Artificial Intelligence Review (2024) 57:244
https://doi.org/10.1007/s10462-024-10885-1

Link prediction for hypothesis generation: an active curriculum learning infused temporal graph-based approach

Uchenna Akujuobi[1] · Priyadarshini Kumari[2] · Jihun Choi[3] · Samy Badreddine[1] · Kana Maruyama[3] · Sucheendra K. Palaniappan[4] · Tarek R. Besold[1]

Accepted: 25 July 2024 / Published online: 12 August 2024

Abstract

Over the last few years Literature-based Discovery (LBD) has regained popularity as a means to enhance the scientific research process. The resurgent interest has spurred the development of supervised and semi-supervised machine learning models aimed at making previously implicit connections between scientific concepts/entities within often extensive bodies of literature explicit—i.e., suggesting novel scientific hypotheses. In doing so, understanding the temporally evolving interactions between these entities can provide valuable information for predicting the future development of entity relationships. However, existing methods often underutilize the latent information embedded in the temporal aspects of the interaction data. Motivated by applications in the food domain—where we aim to connect nutritional information with health-related benefits—we address the hypothesis-generation problem using a temporal graph-based approach. Given that hypothesis generation involves predicting future (i.e., still to be discovered) entity connections, in our view the ability to capture the dynamic evolution of connections over time is pivotal for a robust model. To address this, we introduce *THiGER*, a novel batch contrastive temporal node-pair embedding method. *THiGER* excels in providing a more expressive node-pair encoding by effectively harnessing node-pair relationships. Furthermore, we present *THiGER-A*, an incremental training approach that incorporates an active curriculum learning strategy to mitigate label bias arising from unobserved connections. By progressively training on increasingly challenging and high-utility samples, our approach significantly enhances the performance of the embedding model. Empirical validation of our proposed method demonstrates its effectiveness on established temporal-graph benchmark datasets, as well as on real-world datasets within the food domain.

Keywords Temporal graph neural network · Active learning · Hierarchical transformer · Curriculum learning · Literature based discovery · Edge prediction

Uchenna Akujuobi and Priyadarshini Kumari have contributed equally to this work.

Extended author information available on the last page of the article

1 Introduction

The saddest aspect of life right now is that science gathers knowledge faster than society gathers wisdom. — Isaac Asimov. Science is advancing at an increasingly quick pace, as evidenced, for instance, by the exponential growth in the number of published research articles per year (White 2021). Effectively navigating this ever-growing body of knowledge is tedious and time-consuming in the best of cases, and more often than not becomes infeasible for individual scientists (Brainard 2020). In order to augment the efforts of human scientists in the research process, computational approaches have been introduced to automatically extract hypotheses from the knowledge contained in published resources. Swanson (1986) systematically used a scientific literature database to find potential connections between previously disjoint bodies of research, as a result hypothesizing a (later confirmed) curative relationship between dietary *fish oils* and *Raynaud's syndrome*. Swanson and Smalheiser then automatized the search and linking process in the ARROWSMITH system (Swanson and Smalheiser 1997). Their work and other more recent examples (Fan and Lussier 2017; Trautman 2022) clearly demonstrate the usefulness of computational methods in extracting latent information from the vast body of scientific publications.

Over time, various methodologies have been proposed to address the Hypothesis Generation (HG) problem. Swanson and Smalheiser (Smalheiser and Swanson 1998; Swanson and Smalheiser 1997) pioneered the use of a basic ABC model grounded in a stringent interpretation of structural balance theory (Cartwright and Harary 1956). In essence, if entities A and B, as well as entities A and C, share connections, then entities B and C should be associated. Subsequent years have seen the exploration of more sophisticated machine learning-based approaches for improved inference. These encompass techniques such as text mining (Spangler et al. 2014; Spangler 2015), topic modeling (Sybrandt et al. 2017; Srihari et al. 2007; Baek et al. 2017), association rules (Hristovski et al. 2006; Gopalakrishnan et al. 2016; Weissenborn et al. 2015), and others (Jha et al. 2019; Xun et al. 2017; Shi et al. 2015; Sybrandt et al. 2020)

In the context of HG, where the goal is to predict novel relationships between entities extracted from scientific publications, comprehending prior relationships is of paramount importance. For instance, in the domain of social networks, the principles of social theory come into play when assessing the dynamics of connections between individuals. When there is a gradual reduction in the social distance between two distinct individuals, as evidenced by factors such as the establishment of new connections with shared acquaintances and increased geographic proximity, there emerges a heightened likelihood of a subsequent connection between these two individuals (Zhang and Pang 2015; Gitmez and Zárate 2022). This concept extends beyond social networks and finds relevance in predicting scientific relationships or events through the utilization of temporal information (Crichton et al. 2018; Krenn et al. 2023; Zhang et al. 2022). In both contexts, the principles of proximity and evolving relationships serve as valuable indicators, enabling a deeper understanding of the intricate dynamics governing these complex systems.

Modeling these relationships' temporal evolution assumes a critical role in constructing an effective and resilient hypothesis generation model. To harness the temporal dynamics, Akujuobi et al. (2020b, 2020a) and Zhou et al. (2022) conceptualize the HG task as a temporal graph problem. More precisely, given a sequence of graphs $G = \{G_0, G_2, \dots, G_T\}$, the objective is to deduce which previously unlinked nodes in G_T ought to be connected. In this framework, nodes denote biomedical entities, and the graphs G_τ represent temporal graphlets (see Fig. 1).

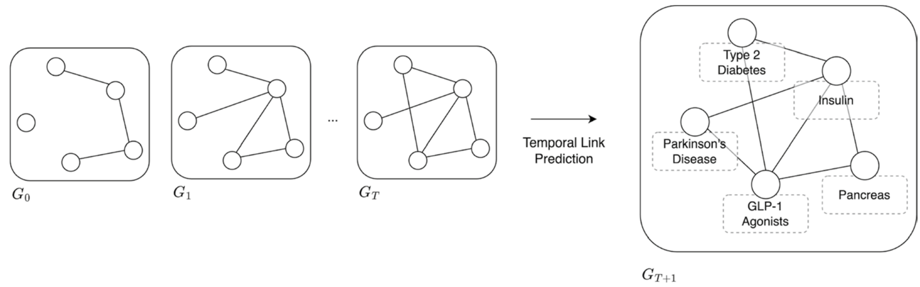


Fig. 1 Modeling hypothesis generation as a temporal link prediction problem

Definition 1 **Temporal graphlet**: A temporal graphlet $G_\tau = \{V^\tau, E^\tau\}$ is a temporal subgraph at time point τ, where $V^\tau \subset V$ and $E^\tau \subset E$ are the temporal set of nodes and edges of the subgraph.

Their approach tackles the HG problem by introducing a temporal perspective. Instead of relying solely on the final state E_T on a static graph, it considers how node pairs evolve over discrete time steps $E^\tau : \tau = 0 \ldots T$. To model this sequential evolution effectively, Akujuobi et al. and Zhou et al. leverage the power of recurrent neural networks (RNNs) (see Fig. 2a). However, it is essential to note that while RNNs have traditionally been the preferred choice for HG, their sequential nature may hinder capturing long-range dependencies, impacting performance for lengthy sequences.

To shed these limitations, we propose *THiGER* (**T**emporal **Hi**erarchical **G**raph-based **E**ncoder **R**epresentation), a robust transformer-based model designed to capture the evolving relationships between node pairs. THiGER overcomes the constraints of previous methods by representing temporal relationships hierarchically (see Fig. 2b). The proposed hierarchical layer-wise framework presents an incremental approach to comprehensively model the temporal dynamics among given concepts. It achieves this by progressively extracting the temporal interactions between consecutive time steps, thus enabling the model to prioritize attention to the informative regions of temporal evolution during the process. Our method effectively addresses issues arising from imbalanced temporal information (see Sect. 5.2). Moreover, it employs a contrastive learning strategy to improve the quality of task-specific node embeddings for node-pair representations and relationship inference tasks.

An equally significant challenge in HG is the lack of negative-class samples for training. Our dataset provides positive-class samples, which represent established connections between entities, but it lacks negative-class samples denoting non-existent connections (as opposed to undiscovered connections, which could potentially lead to scientific breakthroughs). This situation aligns with the positive-unlabeled (PU) learning problem. Prior approaches have typically either discarded unobserved connections as uninformative or wrongly treated them as negative-class samples. The former approach leads to the loss of valuable information, while the latter introduces label bias during training.

In response to these challenges, we furthermore introduce THiGER-A, an active curriculum learning strategy designed to train the model incrementally. THiGER-A utilizes progressively complex positive samples and highly informative, diverse unobserved connections as negative-class samples. Our experimental results demonstrate that by employing

