds of "social contexts" – e.g., similarity – in the node sampling strategy. Our future work will focus on two broad categories of problems: (1) we will investigate the impact of cross-institutional collaboration on citations, and (2) we will leverage SI-HDGNN to develop effective and efficient methodologies for other tasks related to scientific publications, such as co-author and paper recommendations [82,83].

CRediT authorship contribution statement

Xovee Xu: Writing – original draft, Visualization, Conceptualization, Methodology, Formal analysis, Revision. **Ting Zhong:** Supervision, Funding acquisition, Investigation, Resources, Review & editing, Project administration. **Ce Li:** Conceptualization, Writing – review & editing, Methodology, Software, Investigation, Visualization, Revision. **Goce Trajcevski:** Writing – review & editing, Investigation, Funding acquisition, Revision. **Fan Zhou:** Supervision, Funding acquisition, Investigation, Resources, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No. 62176043 and No. 62072077) and the NSF SWIFT grant 2030249.

References

[1] D. Zhang, S. Mishra, E. Brynjolfsson, J. Etchemendy, D. Ganguli, B. Grosz, T. Lyons, J. Manyika, J.C. Niebles, M. Sellitto, et al., The ai index 2021 annual report, 2021, arXiv:2103.06312.
[2] D. Wang, C. Song, A.L. Barabási, Quantifying long-term scientific impact, Science 342 (2013) 127–132.
[3] S. Fortunato, C.T. Bergstrom, K. Börner, J.A. Evans, D. Helbing, S. Milojević, A.M. Petersen, F. Radicchi, R. Sinatra, B. Uzzi, et al., Science of science, Science 359 (2018).
[4] S.U. Hassan, N.R. Aljohani, N. Idrees, R. Sarwar, R. Nawaz, E. Martínez-Cámara, S. Ventura, F. Herrera, Predicting literature's early impact with sentiment analysis in Twitter, Knowl.-Based Syst. 192 (2020) 105383.
[5] T. Pradhan, S. Pal, A multi-level fusion based decision support system for academic collaborator recommendation, Knowl.-Based Syst. 197 (2020) 105784.
[6] S. Molaei, H. Zare, H. Veisi, Deep learning approach on information diffusion in heterogeneous networks, Knowl.-Based Syst. 189 (2020) 105153.
[7] R. Sinatra, D. Wang, P. Deville, C. Song, A.L. Barabási, Quantifying the evolution of individual scientific impact, Science 354 (2016) aaf5239.
[8] Q. Ke, E. Ferrara, F. Radicchi, A. Flammini, Defining and identifying sleeping beauties in science, Proc. Nat. Acad. Sci. (PNAS) 112 (2015) 7426–7431.
[9] D.J.D.S. Price, Networks of scientific papers, Science 149 (1965) 510–515.
[10] R. Yan, J. Tang, X. Liu, D. Shan, X. Li, Citation count prediction: learning to estimate future citations for literature, in: CIKM, 2011.
[11] D.E. Acuna, S. Allesina, K.P. Kording, Predicting scientific success, Nature 489 (2012) 201–202.
[12] B. Uzzi, S. Mukherjee, M. Stringer, B. Jones, Atypical combinations and scientific impact, Science 342 (2013) 468–472.
[13] Y. Dong, R.A. Johnson, N.V. Chawla, Can scientific impact be predicted? IEEE Trans. Big Data 2 (2016) 18–30.
[14] X.P. Zhu, Z. Ban, Citation count prediction based on academic network features, in: International Conference on Advanced Information Networking and Applications, AINA, 2018, pp. 534–541.
[15] A. Clauset, D.B. Larremore, R. Sinatra, Data-driven predictions in the science of science, Science 355 (2017).
[16] H. Shen, D. Wang, C. Song, A.L. Barabási, Modeling and predicting popularity dynamics via reinforced Poisson processes, in: AAAI, 2014, pp. 291–297.