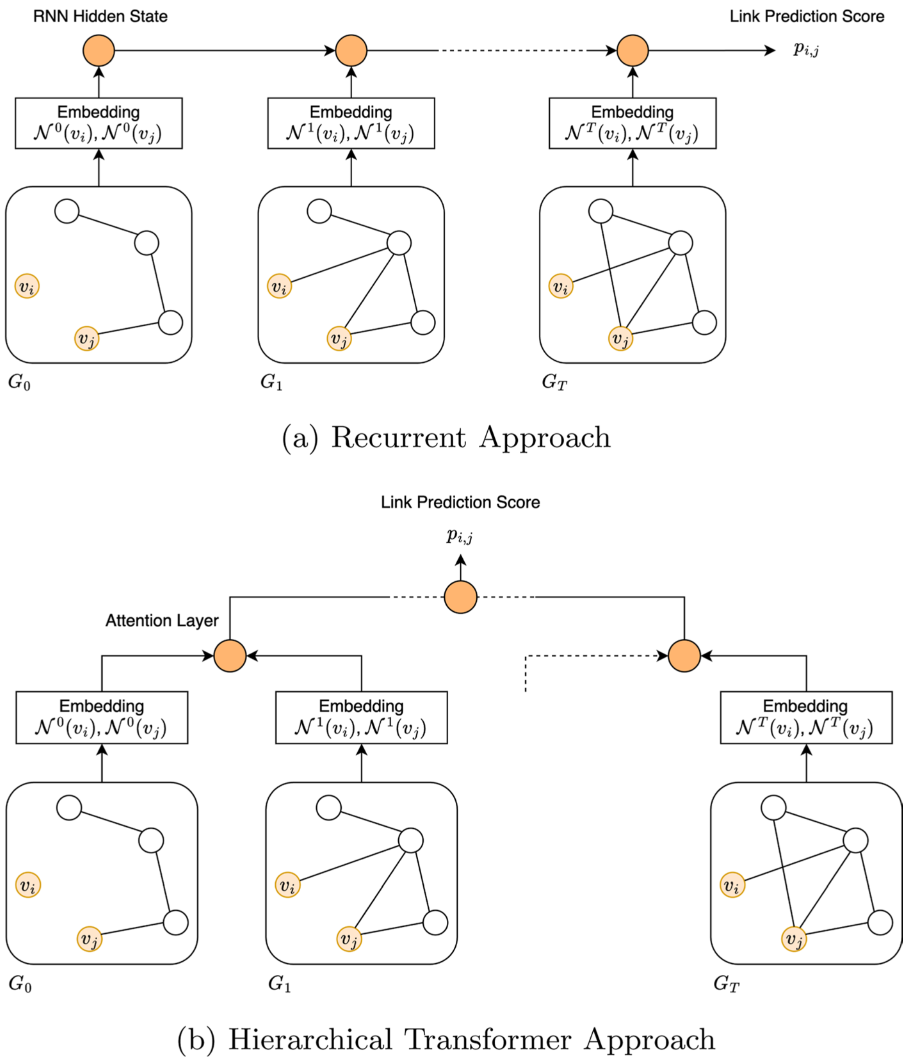


Fig. 2 Predicting the link probability $p_{i,j}$ for a node pair v_i and v_j using **a** a Recurrent Neural Network approach (Akujuobi et al. 2020b; Zhou et al. 2022), **b** THiGER, our approach. The recurrent approach aggregates the neighborhood information $\mathcal{N}^t(v_i)$ and $\mathcal{N}^t(v_j)$ sequentially while THiGER aggregates the neighborhood information hierarchically in parallel

incremental training with THiGER-A, we achieve enhanced convergence and performance for hypothesis-generation models compared to training on the entire dataset in one go. Remarkably, our approach demonstrates strong generalization capabilities, especially in challenging inductive test scenarios where the entities were not part of the seen training dataset.

Inspired by Swanson's pioneering work, we chose the food domain as a promising application area for THiGER. This choice is motivated by the increasing prevalence of diet-related health conditions, such as obesity and type-2 diabetes, alongside the growing recognition and utilization of the health benefits associated with specific food products in wellness and medical contexts.

In summary, our contributions are as follows:

Methodology: We propose a novel temporal hierarchical transformer-based architecture for node pair encoding. In utilizing the temporal batch-contrastive strategy, our architecture differs from existing approaches that learn in conventional static or temporal graphs. In addition, we present a novel incremental training strategy for temporal graph node pair embedding and future relation prediction. This strategy effectively mitigates negative-label bias through active learning and improves generalization by training the model progressively on increasingly complex positive samples using curriculum learning.

Evaluation: We test the model's efficacy on several real-world graphs of different sizes to give evidence for the model's strength for temporal graph problems and hypothesis generation. The model is trained end-to-end and shows superior performance on HG tasks.

Application: To the best of our knowledge, this is the first application of temporal hypothesis generation in the health-related food domain. Through case studies, we validate the practical relevance of our findings.

The remaining sections of this paper include a discussion of related work in Sect. 2, a detailed introduction of the proposed *THiGER* model and the *THiGER-A* active curriculum learning strategy in Sect. 3, an overview of the datasets, the model setup and parameter tuning, and our evaluation approach in Sect. 4, the results of our experimental evaluations in Sect. 5, and finally, our conclusions and a discussion of future work in Sect. 6.

2 Related works

2.1 Hypothesis generation

The development of effective methods for machine-assisted discovery is crucial in pushing scientific research into the next stage (Kitano 2021). In recent years, several approaches have been proposed in a bid to augment human abilities relevant to the scientific research process including tools for research design and analysis (Tabachnick and Fidell 2000), process modelling and simulation (Klein et al. 2002), or scientific hypothesis generation (King et al. 2004, 2009).

The early pioneers of the hypothesis generation domain proposed the so called ABC model for generating novel scientific hypothesis based on existing knowledge (Swanson 1986; Swanson and Smalheiser 1997). ABC-based models are simple and efficient, and have been implemented in classical hypothesis generation systems such as ARROWSMITH (Swanson and Smalheiser 1997). However, several drawbacks remain, including the need for similarity metrics defined on heuristically determined term lists and significant costs in terms of computational complexity with respect to the size of common entities.

More recent approaches, thus, have aimed to curtain the limitation of the ABC model. Spangler et al. (2014); Spangler (2015) proposed text mining techniques to identify entity relationships from unstructured medical texts. AGATHA (Sybrandt et al. 2020) used a transformer encoder architecture to learn the ranking criteria between regions of a given semantic graph and the plausibility of new research connections. Srihari et al. (2007); Baek et al. (2017) proposed several text mining approaches to detect how concepts are linked within and across multiple text documents. Sybrandt et al. (2017) proposed incorporating machine learning techniques such as clustering and topical phrase mining. Shi et al. (2015)

modeled the probability that concepts will be linked based on a given time window using random walks.

The previously mentioned methods do not consider temporal attributes of the data. More recent works (Jha et al. 2019; Akujuobi et al. 2020a; Zhou et al. 2022; Xun et al. 2017) argue that capturing the temporal information available in scholarly data can lead to better predictive performance. Jha et al. (2019) explored the co-evolution of concepts across knowledge bases using a temporal matrix factorization framework. Xun et al. (2017) modeled concepts' co-occurrence probability using their temporal embedding. Akujuobi et al. (2020a, 2020b) and Zhou et al. (2022) captured the temporal information in the scholarly data using RNN techniques.

Our approach captures the dynamic relationship information using a temporal hierarchical transformer encoder model. This strategy alleviates the limitations of the RNN-based models. Furthermore, with the incorporation of active curriculum learning strategies, our model can incrementally learn from the data.

2.2 Temporal graph learning

Learning on temporal graphs has received considerable attention from the research community in recent years. Some works (Hisano 2018; Ahmed et al. 2016; Milani Fard et al. 2019) apply static methods on aggregated graph snapshots. Others, including (Zhou et al. 2018; Singer et al. 2019), utilize time as a regularizer over consecutive snap-shots of the graph to impose a smoothness constraint on the node embeddings. A popular category of approaches for dynamic graphs is to introduce point processes that are continuous in time. DyRep (Trivedi et al. 2019) models the occurrence of an edge as a point process using graph attention on the destination node neighbors. Dynamic-Triad (Zhou et al. 2018) models the evolution patterns in a graph by imposing a triadic closure-where a triad with three nodes is developed from an open triad (i.e., with two nodes not connected).

Some recent works on temporal graphs apply several combinations of GNNs and recurrent architectures (e.g., GRU). EvolveGCN (Pareja et al. 2020) adapts the graph convolutional network (GCN) model along the temporal dimension by using an RNN to evolve the GCN parameters. T-PAIR (Akujuobi et al. 2020b, a) recurrently learns a node pair embedding by updating GraphSAGE parameters using gated neural networks (GRU). TGN (Rossi et al. 2020) introduces a memory module framework for learning on dynamic graphs. TDE (Zhou et al. 2022) captures the local and global changes in the graph structure using hierarchical RNN structures. TNodeEmbed (Singer et al. 2019) proposes the use of orthogonal procrustes on consecutive time-step node embeddings along the time dimension.

However, the limitation of RNN remains due to their sequential nature and robustness especially when working on a long timeline. Since the introduction of transformers, there has been interest in their application on temporal graph data. More related to this work, Zhong and Huang (2023) and Wang et al. (2022) both propose the use of a transformer architecture to aggregate the node neighborhood information while updating the memory of the nodes using GRU. TLC (Wang et al. 2021a) design a two-stream encoder that independently processes temporal neighborhoods associated with the two target interaction nodes using a graph-topology-aware Transformer and then integrates them at a semantic level through a co-attentional Transformer.

Our approach utilizes a single hierarchical encoder model to better capture the temporal information in the network while simultaneously updating the node embedding on the task. The model training and node embedding learning is performed end-to-end.

2.3 Active curriculum learning

Active learning (AL) has been well-explored for vision and learning tasks (Settles 2012). However, most of the classical techniques rely on single-instance-oracle strategies, wherein, during each training round, a single instance with the highest utility is selected using measures such as uncertainty sampling (Kumari et al. 2020), expected gradient length (Ash et al. 2020), or query by committee (Gilad-Bachrach et al. 2006). The single-instance-oracle approach becomes computationally infeasible with large training datasets such as ours. To address these challenges, several batch-mode active learning methods have been proposed (Priyadarshini et al. 2021; Kirsch et al. 2019; Pinsler et al. 2019). Priyadarshini et al. (2021) propose a method for batch active metric learning, which enables sampling of informative and diverse triplet data for relative similarity ordering tasks. In order to prevent the selection of correlated samples in a batch, Kirsch et al. (2019); Pinsler et al. (2019) develop distinct methods that integrate mutual information into the utility function. All three approaches demonstrate effectiveness in sampling diverse batches of informative samples for metric learning and classification tasks. However, none of these approaches can be readily extended to our specific task of hypothesis prediction on an entity-relationship graph.