[17] S. Xiao, J. Yan, C. Li, B. Jin, X. Wang, X. Yang, S.M. Chu, H. Zha, On modeling and predicting individual paper citation count over time, in: IJCAI, 2016, pp. 2676–2682.
[18] Q. Cao, H. Shen, K. Cen, W. Ouyang, X. Cheng, Deephawkes: Bridging the gap between prediction and understanding of information cascades, in: CIKM, 2017, pp. 1149–1158.
[19] F. Zhou, X. Xu, G. Trajcevski, K. Zhang, A survey of information cascade analysis: Models, predictions, and recent advances, ACM Comput. Surv. 54 (2021) 27:1–27:36, http://dx.doi.org/10.1145/3433000.
[20] C. Li, J. Ma, X. Guo, Q. Mei, Deepcas: An end-to-end predictor of information cascades, in: WWW, 2017, pp. 577–586.
[21] J. Chung, C. Gulcehre, K. Cho, Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling, 2014, arXiv:1412.3555.
[22] X. Chen, F. Zhou, K. Zhang, G. Trajcevski, T. Zhong, F. Zhang, Information diffusion prediction via recurrent cascades convolution, in: ICDE, 2019, pp. 770–781.
[23] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, in: ICLR, 2017.
[24] X. Xu, F. Zhou, K. Zhang, S. Liu, G. Trajcevski, CasFlow: EXploring hierarchical structures and propagation uncertainty for cascade prediction, IEEE Trans. Knowl. Data Eng. (TKDE) (2021) http://dx.doi.org/10.1109/TKDE.2021.3126475.
[25] F. Zhou, X. Xu, K. Zhang, G. Trajcevski, T. Zhong, Variational information diffusion for probabilistic cascades prediction, in: INFOCOM, 2020, pp. 1618–1627.
[26] H. Huang, R. Shi, W. Zhou, X. Wang, H. Jin, X. Fu, Temporal heterogeneous information network embedding, in: IJCAI, 2021, pp. 1470–1476, http://dx.doi.org/10.24963/ijcai.2021/203.
[27] R. Trivedi, M. Farajtabar, P. Biswal, H. Zha, Dyrep: Learning representations over dynamic graphs, in: ICLR, 2019.
[28] F. Manessi, A. Rozza, M. Manzo, Dynamic graph convolutional networks, Pattern Recognit. 97 (2020) 1–18.
[29] C. Zhang, D. Song, C. Huang, A. Swami, N.V. Chawla, Heterogeneous graph neural network, in: KDD, 2019, pp. 793–803.
[30] Y. Lu, C. Shi, L. Hu, Z. Liu, Relation structure-aware heterogeneous information network embedding, in: AAAI, 2019, pp. 4456–4463.
[31] Y. Shi, Q. Zhu, F. Guo, C. Zhang, J. Han, Easing embedding learning by comprehensive transcription of heterogeneous information networks, in: KDD, 2018, pp. 2190–2199.
[32] Y. Cen, X. Zou, J. Zhang, H. Yang, J. Zhou, J. Tang, Representation learning for attributed multiplex heterogeneous network, in: KDD, 2019, pp. 1358–1368.
[33] J.E. Hirsch, An index to quantify an individual's scientific research output, Proc. Nat. Acad. Sci. (PNAS) 102 (2005) 16569–16572.
[34] E. Garfield, Citation indexes for science, Science 122 (1955) 108–111.
[35] L. Page, S. Brin, R. Motwani, T. Winograd, The PageRAnk Citation Ranking: Bringing Order To the Web, Technical Report, Stanford InfoLab, 1999.
[36] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent Dirichlet allocation, J. Mach. Learn. 3 (2003) 993–1022.
[37] E. Bütün, M. Kaya, R. Alhajj, A supervised learning method for prediction citation count of scientists in citation networks, in: ASONAM, 2017, pp. 952–958.
[38] E. Bütün, M. Kaya, Predicting citation count of scientists as a link prediction problem, IEEE Trans. Cybern. 50 (2019) 4518–4529.