Inspired by human learning, Bengio et al. (2009) introduced the concept of progressive training, wherein the model is trained on increasingly difficult training samples. Various prior works have proposed different measures to quantify the difficulty of training examples. Hacohen and Weinshall (2019) introduced curriculum learning by transfer, where they developed a score function based on the prediction confidence of a pre-trained model. Wang et al. (2021b) proposed a curriculum learning approach specifically for graph classification tasks. Another interesting work is relational curriculum learning (RCL) (Zhang et al. 2023) suggests training the model progressively on complex samples. Unlike most prior work, which typically consider data to be independent, RCL quantifies the difficulty level of an edge by aggregating the embeddings of the neighboring nodes. While their approach utilizes similar relational data to ours, their method does not specifically tackle the challenges inherent to the PU learning setting, which involves sampling both edges and unobserved relationships from the training data. In contrast, our proposed method introduces an incremental training strategy that progressively trains the model by focusing on positive edges of increasing difficulty, as well as incorporating highly informative and diverse negative edges.

3 Methodology

3.1 Notation

- $G = \{G_0, \dots, G_T\}$ is a temporal graph such that $G_\tau = \{V^\tau, E^\tau\}$ evolves over time $\tau = 0 \dots T$,
- $e(v_i, v_j)$ or e_{ij} is used to denote the edge between nodes v_i and v_j, and (v_i, v_j) is used to denote the node pair corresponding to the edge,
- $y_{i,j}$ is the label associated with the edge $e(v_i, v_j)$,
- $\mathcal{N}^\tau(v)$ gives the neighborhood of a node v in V^τ,
- x_v is the embedding of a node v and is static across time steps,

- $z_{i,j}^{\tau}$ is the embedding of a node pair $\langle v_i, v_j \rangle$. It depends on the neighborhood of the nodes at a time step τ,
- $h_{i,j}^{[\tau_0,\tau_f]}$ is the embedding of a node pair over a time step window $\tau_0, \dots, \tau_f$ where $0 \le \tau_0 \le \tau_f \le T$,
- $f(.;\theta)$ is a neural network depending on a set of parameters θ. For brevity, θ can be omitted if it is clear from the context.
- E^+ and E^- are the subsets of positive and negative edges, denoting observed and non-observed connections between biomedical concepts, respectively.
- L is the number of encoder layers in the proposed model.

Algorithm 1 Hierarchical Node-Pair Embedding $h_{i,j}^{[\tau_0,\tau_f]}$

Require: $\{x_v : v \in V\}, f_A, f_E$
procedure EMBED$(v_i, v_j, \tau_0, \tau_f)$
  $\mathbf{T} \leftarrow (\tau_0, \tau_0 + 1, \dots, \tau_f)$ ▷ Sequence of leaf time steps
  $\mathbf{H} \leftarrow (z_{i,j}^{\tau_0}, z_{i,j}^{\tau_0+1}, \dots, z_{i,j}^{\tau_f})$ ▷ Sequence of leaf embeddings
    where $z_{i,j}^{\tau} = f_A(x_{v_i}, x_{v_j}, \mathbf{x}_{\mathcal{N}^{\tau}(v_i)}, \mathbf{x}_{\mathcal{N}^{\tau}(v_j)}; \theta_A)$
  $L \leftarrow \lceil \log_2 |\mathbf{T}| \rceil$
  for $l \leftarrow 1 \dots L$ **do**
    if $|\mathbf{T}| \bmod 2 \neq 0$ **then** ▷ Add padding for parity
      $\mathbf{H} \leftarrow (\mathbf{H}_0, \mathbf{H}_1, \dots, \mathbf{H}_s, H_{\text{padding}})$
    end if
    $\mathbf{T} \leftarrow (\mathbf{T}_0, \mathbf{T}_2, \dots, \mathbf{T}_{|\mathbf{T}|-1})$ ▷ Merge leaves in pairs
    $\mathbf{H} \leftarrow (f_E^l(\mathbf{H}_0, \mathbf{H}_1), f_E^l(\mathbf{H}_2, \mathbf{H}_3), \dots, f_E^l(\mathbf{H}_{|\mathbf{H}|-1}, \mathbf{H}_{|\mathbf{H}|}))$
  end for
  $h_{i,j}^{[\tau_0,\tau_f]} \leftarrow \mathbf{H}_0$
  return $h_{i,j}^{[\tau_0,\tau_f]}$
end procedure

Algorithm 2 Link Prediction

Require: f_C, EMBED
procedure PREDICT(v_i, v_j)
  $h_{i,j}^{[0,T]} \leftarrow$ EMBED$(v_i, v_j, 0, T)$ ▷ See Algorithm 1
  $p_{i,j} \leftarrow f_C(h_{i,j}^{[0,T]})$
  return $p_{i,j}$
end procedure

3.2 Model overview

The whole THiGER(-A) model is shown in Fig. 3b. Let $v_i, v_j \in V_T$ be nodes denoting two concepts. The pair is assigned a positive label $y_{i,j} = 1$ if a corresponding edge (i.e., a link) is observed in G_T. That is, $y_{i,j} = 1$ iff $e(v_i, v_j) \in E^T$, otherwise 0. The model predicts a score $p_{i,j}$ that reflects $y_{i,j}$. The prediction procedure is presented in Algorithm 2.

The link prediction score is given by a neural classifier $p_{i,j} = f_C(h_{i,j}^{[0,T]};\theta_C)$, where $h_{i,j}^{[0,T]}$ is an embedding vector for the node pair. This embedding is calculated in Algorithm 1 using a hierarchical transformer encoder and illustrated in Fig. 3a.

The input to the hierarchical encoder layer is the independent local node pair embedding aggregation at each time step shown in line 3 of algorithm 1 as

$$z_{i,j}^{\tau} = f_A(x_{v_i}, x_{v_j}, \mathbf{x}_{\mathcal{N}^{\tau}(v_i)}, \mathbf{x}_{\mathcal{N}^{\tau}(v_j)};\theta_A), \tag{1}$$

where $\mathbf{x}_{\mathcal{N}^{\tau}(v_i)} = \{x_{v'} : v' \in \mathcal{N}^{\tau}(v_i)\}$ and $\mathbf{x}_{\mathcal{N}^{\tau}(v_j)} = \{x_{v'} : v' \in \mathcal{N}^{\tau}(v_j)\}$ are the embeddings of the neighbors of x_{v_i} and x_{v_j} at the given time step.

Subsequently, the local node pair embeddings aggregation is processed by the aggregation layer illustrated in Fig. 3a and shown in line 10 of Algorithm 1. At each hierarchical layer, temporal node pair embeddings are calculated for a sub-window using

$$h_{i,j}^{[\tau-n,\tau]} = f_E^l(h_{i,j}^{[\tau-n,\tau-\frac{n}{2}]}, h_{i,j}^{[\tau-\frac{n}{2},\tau]};\theta_E^l), \tag{2}$$

where n represents the sub-window size. When necessary, we ensure an even number of leaves to aggregate by adding zero padding values $H_{\text{padding}} = \mathbf{0}_d$, where d is the dimension of the leaf embeddings. The entire encoder architecture is denoted as $f_E = f_E^l : l = 1 \dots L$.

In this work, the classifier $f_C(.;\theta_C)$ is modeled using a multilayer perceptron network (MLP), $f_A(.;\theta_A)$ is elaborated in Sect. 3.3, and $f_E(.;\theta_E)$ is modeled by a multilayer transformer encoder network, which is detailed in Sect. 3.4.

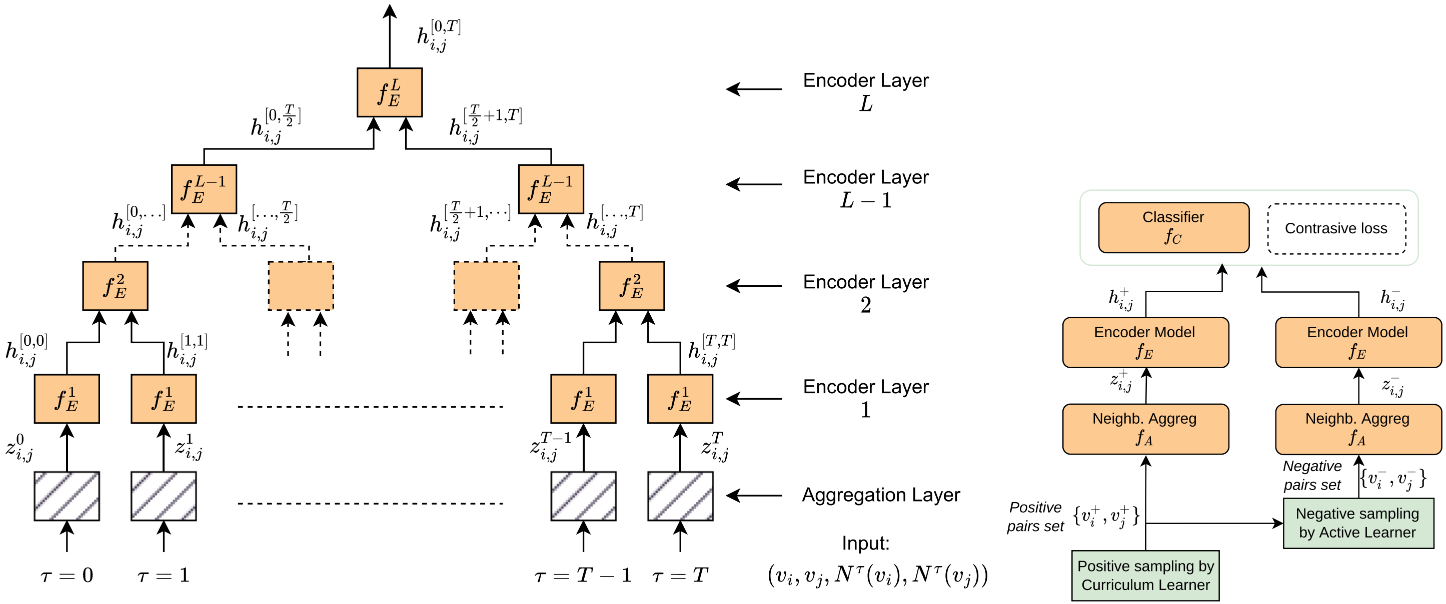
3.3 Neighborhood aggregation

The neighborhood aggregation is modeled using GraphSAGE (Hamilton et al. 2017). GraphSAGE uses K layers to iteratively aggregate a node embedding x_v and its neighbor embeddings $\mathbf{x}_{\mathcal{N}^{\tau}(v)} = \{x_{v'}, v' \in \mathcal{N}^{\tau}(v)\}$. f_A uses the GraphSAGE block to aggregate $(x_{v_i}, \mathbf{x}_{\mathcal{N}^{\tau}(v_i)})$ and $(x_{v_j}, \mathbf{x}_{\mathcal{N}^{\tau}(v_j)})$ in parallel, then merges the two aggregated representations using a MLP layer. In this paper, we explore three models based on the aggregation technique used at each iterative step of GraphSAGE.

Mean Aggregation: This straightforward technique amalgamates neighborhood representations by computing element-wise means of each node's neighbors and subsequently propagating this information iteratively. For all nodes within the specified set:

$$\beta_v^k \leftarrow \sigma\left(W^S \beta_v^{k-1} + \left|N^T(v)\right|^{-1} \sum_{v' \in N^T(v)} W^N \beta_v^{k-1}\right) \tag{3}$$

Here, β_v^k denotes the aggregated vector at iteration k, and β_v^{k-1} at iteration $k-1$. W^S and W^N represent trainable weights, and σ constitutes a sigmoid activation, collectively forming a conventional MLP layer.



(a) Temporal Hierarchical Multilayer Encoder

(b) Training Procedure

Fig. 3 Schematic representation of the proposed model for temporal node-pair link prediction. In **a**, the hierarchical graph transformer model takes as input the aggregated node pair embeddings obtained at each time step τ, these temporal node pair embeddings are further encoded and aggregated at each encoder layer. The final output is the generalized node pair embedding across all time steps. In **b**, a general overview of the model is given, highlighting the incorporation of the Active Curriculum Learning strategy

GIN (Graph Isomorphism Networks): Arguing that traditional graph aggregation methods, like mean aggregation, possess limited expressive power, GIN introduces the concept of aggregating neighborhood representations as follows:

$$\beta_v^k \leftarrow \sigma\left(W\left(\left(1+\epsilon^k\right)\cdot\beta_v^{k-1}+\sum_{v'\in\mathcal{N}^\tau(v)}\beta_{v'}^{k-1}\right)\right). \tag{4}$$

In this formulation, ϵ^k governs the relative importance of the node compared to its neighbors at layer k and can be a learnable parameter or a fixed scalar.

Multi-head Attention: We introduce a multi-head attention-based aggregation technique. This method aggregates neighborhood representations by applying multi-head attention to the node and its neighbors at each iteration:

$$\beta_v^k \leftarrow \sigma(W^S\beta_v^{k-1} + W^N\phi(\{\beta_v^{k-1}\}\cup\{\beta_{v'}^{k-1} : v'\in\mathcal{N}^\tau(v)\})). \tag{5}$$

Here, ϕ represents a multi-head attention function, as detailed in Vaswani et al. (2017).

3.3.1 Neighborhood definition

To balance performance and scalability considerations, we adopt the neighborhood sampling approach utilized in GraphSAGE to maintain a consistent computational footprint for each batch of neighbors. In this context, we employ a uniform sampling method to select a neighborhood node set of fixed size, denoted as $\mathcal{N}'(v)\subset\mathcal{N}^\tau(v)$, from the original neighbor set at each step. This sampling procedure is essential as, without it, the memory and runtime complexity of a single batch becomes unpredictable and, in the worst-case scenario, reaches a prohibitive $\mathcal{O}(|V|)$, making it impractical for handling large graphs.

3.4 Temporal hierarchical multilayer encoder layer

The temporal hierarchical multilayer encoder is the fundamental component of our proposed model, responsible for processing neighborhood representations collected over multiple time steps, specifically $(z_{i,j}^0, z_{i,j}^1, \dots, z_{i,j}^T)$. These neighborhood representations are utilized to construct a hierarchical tree.

At the initial hierarchical layer, we employ an encoder, denoted as f_E^1, to distill adjacent sequential local node-pair embeddings, represented as $(z_{i,j}^\tau, z_{i,j}^{\tau+1})$, combining them into a unified embedding, denoted as $h_{i,j}^{[\tau,\tau+1]}$. In cases where the number of time steps is not an even multiple of 2, a zero-vector dummy input is appended.

This process repeats at each hierarchical level l within the tree, with $h_{i,j}^{[\tau-n,\tau]} = f_E^l(h_{i,j}^{[\tau-n,\tau-\frac{n}{2}]}, h_{i,j}^{[(\tau-\frac{n}{2})+1,\tau]}; \theta_E^l)$. Each layer f_E^l consists of a transformer encoder block and may contain $N-1$ encoder sublayers, where $N\geq 1$. This mechanism can be viewed as an iterative knowledge aggregation process, wherein the model progressively summarizes the information from pairs of local node pair embeddings.

The output of each encoder layer, denoted as $h_{i,j}^{[\tau_0,\tau_f]}$, offers a comprehensive summary of temporal node pair information from time step τ_0 to τ_f. Finally, the output of the last layer, $h_{i,j}^{[0,T]}$, is utilized for inferring node pair relationships.

3.5 Parameter learning

The trainable parts of the architecture are the weights and parameters of the neighborhood aggregator f_A, the transformer network f_E, the classifier f_C and the embedding representations $\{x_v : v \in V\}$.

To obtain suitable representations, we employ a combination of supervised and contrastive loss functions on the output of the hierarchical encoder layer $h_{i,j}^{[0,T]}$. The contrastive loss function encourages the embeddings of positive (i.e. a link exists in E^T) node pairs to be closer while ensuring that the embeddings of negative node pairs are distinct.

We adopt a contrastive learning framework (Chen et al. 2020) to distinguish between positive and negative classes. For brevity, we temporarily denote $h_{i,j}^{[0,T]}$ as $h_{i,j}$. Given two positive node pairs with corresponding embeddings $e(v_i, v_j) \rightarrow h_{i,j}$ and $e(v_o, v_n) \rightarrow h_{o,n}$, the loss function is defined as follows:

$$\mathcal{S}(e(v_i, v_j), e(v_o, v_n)) = -\log \frac{\exp(\mathrm{sim}(h_{i,j}, h_{o,n})/\alpha)}{\sum_{\substack{(k,w) \in B \\ \mathbb{1}_{(k,w)\neq(i,j)}}} \exp(\mathrm{sim}(h_{i,j}, h_{k,w})/\alpha)}, \tag{6}$$

where α represents a temperature parameter, B is a set of node pairs in a given batch, and $\mathbb{1}_{(k,w)\neq(i,j)}$ indicates that the labels of node pair (k, w) and (i, j) are different. We employ the angular similarity function $\mathrm{sim}(x) = 1 - \arccos(x)/\pi$. We do not explicitly sample negative examples, following the methodology outlined in Chen et al. (2020).

The contrastive loss is summed over the positive training data E^+:

$$\mathcal{L}_c = \sum_{e(v_i,v_j),e(v_o,v_n)\in E^+} \mathcal{S}(e(v_i, v_j), e(v_o, v_n)). \tag{7}$$

To further improve the discriminative power of the learned features, we also minimize the center loss:

$$\mathcal{L}_d = \frac{1}{2} \sum_{(i,j)\in E} \|h_{i,j}^{[0,T]} - c_{y_{i,j}}\|_2^2, \tag{8}$$

where E is the data of positive and negative edges, $y_{i,j}$ is the class of the pair (0 or 1), $c_{y_{i,j}} \in R^d$ denotes the corresponding class center. The class centers are updated after each mini-batch step following the method proposed in Wen et al. (2016).

Finally, a good node pair vector $h_{i,j}^{[0,T]}$ should minimize the binary cross entropy loss of the node pair prediction task:

$$\mathcal{L}_s = -\frac{1}{|E^+| + |E^-|} \left[\sum_{(i,j)\in E^+} \log \left(f_C(h_{i,j}^{[0,T]}) \right) + \sum_{(o,n)\in E^-} \log \left(1 - f_C(h_{o,n}^{[0,T]}) \right) \right] \tag{9}$$

We adopt the joint supervision of the prediction loss, contrastive loss, and center loss to jointly train the model for discriminative feature learning and relationship inference:

$$\mathcal{L} = \mathcal{L}_c + \mathcal{L}_d + \mathcal{L}_s. \tag{10}$$

As is usual, the losses are applied over subsets of the entire dataset. In this case, we have an additional requirement for pairs of nodes in E^-: at least one of the two nodes needs to appear in E^{+}. An elaborate batch sampling strategy is proposed in the following section. The model parameters are trained end to end.

Algorithm 3 Training Procedure in THiGER-A

Require: $G = \{V, E\}$, link predictor $f(.; \boldsymbol{\theta} = \{\theta_A, \theta_E, \theta_C\})$, sampling size k

$\theta_E^0 \leftarrow \theta_E$

$E^- \leftarrow (V \times V) \setminus E$ ▷ Entire pool of negative samples

for $m \leftarrow 1 \ldots M$ **do** ▷ Incremental training round

    $B_U = \underset{B_U \subset E^- \setminus E^-_{m-1}}{\operatorname{argmax}} \sum_{e^-_{ij}} S_{AL}(e^-_{ij})$ ▷ Uncertain -ve subset (Eq. 12)

    Sample a diverse negative subset $B_N^* \subset B_U$ using Eq 13

    $B_P^* = \underset{B_P \subset E \setminus E^+_{m-1}}{\operatorname{argmax}} \sum_{e_{ij}} S_{CL}(e_{ij})$ ▷ Uncertain +ve subset (Eq. 14)

    $E^-_m \leftarrow E^-_{m-1} \cup B_N^*$

    $E^+_m \leftarrow E^+_{m-1} \cup B_P^*$

    Train a model starting from $f(.; \boldsymbol{\theta}^{m-1})$ on $E^-_m \cup E^+_m$, and obtain updated parameters $\boldsymbol{\theta}^m$

end for

return $\boldsymbol{\theta}^M$

3.6 Incremental training strategy

This section introduces the incremental training strategy *THiGER-A*, which extends our base *THiGER* model. The pseudo-code for *THiGER-A* is presented in Algorithm 3. We represent the parameters used in the entire architecture as $\boldsymbol{\theta} = (\theta_A, \theta_E, \theta_C)$. Let $P(y \mid e_{i,j}; \boldsymbol{\theta})$, where $y \in \{0, 1\}$, denote the link predictor for the nodes (v_i, v_j). Specifically, in shorthand, we denote $P(y = 1 \mid e_{i,j}; \boldsymbol{\theta})$ by $p_{i,j}$ as in line 3 of Algorithm 2, likewise $P(y = 0 \mid e_{i,j}; \boldsymbol{\theta}) = 1 - p_{i,j}$.