[39] Y. Dong, R.A. Johnson, N.V. Chawla, Will this paper increase your h-index?: Scientific impact prediction, in: WSDM, 2015, pp. 149–158.
[40] T. Chen, C. Guestrin, XGBoost: A scalable tree boosting system, in: KDD, 2016, pp. 785–794.
[41] R. Crane, D. Sornette, Robust dynamic classes revealed by measuring the response function of a social system, Proc. Nat. Acad. Sci. (PNAS) 105 (2008) 15649–15653.
[42] Q. Zhao, M.A. Erdogdu, H.Y. He, A. Rajaraman, J. Leskovec, Seismic: A self-exciting point process model for predicting tweet popularity, in: KDD, 2015, pp. 1513–1522.
[43] Q. Kong, M.A. Rizoiu, L. Xie, Describing and predicting online items with reshare cascades via dual mixture self-exciting processes, in: CIKM, 2020, pp. 645–654.
[44] S. Mishra, M.A. Rizoiu, L. Xie, Feature driven and point process approaches for popularity prediction, in: CIKM, 2016, pp. 1069–1078.
[45] Q. Kong, M.A. Rizoiu, L. Xie, Modeling information cascades with self-exciting processes via generalized epidemic models, in: WSDM, 2020, pp. 286–294.
[46] X. Feng, Q. Zhao, J. Ma, G. Jiang, On modeling and predicting popularity dynamics via integrating generative model and rich features, Knowl.-Based Syst. 196 (2020) 105786.
[47] H. Wang, C. Yang, Information diffusion prediction with latent factor disentanglement, 2020, arXiv:2012.08828.
[48] S. Li, W.X. Zhao, E.J. Yin, J.R. Wen, A neural citation count prediction model based on peer review text, in: EMNLP-IJCNLP, 2019, pp. 4916–4926.

[49] T. van Dongen, G.M.d.B. Wenniger, L. Schomaker, Schubert: Scholarly document chunks with bert-encoding boost citation count prediction, 2020, arXiv:2012.11740.
[50] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: NIPS, 2013, pp. 3111–3119.
[51] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: NIPS, 2017, pp. 5998–6008.
[52] J. Wen, L. Wu, J. Chai, Paper citation count prediction based on recurrent neural network with gated recurrent unit, in: IEEE International Conference on Electronics Information and Emergency Communication, ICEIEC, 2020, pp. 303–306.
[53] S. Yuan, J. Tang, Y. Zhang, Y. Wang, T. Xiao, Modeling and predicting citation count via recurrent neural network with long short-term memory, 2018, arXiv:1811.02129.
[54] F. Zhou, X. Jing, X. Xu, T. Zhong, G. Trajcevski, J. Wu, Continual information cascade learning, in: IEEE Global Communications Conference, GLOBECOM, 2020.
[55] X. Tang, D. Liao, W. Huang, J. Xu, L. Zhu, M. Shen, Fully exploiting cascade graphs for real-time forwarding prediction, in: AAAI, 2021, pp. 582–590.
[56] Z. Huang, Z. Wang, R. Zhang, Y. Zhao, F. Zheng, Learning bi-directional social influence in information cascades using graph sequence attention networks, in: WWW Companion, 2020, pp. 19–21.
[57] A.N. Holm, B. Plank, D. Wright, I. Augenstein, Longitudinal citation prediction using temporal graph neural networks, 2020, arXiv:2012.05742.
[58] C. Shi, Y. Li, J. Zhang, Y. Sun, P.S. Yu, A survey of heterogeneous information network analysis, IEEE Trans. Knowl. Data Eng. (2017) 17–37.
[59] Y. Dong, Z. Hu, K. Wang, Y. Sun, J. Tang, Heterogeneous network representation learning, in: IJCAI, 2020, pp. 4861–4867.
[60] X. Wang, D. Bo, C. Shi, S. Fan, Y. Ye, P.S. Yu, A survey on heterogeneous graph embedding: Methods, techniques, applications and sources, 2020, arXiv:2011.14867.