We define $E^- = (V \times V) \setminus E$ as the set of negative edges representing non-observed connections in the graph. The size of the negative set grows quadratically with the number of nodes, resulting in a computational complexity of $\mathcal{O}(|V|^2)$. For large, sparse graphs like ours, the vast number of negative edges makes it impractical to use all of them for model training.

Randomly sampling negative examples may introduce noise and hinder training convergence. To address this challenge, we propose an approach to sample a smaller subset of "informative" negative edges that effectively capture the entity relationships within the graph. Leveraging active learning, a technique for selecting high-utility datasets, we aim to choose a subset $B_N^* \subset E^-$ that leads to improved model learning.

3.6.1 Negative Edge Sampling using Active Learning

Active learning (AL) is an iterative process centered around acquiring a high-utility subset of samples and subsequently retraining the model. The initial step involves selecting a subset of samples with high utility, determined by a specified informativeness measure.

Once this subset is identified, it is incorporated into the training data, and the model is subsequently retrained. This iterative cycle, involving sample acquisition and model retraining, aims to improve the model's performance and generalizability through the learning process.

In this context, we evaluate the informativeness of edges using a score function denoted as $S_{AL} : (v_i^-, v_j^-) \rightarrow \mathbb{R}$. An edge (v_i^-, v_j^-) is considered more informative than (v_k^-, v_l^-) if $S_{AL}(v_i^-, v_j^-) > S_{AL}(v_k^-, v_l^-)$. The key challenge in AL lies in defining S_{AL}, which encodes the learning of the model $P(.;\boldsymbol{\theta})$ trained in the previous iteration.

We gauge the informativeness of an edge based on model uncertainty. An edge is deemed informative when the current model $P(.;\boldsymbol{\theta})$ exhibits high uncertainty in predicting its label. Uncertainty sampling is one of the most popular choices for the quantification of informativeness due to its simplicity and high effectiveness in selecting samples for which the model lacks sufficient knowledge. Similar to various previous techniques, we use Shannon entropy to approximate informativeness (Priyadarshini et al. 2021; Kirsch et al. 2019). It is important to emphasize that ground truth labels are unavailable for negative edges, which represent unobserved entity connections. Therefore, to estimate the informativeness of negative edges, we calculate the expected Shannon entropy across all possible labels. Consequently, the expected entropy for a negative edge (v_i^-, v_j^-) at the m^{th} training round is defined as:

$$S_{AL}(e_{ij}^-) = -\sum_{y \in \{0,1\}} P(y \mid e_{ij}^-;\boldsymbol{\theta}^{m-1}) \log P(y \mid e_{ij}^-;\boldsymbol{\theta}^{m-1}) \tag{11}$$

$$B_U = \underset{B_U \subset E^- \setminus E_{m-1}^-}{\operatorname{argmax}} \sum_{e_{ij}^-} S_{AL}(e_{ij}^-) \tag{12}$$

Here, $\boldsymbol{\theta}^{m-1}$ is the base hypothesis predictor model trained at the $(m-1)^{th}$ training round and $m = 0, 1, \cdots, M$ denotes the AL training round. Selecting a subset of uncertain edges, B_U using Eq. 12 unfortunately does not ensure diversity among the selected subset. The diversity metric is crucial in subset selection as it encourages the selection of diverse samples within the embedding space. This, in turn, results in a higher cumulative informativeness for the selected subset, particularly when the edges exhibit overlapping features. The presence of a highly-correlated edges in the selected subset can lead to a sub-optimal batch with high redundancy. The importance of diversity in selecting informative edges has been emphasized in several prior works (Kirsch et al. 2019; Priyadarshini et al. 2021). To obtain a diverse subset, both approaches aim to maximize the joint entropy (and consequently, minimize mutual information) among the samples in the selected batch. However, maximizing joint entropy is an expensive combinatorial optimization problem and does not scale well for larger datasets, as in our case.

We adopt a similar approach as (Kumari et al. 2020) and utilize the k-means++ algorithm (Arthur and Vassilvitskii 2006) to cluster the selected batch B_U into diverse landmark points. While (Kumari et al. 2020) is tailored for metric learning tasks with the triplet samples as inputs, our adaptation of the k-means++ algorithm is designed for graph datasets, leading to the selection of diverse edges within the gradient space. Although diversity in the gradient space is effective for gradient-based optimizer, a challenge arises due to the high dimensionality of the gradient space, particularly when the model is large. To overcome this challenge, we compute the expected gradient of the loss function with respect to only the penultimate layer of the network, $\nabla_{\theta_{out}} \mathcal{L}_{e_{ij}^-}$, assuming it captures task-specific features. We begin to construct an optimal subset $B_N^* \in B_U$ by initially (say, at $k = 0$) selecting the two edges with the most distinct gradients. Subsequently, we iteratively select the most dissimilar gradient edge from the selected subset using the maxmin optimization objective defined in Eq. 13.

$$B_k^* = B_{k-1}^* \cup \operatorname{argmax}_{e_{ij}^- \in B_U \backslash B_{k-1}^*} \operatorname{argmin}_{e_{kw}^- \in B_{k-1}^*} d_E(\nabla_{\theta_{out}} \mathcal{L}_{e_{ij}^{-1}}, \nabla_{\theta_{out}} \mathcal{L}_{e_{kw}^{-1}}) \tag{13}$$

Here d_E represents the Euclidean distance between two vectors in the gradient space, consisting of $\nabla_{\theta_{out}} \mathcal{L}_{e_{ij}^{-1}}$, which denotes the gradient of the loss function $\mathcal{L}$ with respect to the penultimate layer of the network θ_{out}. The process continues until we reach the allocated incremental training budget, $|B_N^*| = K$. The resulting optimal subset of negative edges, B_N^*, comprises negative edges that are both diverse and informative.

3.6.2 Positive Edge Sampling

Inspired by Curriculum Learning (CL), a technique mimicking certain aspects of human learning, we investigate its potential to enhance the performance and generalization of the node pair predictor model. Curriculum Learning involves presenting training data to the model in a purposeful order, starting with easier examples and gradually progressing to more challenging ones. We hypothesize that applying CL principles can benefit our node pair predictor model. By initially emphasizing the learning of simpler connections and leveraging prior knowledge, the model can effectively generalize to more complex connections during later stages of training. Although Active Learning (AL) and CL both involve estimating the utility of training samples, they differ in their approach to label availability. AL operates in scenarios where labels are unknown and estimates sample utility based on expected scores. In contrast, CL uses known labels to assess sample difficulty. For our model, we use one of the common approaches to define a difficulty score S_{CL} based on the model's prediction confidence. The model's higher prediction confidence indicates easier samples.

$$S_{CL}(e_{ij}) = -\sum_{e_{ij}} P(y = 1 \mid e_{ij}; \boldsymbol{\theta}^{m-1}) \log P(y = 1 \mid e_{ij}; \boldsymbol{\theta}^{m-1}) \tag{14}$$

Here, $S_{CL}(v_i, v_j)$ indicates predictive uncertainty of an edge e_{ij} to be positive by an existing trained model θ^{m-1} at $(m-1)^{th}$ iteration. In summary, for hypothesis prediction using a large training dataset, Active Curriculum Learning provides a natural approach to sample an informative and diverse subset of high-quality samples, helping to alleviate the challenges associated with label bias.

4 Experimental setup

In this section, we present the experimental setup for our evaluation. We compare our proposed model, THiGER(-A), against several state-of-the-art (SOTA) methods to provide context for the empirical results on benchmark datasets. To ensure fair comparisons, we utilize publicly available baseline implementations and modify those as needed to align with our model's configuration and input requirements. All experiments were conducted using Python. For the evaluation of the interaction datasets, we train all models on a single NVIDIA A10G GPU. In the case of the food-related biomedical dataset, we employ 4 NVIDIA V100 GPUs for model training. Notably, all models are trained on single machines. In our experiments, we consider graphs as undirected. The node attribute embedding dimension is set to $d = 128$ for all models evaluated. For baseline methods, we performed a parameter search on the learning rate and training steps, and we report the best results achieved. Our model is implemented in TensorFlow.

4.1 Datasets and model setup

Table 1 shows the statistics of the datasets used in this study. Unless explicitly mentioned, all methods, including our model, share the same initial node attributes provided by pre-trained Node2Vec (Grover and Leskovec 2016). The pre-trained Node2vec embedding effectively captures the structural information of nodes in the training graph. In our proposed framework, the choice of a fixed node embedding is to enable the model capture the temporal evolution of network relations, given that the node embeddings are in the same vector space. While employing a dynamic node embedding framework may enhance results, it introduces complexities associated with aligning vector spaces across different timestamps. This aspect is deferred to future research. It is important to note that the Node-2vec embeddings serve solely as initializations for the embedding layer, and the embedding vectors undergo fine-tuning during the learning process to further capture the dynamic evolution of node relationships. For models that solely learn embedding vectors for individual nodes, we represent the $h_{i,j}$ of a given node pair as the concatenation of the embedding vectors for nodes $\langle x_i, x_j \rangle$.

4.1.1 Interaction datasets

We have restructured the datasets to align with our specific use case. We partition the edges in the temporal graphs into five distinct groups based on their temporal labels. For example, if a dataset is labeled up to 500 time units, we reorganize them as follows: $\{0, \dots, 100\} \rightarrow 0$, $\{101, \dots, 200\} \rightarrow 1$, $\{201, \dots, 300\} \rightarrow 2$, $\{301, \dots, 400\} \rightarrow 3$, and $\{401, \dots, 500\} \rightarrow 4$. These User-Item based datasets create bipartite graphs. For all inductive evaluations, we assume knowledge of three nearest node neighbors for each of the unseen nodes. Neighborhood information is updated after model training to incorporate this knowledge, with zero vectors assigned to new nodes.

Table 1 Statistics of the evaluation datasets

	Number of Edges	Number of Nodes
Interaction		
Wikipedia	21,905	6257
Reddit	151,294	10,984
LastFM	257,941	1980
Biomedical		
All	5,674,515	84,735
Ingredient—Disease	4,868,208	76,076
Ingredient—Chemical	4,710,062	78,996

4.1.2 Food-related biomedical temporal datasets

To construct the relationship graph, we extract sentences containing predefined entities (Genes, Diseases, Chemical Compounds, Nutrition, and Food Ingredients). We establish connections between two concepts that appear in the same sentence within any publication in the dataset. The time step for each relationship between concept pairs corresponds to the publication year when the first mention was identified (i.e., the oldest publication year among all the publications where the concepts are associated). We generate three datasets for evaluation based on concept pair domains: *Ingredient, disease* pairs, *Ingredient, Chemical compound* pairs, and all pairs (unfiltered). Graph statistics are provided in Table 1. For training and testing sets, we divide the graph into 10-year intervals starting from 1940 (i.e., $\{\leq 1940\}$, $\{1941–1950\}$, $\dots$, $\{2011–2020\}$). The splits ≤ 2020 are used for training, and the split $\{2021–2022\}$ is used for testing. In accordance with the problem configuration in the interaction dataset, we update the neighborhood information and also assume knowledge of three nearest node neighbors pertaining to each of the unseen nodes for inductive evaluations.

4.1.3 Model setup & parameter tuning

Model Configuration: We employ a hierarchical encoder with $N\lceil \log_2 T \rceil$ layers, where N is a multiple of each hierarchical layer (i.e., with $N-1$ encoder sublayers), and T represents the number of time steps input to each hierarchical encoder layer. In our experiments, we set the number of encoder layer multiples to $N = 2$. We use 8 attention heads with 128 dimensional states. For the position-wise feed-forward networks, we use 512 dimensional inner states. For the activation function, we applied the Gaussian Error Linear Unit (GELU, Hendrycks and Gimpel 2016). We apply a dropout (Srivastava et al. 2014) to the output of each sub-layer with a rate of $P_{drop} = 0.1$.

Optimizer: Our models are trained using the AdamW optimizer (Loshchilov and Hutter 2017), with the following hyper-parameters: $\beta_1 = 0.9$, $\beta_2 = 0.99$, and $\epsilon = 10^{-7}$. We use a linear decay of the learning rate. We set the number of warmup steps to 10% of the number of train steps. We vary the learning rate with the size of the training data.

Time Embedding: We use Time2Vec (T2V, Kazemi et al. 2019) to generate time-step embeddings which encode the temporal sequence of the time steps. The T2V model is learned and updated during the model training.

Active learning: The size of subset B_U is twice the size of the optimal subset B^*. The model undergoes seven training rounds for the Wikipedia, Reddit, and Last FM datasets, while it is trained for three rounds for the food-related biomedical dataset (All, Ingredient-Disease, Ingredient-Chemical). Due to the large size of biomedical dataset, we limit the model training to only three rounds. However, we anticipate that increasing the number of training rounds will lead to further improvements in performance.

4.2 Evaluation metrics

In this study, we assess the efficacy of the models by employing the binary F1 score and average precision score (AP) as the performance metrics. The binary F1 score is defined as the harmonic mean of precision and recall, represented by the formula:

$$F1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{15}$$

Here, precision denotes the ratio of true positive predictions to the total predicted positives, while recall signifies the ratio of true positive predictions to the total actual positives.

The average precision score is the weighted mean of precisions achieved using different thresholds, using the incremental change in recall from the previous threshold as weight:

$$AP = \sum_{k=1}^{N} P_k \cdot \Delta R_k, \tag{16}$$

where N is the total number of thresholds, P_k is the precision at cut-off k, and $\Delta R_k = R_k - R_{k-1}$ is a sequential change in the recall value. Our emphasis on positive predictions in the evaluations is driven by our preference for models that efficiently forecast future connections between pairs of nodes.

4.3 Method categories

We categorize the methods into two main groups based on their handling of temporal information:

Static Methods: These methods treat the graph as static data and do not consider the temporal aspect. The static methods under consideration include the Logistic regression model, GraphSAGE (Hamilton et al. 2017), and AGATHA (Sybrandt et al. 2020).

Temporal Methods: These state-of-the-art methods leverage temporal information to create more informative node representations. We evaluate the performance of our base model, THiGER, and the final model, THiGER-A, against the following temporal methods: CTDNE (Nguyen et al. 2018), TGN (Rossi et al. 2020), JODIE (Kumar et al. 2019), TNodeEmbed (Singer et al. 2019), DyRep (Trivedi et al. 2019), T-PAIR (Akujuobi et al. 2020b), and TDE (Zhou et al. 2022).

5 Experiments

The performance of THiGER-A is rigorously assessed across multiple benchmark datasets, as presented in Tables 2 and 3. The experimental evaluations are primarily geared toward two distinct objectives:

1. Assessing the model's effectiveness in handling interaction datasets pertinent to temporal graph problems.
2. Evaluating the model's proficiency in dealing with food-related biomedical datasets, specifically for predicting relationships between food-related concepts and other biomedical terms.

In Sects. 4.1.1 and 4.1.2, a comprehensive overview of the used datasets is provided. Our evaluations encompass two fundamental settings:

Table 2 Binary F1 score (F1) and Average Precision (AP) for future node pair prediction task on the standard datasets in transductive and inductive settings

	Transductive						Inductive					
	Wikipedia		Reddit		Last FM		Wikipedia		Reddit		Last FM	
	F1	AP	F1	AP	F1	AP	F1	AP	F1	AP	F1	AP
Logistic Regression*	0.30	0.19	0.41	0.25	0.05	0.16	0.09	0.05	0.59	0.42	0.44	0.28
GraphSAGE*	0.56	0.38	0.76	0.60	0.49	0.32	0.15	0.07	***0.79***	***0.65***	0.34	0.21
AGATHA*	***0.92***	***0.86***	0.83	*0.72*	0.62	0.44	0.00	0.01	0.05	0.12	0.00	0.16
CTDNE	0.78	0.64	0.72	0.55	0.34	0.24	0.02	0.02	0.59	0.39	0.01	0.16
TGN	0.46	0.29	0.69	0.52	0.35	0.22	0.15	0.07	0.69	0.52	**0.47**	**0.30**
JODIE	0.38	0.23	0.26	0.16	0.22	0.17	0.09	0.04	0.34	0.19	0.33	0.20
tNodeEmbed	0.64	0.48	0.78	0.65	*0.69*	*0.53*	0.18	0.05	0.75	0.60	0.00	0.16
DyRep	0.35	0.22	0.27	0.16	0.06	0.16	0.08	0.04	0.38	0.23	0.21	0.16
T-PAIR	0.29	0.17	0.42	0.27	0.55	0.38	0.04	0.02	0.34	0.21	***0.54***	***0.37***
TDE	0.65	0.48	0.13	0.17	0.39	0.25	0.09	0.03	0.13	0.13	0.31	0.20
THiGER-attn	0.78	*0.69*	0.83	*0.72*	*0.69*	0.51	***0.56***	***0.32***	0.54	0.38	0.02	0.16
THiGER-gin	0.75	0.65	0.83	**0.72**	*0.70*	***0.54***	0.07	0.02	0.29	0.20	0.01	0.16
THiGER-mean	0.71	0.54	0.82	0.70	**0.69**	*0.53*	0.17	0.06	0.73	0.57	0.10	0.16
THiGER-A-attn	**0.80**	**0.69**	***0.85***	***0.74***	***0.70***	**0.53**	0.23	0.07	0.68	0.51	0.19	0.17
THiGER-A-gin	0.79	0.68	***0.85***	***0.74***	*0.70*	0.52	0.18	0.14	0.47	0.30	0.39	0.25
THiGER-A-mean	0.74	0.60	**0.84**	***0.74***	*0.70*	*0.53*	**0.28**	**0.10**	**0.78**	**0.63**	0.28	0.20

*Static graph method

Showing the ***best*** and **second best** models

Table 3 Binary F1 score (F1) and Average Precision (AP) for future node pair prediction task on the standard datasets in transductive and inductive settings