[61] C. Yang, Y. Xiao, Y. Zhang, Y. Sun, J. Han, Heterogeneous network representation learning: A unified framework with survey and benchmark, IEEE Trans. Knowl. Data Eng. (2020).
[62] J. Skardinga, B. Gabrys, K. Musial, Foundations and modelling of dynamic networks using dynamic graph neural networks: A survey, IEEE Access (2021).
[63] B. Perozzi, R. Al-Rfou, S. Skiena, Deepwalk: Online learning of social representations, in: KDD, 2014, pp. 701–710.
[64] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, Q. Mei, Line: Large-scale information network embedding, in: WWW, 2015, pp. 1067–1077.
[65] Y. Dong, N.V. Chawla, A. Swami, metapath2vec: Scalable representation learning for heterogeneous networks, in: KDD, 2017, pp. 135–144.
[66] T.y. Fu, W.C. Lee, Z. Lei, Hin2vec: Explore meta-paths in heterogeneous information networks for representation learning, in: CIKM, 2017, pp. 1797–1806.
[67] Y. He, Y. Song, J. Li, C. Ji, J. Peng, H. Peng, Hetespaceywalk: A heterogeneous spacey random walk for heterogeneous information network embedding, in: CIKM, 2019, pp. 639–648.
[68] X. Wang, H. Ji, C. Shi, B. Wang, Y. Ye, P. Cui, P.S. Yu, Heterogeneous graph attention network, in: WWW, 2019, pp. 2022–2032.
[69] Z. Hu, Y. Dong, K. Wang, Y. Sun, Heterogeneous graph transformer, in: WWW, 2020, pp. 2704–2710.
[70] J. Zhao, X. Wang, C. Shi, B. Hu, G. Song, Y. Ye, Heterogeneous graph structure learning for graph neural networks in: AAAI, 2021.
[71] S. Jiang, B. Koch, Y. Sun, HINTS: citation time series prediction for new publications via dynamic heterogeneous information network embedding, in: WWW, 2021.
[72] J. Devlin, M.W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, in: NAACL-HLT, 2019, pp. 4171–4186.
[73] H. Xiao, Bert-as-service, 2018, https://github.com/hanxiao/bert-as-service.
[74] H. Tong, C. Faloutsos, J.Y. Pan, Fast random walk with restart and its applications, in: ICDM, 2006, pp. 613–622.
[75] W. Hamilton, Z. Ying, J. Leskovec, Inductive representation learning on large graphs, in: NIPS, 2017, pp. 1024–1034.
[76] G. Veličović P. Cucurull, A. Casanova, A. Romero, P. Liò, Y. Bengio, Graph attention networks, in: ICLR, 2018.
[77] S. Hochreiter, The vanishing gradient problem during learning recurrent neural nets and problem solutions, Int. J. Uncertain. Fuzziness Knowl.-Based Syst. 6 (1998) 107–116.
[78] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: NeurIPS, 2013, pp. 3111–3119.
[79] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, Z. Su, Arnetminer: extraction and mining of academic social networks, in: KDD, 2008, pp. 990–998.
[80] J. Zhang, Y. Dong, Y. Wang, J. Tang, M. Ding, Prone: fast and scalable network representation learning, in: IJCAI, 2016, pp. 4278–4284.
[81] B. Kang, S. Xie, M. Rohrbach, Z. Yan, A. Gordo, J. Feng, Y. Kalantidis, Decoupling representation and classifier for long-tailed recognition, in: ICLR, 2020.
[82] Y. Zhu, Q. Lin, H. Lu, K. Shi, P. Qiu, Z. Niu, Recommending scientific paper via heterogeneous knowledge embedding based attentive recurrent neural networks, Knowl.-Based Syst. 215 (2021) 106744.
[83] X. Bai, M. Wang, I. Lee, Z. Yang, X. Kong, F. Xia, Scientific paper recommendation: A survey, IEEE Access 7 (2019) 9324–9339.