	Transductive						Inductive					
	F-A		F-ID		F-IC		F-A		F-ID		F-IC	
	F1	AP	F1	AP	F1	AP	F1	AP	F1	AP	F1	AP
Logistic Regression*	0.71	0.55	0.70	0.53	0.67	0.50	0.50	0.29	0.50	0.29	0.54	0.33
GraphSAGE*	0.75	0.60	0.76	0.61	0.75	0.61	0.00	0.08	0.00	0.08	0.00	0.08
AGATHA*	0.86	0.75	0.83	0.71	0.84	0.72	0.61	0.40	0.61	0.41	0.59	0.38
CTDNE	-	-	-	-	-	-	-	-	-	-	-	-
TGN	0.31	0.18	0.35	0.21	0.32	0.19	0.29	0.14	0.27	0.13	0.27	0.13
JODIE	0.31	0.18	0.34	0.20	0.54	0.35	0.19	0.10	0.09	0.07	0.02	0.08
tNodeEmbed	0.54	0.38	0.56	0.40	0.59	0.46	0.31	0.15	0.32	0.16	0.31	0.17
DyRep	0.46	0.29	0.30	0.18	0.32	0.19	0.05	0.07	0.17	0.09	0.18	0.09
T-PAIR	0.38	0.23	0.44	0.29	0.47	0.31	0.21	0.12	0.25	0.14	0.27	0.16
TDE	0.53	0.43	0.67	0.51	0.65	0.47	0.00	0.08	0.00	0.08	0.00	0.08
THiGER-attn	*0.96*	0.92	*0.95*	**0.91**	***0.95***	0.91	0.61	0.45	***0.63***	***0.46***	0.55	0.37
THiGER-gin	***0.96***	*0.93*	***0.95***	0.90	0.93	0.88	0.60	0.43	0.47	0.30	0.39	0.23
THiGER-mean	***0.97***	***0.94***	*0.95*	***0.92***	***0.96***	***0.92***	***0.64***	***0.48***	0.62	0.44	***0.64***	***0.47***
THiGER-A-attn	***0.97***	***0.93***	***0.95***	**0.91**	***0.96***	**0.92**	**0.69**	*0.48*	**0.65**	**0.49**	*0.55*	*0.35*
THiGER-A-gin	***0.97***	***0.94***	***0.96***	***0.92***	*0.93*	***0.91***	***0.67***	**0.53**	*0.56*	*0.37*	*0.44*	*0.28*
THiGER-A-mean	***0.97***	***0.93***	***0.96***	***0.92***	***0.96***	***0.93***	***0.74***	***0.58***	***0.67***	***0.50***	***0.71***	***0.55***

*Static graph method

Showing the **best** and **second best** models on the food-related biomedical dataset

1. Transductive setup: This scenario involves utilizing data from all nodes during model training.
2. Inductive setup: In this configuration, at least one node in each evaluated node pair has not been encountered during the model's training phase.

These experiments are designed to rigorously assess THiGER-A's performance across diverse datasets, offering insights into its capabilities under varying conditions and problem domains.

5.1 Quantitative evaluation: interaction temporal datasets

We assess the performance of our proposed model in the context of future interaction prediction (Rossi et al. 2020; Kumar et al. 2019). The datasets record interactions between users and items.

We evaluate the performance on three distinct datasets: (i) Reddit, (ii) LastFM, and (iii) Wikipedia, considering both transductive and inductive settings. In the transductive setting, THiGER-A outperforms other models across all datasets, except Wikipedia, where AGATHA exhibits significant superiority. Our analysis reveals that AGATHA's advantage lies in its utilization of the entire graph for neighborhood and negative sampling, which gives it an edge over models using a subset of the graph due to computational constraints. This advantage is more evident in the transductive setup since AGATHA's training strategy leans towards seen nodes. Nevertheless, THiGER-A consistently achieves comparable or superior performance even in the presence of AGATHA's implicit bias. It is imperative to clarify that AGATHA was originally designed for purposes other than node-pair predictions. Nonetheless, we have adapted the algorithm to align with the node-pair configuration specifically for our research evaluations.

In the inductive setup, our method excels in the Wikipedia and Reddit datasets but lags behind some baselines in the LastFM dataset. Striking a balance between inductive and transductive performance, THiGER-A's significant performance gain over THiGER underscores the effectiveness of the proposed incremental learning strategy. This advantage is particularly pronounced in the challenging inductive test setting.

5.2 Quantitative evaluation: food-related biomedical temporal datasets

This section presents the quantitative evaluation of our proposed model on temporal node pair (or "link") prediction, explicitly focusing on food-related concept relationships extracted from scientific publications in the PMC dataset. The evaluation encompasses concept pairs from different domains, including *Ingredient, Disease* pairs (referred to as F-ID), *Ingredient, Chemical Compound* pairs (F-IC), and all food-related pairs (F-A). The statistical characteristics of the dataset are summarized in Table 1.

Table 3 demonstrates that our model outperforms the baseline models in both inductive and transductive setups. The second-best performing model is AGATHA, which, as discussed in the previous section, exhibits certain advantages over alternative methods. It is noteworthy that the CTDNE method exhibits scalability issues with larger datasets.

An intriguing observation from this evaluation is that, aside from our proposed model, static methods outperform temporal methods on this dataset. Further investigation revealed that the data is more densely distributed toward the later time steps. Notably, a substantial

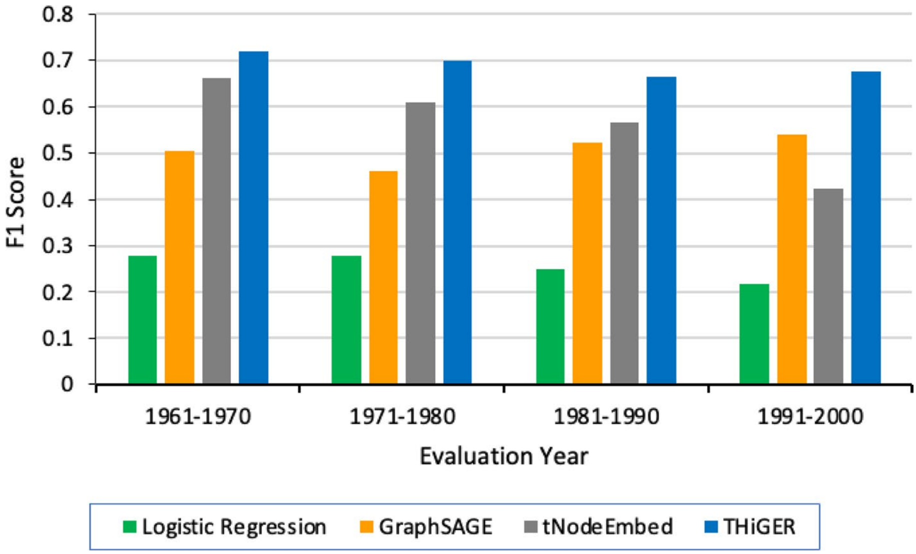


Fig. 4 Transductive F1 score of incremental prediction (per year) made by THiGER and three other baselines. The models are incrementally trained with data before the displayed evaluation time window

increase in information occurs during the last time steps. Up to the year 2000, the average number of edges per time step is approximately 100, 000. However, this number surges to about 1 million in the time window from 2001 to 2010, followed by another leap to around 4 million in the 2011–2020 time step. This surge indicates a significant influx of knowledge in food-related research in recent years.

We hypothesize that while this influx is advantageous for static methods, it might adversely affect some temporal methods due to limited temporal information. To test this hypothesis, we conduct an incremental evaluation, illustrated in Fig. 4, using two comparable link prediction methods (Logistic Regression and GraphSAGE) and the two best temporal methods (tNodeEmbed and THiGER). In this evaluation, we incrementally assess the transductive performance on testing pairs up to the year 2000. Specifically, we evaluate the model performance on the food dataset (F-A) in the time intervals 1961–1970 by using all available training data up to 1960, and similarly for subsequent time intervals.

From Fig. 4, it is evident that temporal methods outperform static methods when the temporal data is more evenly distributed, i.e., when there is an incremental increase in temporal data. The sudden exponential increase in data during the later years biases the dataset towards the last time steps. However, THiGER consistently outperforms the baseline methods in the incremental evaluation, underscoring its robustness and flexibility.

5.3 Ablation study

In this section, we conduct an ablation study to assess the impact of various sampling strategies on the base model's performance. The results are presented in Table 4, demonstrating the performance improvements achieved by the different versions of the THiGER model

Table 4 Ablation study to show a performance comparison of individual components of our framework

Model	Transductive		Inductive	
	F1	AP	F1	AP
Wikipedia				
THiGER-gin	0.75	0.65	0.07	0.02
THiGER-gin + AL	**0.79**	0.67	0.16	0.12
THiGER-gin + AL + CL	**0.79**	**0.68**	**0.18**	**0.14**
Last FM				
THiGER-mean	0.69	**0.53**	0.10	0.16
THiGER-mean + AL	**0.70**	**0.53**	0.17	0.18
THiGER-mean + AL + CL	**0.70**	**0.53**	**0.28**	**0.20**
Reddit				
THiGER-attn	0.83	0.72	0.54	0.38
THiGER-attn + AL	**0.85**	**0.74**	0.67	0.46
THiGER-attn + AL + CL	**0.85**	**0.74**	**0.68**	**0.51**

Our proposed model is THiGER-A (i.e., THiGER + AL + CL)

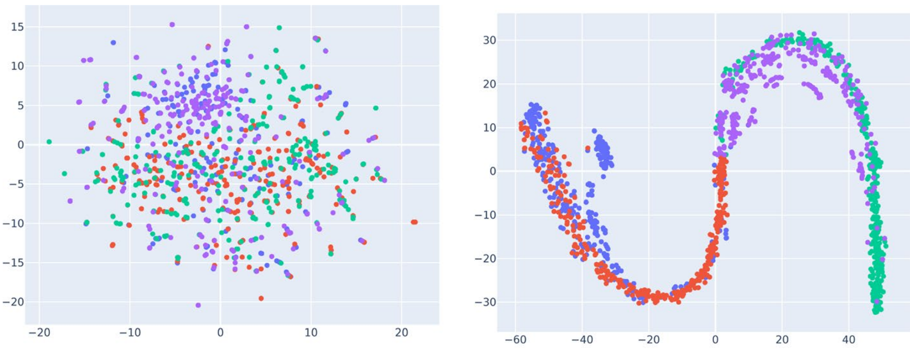
(-mean, -gin and -attn) for each dataset. Due to the much larger size of the food-related biomedical dataset, we conduct the ablation study only for the baseline datasets.

First, we investigate the influence of the active learning (AL)-based negative sampler on the base THiGER model. A comparison of the model's performance with and without the AL-based negative sampler reveals significant improvements across all datasets. Notably, the performance gains are more pronounced in the challenging inductive test cases where at least one node of an edge is unseen in the training data. This underscores the effectiveness and generalizability of the AL-based learner for the hypothesis prediction model in the positive-unlabeled (PU) learning setup.

Next, we integrate curriculum learning (CL) as a positive data sampler, resulting in further enhancements to the base model. Similar to the AL-based sampling, the performance gains are more pronounced in the inductive test setting. The relatively minor performance improvement in the transductive case may be attributed to the limited room for enhancement in that specific context. Nevertheless, both AL alone and AL combined with CL enhance the base model's performance and generalizability, particularly in the inductive test scenario.

5.4 Pair embedding visualization

In this section, we conduct a detailed analysis of the node pair embeddings generated by THiGER using the F-ID dataset. To facilitate visualization, we randomly select 900 pairs and employ t-SNE (Van der Maaten and Hinton 2008) to compare these embeddings with those generated by Node2Vec, as shown in Fig. 5. We employ color-coding to distinguish between the observed labels and the predicted labels. Notably, we observe distinct differences in the learned embeddings. THiGER effectively separates positive and negative node pairs in the embedding space. True positives (denoted in green) and true negatives (denoted in blue) are further apart in the embedding space, while false positives (indicated in red) and false negatives (shown in purple) occupy an intermediate region.



(a) Node2Vec (b) THiGER

Fig. 5 Pair embedding visualization. The blue color denotes the true negative samples, the red points are false negative, the green points are true positive, and the purple points are false positive

This observation aligns with the idea that unknown connections are not unequivocal in our application domain, possibly due to missing data or discoveries yet to be made.

5.5 Case study

To assess the predictive accuracy of our model, we conducted a detailed analysis using the entire available food-related biomedical temporal dataset. We collaborated with biologists to evaluate the correctness of the generated hypotheses. Unlike providing binary predictions (1 or 0), we take a probabilistic approach by assigning a probability score within the range of 0 to 1. This score reflects the likelihood of a connection existing between the predicted node pairs. Consequently, the process of ranking a set of relation predictions associated with a specific node is tantamount to ranking the corresponding predicted probabilities.

Using this methodology, we selected 402 node pairs and presented them to biomedical researchers for evaluation. The researchers sought hypotheses related to specific oils. Subsequently, we generated hypotheses representing potential future connections between the oil nodes and other nodes, resulting in a substantial list. Given the anticipated extensive list, we implemented a filtering process based on the associated probability scores. This enabled us to selectively identify predictions with high probabilities, which were then communicated to the biomedical researchers for evaluation. The evaluation encompassed two distinct approaches.

First, they conducted manual searches for references to the predicted positive node pairs in various biology texts, excluding our dataset. Their findings revealed relationships in 70 percent of the node pairs through literature searches and reviews.

Secondly, to explore cases where no direct relationship was apparent in existing literature, they randomly selected and analyzed three intriguing node pairs: (i) *Flaxseed oil and Root caries*, (ii) *Benzoxazinoid and Gingelly oil*, and (iii) *Senile osteoporosis and Soybean oil*.

5.5.1 Flaxseed oil and root caries

Root caries refers to a dental condition characterized by the decay and demineralization of tooth root surfaces. This occurs when tooth roots become exposed due to gum recession, allowing bacterial invasion and tooth structure erosion. While the scientific literature does not explicitly mention the use of flaxseed oil for root caries, it is well-established that flaxseed oil possesses antibacterial properties (Liu et al. 2022). These properties may inhibit bacterial species responsible for root caries. Furthermore, flaxseed oil is a rich source of omega-3 fatty acids and lignans, factors potentially relevant to this context. Interestingly, observational studies are investigating the oil's effects on gingivitis (Deepika 2018).

5.5.2 Benzoxazinoid and gingelly oil

Benzoxazinoids are plant secondary metabolites synthesized in many monocotyledonous species and some dicotyledonous plants (Schullehner et al. 2008). Gingelly oil, derived from sesame seeds, originates from a dicotyledonous plant. In the biologists' opinion, this concurrence suggests a valid basis for the hypothesized connection.

5.5.3 Senile osteoporosis and soybean oil

Senile osteoporosis is a subtype of osteoporosis occurring in older individuals due to age-related bone loss. Soybean oil, a common vegetable oil derived from soybeans, contains phytic acid (Anderson and Wolf 1995). Phytic acid is known to inhibit the absorption of certain minerals, including calcium, which is essential for bone strength (Lönnerdal et al. 1989). Again, in the experts' opinion, this suggests a valid basis for a (unfortunately detrimental) connection between the oil and the health condition.

6 Conclusions

We introduce an innovative approach to tackle the hypothesis generation problem within the context of temporal graphs. We present THiGER, a novel transformer-based model designed for node pair prediction in temporal graphs. THiGER leverages a hierarchical framework to effectively capture and learn from temporal information inherent in such graphs. This framework enables efficient parallel temporal information aggregation. We also introduce THiGER-A, an incremental training strategy that enhances the model's performance and generalization by training it on high-utility samples selected through active curriculum learning, particularly benefiting the challenging inductive test setting. Quantitative experiments and analyses demonstrate the efficiency and robustness of our proposed method when compared to various state-of-the-art approaches. Qualitative analyses illustrate its practical utility.

For future work, an enticing avenue involves incorporating additional node-pair relationship information from established biomedical and/or food-related knowledge graphs. In scientific research, specific topics often experience sudden exponential growth, leading to temporal data distribution imbalances. Another intriguing research direction, thus, is the study of the relationship between temporal data distribution and the performance of temporal graph neural network models. We plan to analyze the performance of several temporal

GNN models across diverse temporal data distributions and propose model enhancement methods tailored to such scenarios.

Due to the vast scale of the publication graph, training the hypothesis predictor with all positive and negative edges is impractical and limits the model's ability to generalize, especially when the input data is noisy. Thus, it is crucial to train the model selectively on a high-quality subset of the training data. Our work presents active curriculum learning as a promising approach for feasible and robust training for hypothesis predictors. However, a static strategy struggles to generalize well across different scenarios. An exciting direction for future research could be to develop dynamic policies for data sampling that automatically adapt to diverse applications. Furthermore, improving time complexity is a critical challenge, particularly for applications involving large datasets and models.

Author Contributions U.A. and P.K. co-lead the reported work and the writing of the manuscript, J.C., S.B., K.M., and S.P. supported the work and the writing of the manuscript. T.B. supervised the work overall. All authors reviewed the manuscript and contributed to the revisions based on the reviewers' feedback.

Declarations

Conflict of interest The authors have no Conflict of interest to declare that are relevant to the content of this article.

References

Ahmed NM, Chen L, Wang Y et al. (2016) Sampling-based algorithm for link prediction in temporal networks. Inform Sci 374:1–14

Akujuobi U, Chen J, Elhoseiny M et al. (2020) Temporal positive-unlabeled learning for biomedical hypothesis generation via risk estimation. Adv Neural Inform Proc Syst 33:4597–4609

Akujuobi U, Spranger M, Palaniappan SK et al. (2020) T-pair: Temporal node-pair embedding for automatic biomedical hypothesis generation. IEEE Trans Knowledge Data Eng 34(6):2988–3001

Anderson RL, Wolf WJ (1995) Compositional changes in trypsin inhibitors, phytic acid, saponins and isoflavones related to soybean processing. J Nutr 125(suppl–3):581S-588S

Arthur D, Vassilvitskii S (2006) *k*-means++: The advantages of careful seeding. Stanford University, Tech. rep

Ash JT, Zhang C, Krishnamurthy A et al. (2020) Deep batch active learning by diverse, uncertain gradient lower bounds. ICLR, Vienna

Baek SH, Lee D, Kim M et al. (2017) Enriching plausible new hypothesis generation in pubmed. PloS One 12(7):e0180539

Bengio Y, Louradour J, Collobert R, et al. (2009) Curriculum learning. In: Proceedings of the 26th Annual International Conference on Machine Learning, 41–48

Brainard J (2020) Scientists are drowning in COVID-19 papers. Can new tools keep them afloat? — science.org. https://www.science.org/content/article/scientists-are-drowning-covid-19-papers-can-new-tools-keep-them-afloat, [Accessed 25-May-2023]

Cartwright D, Harary F (1956) Structural balance: a generalization of Heider's theory. Psychol Rev 63(5):277

Chen T, Kornblith S, Norouzi M, et al. (2020) A simple framework for contrastive learning of visual representations. In: International Conference on Machine Learning, PMLR, 1597–1607

Crichton G, Guo Y, Pyysalo S et al. (2018) Neural networks for link prediction in realistic biomedical graphs: a multi-dimensional evaluation of graph embedding-based approaches. BMC Bioinform 19(1):1–11

Deepika A (2018) Effect of flaxseed oil in plaque induced gingivitis-a randomized control double-blind study. J Evid Based Med Healthc 5(10):882–5

Fan Jw, Lussier YA (2017) Word-of-mouth innovation: hypothesis generation for supplement repurposing based on consumer reviews. In: AMIA Annual Symposium Proceedings, American Medical Informatics Association, p 689

Gilad-Bachrach R, Navot A, Tishby N (2006) Query by committee made real. NeurIPS, Denver

Gitmez AA, Zárate RA (2022) Proximity, similarity, and friendship formation: Theory and evidence. arXiv preprint arXiv:2210.06611

Gopalakrishnan V, Jha K, Zhang A, et al. (2016) Generating hypothesis: Using global and local features in graph to discover new knowledge from medical literature. In: Proceedings of the 8th International Conference on Bioinformatics and Computational Biology, BICOB, 23–30

Grover A, Leskovec J (2016) node2vec: Scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 855–864

Hacohen G, Weinshall D (2019) On the power of curriculum learning in training deep networks. In: International Conference on Machine Learning, PMLR, 2535–2544

Hamilton W, Ying Z, Leskovec J (2017) Inductive representation learning on large graphs. Adv Neural Inform Proc Syst. https://doi.org/10.48550/arXiv.1706.02216

Hendrycks D, Gimpel K (2016) Bridging nonlinearities and stochastic regularizers with gaussian error linear units. CoRR, abs/160608415 3

Hisano R (2018) Semi-supervised graph embedding approach to dynamic link prediction. In: Complex Networks IX: Proceedings of the 9th Conference on Complex Networks CompleNet 2018 9, Springer, 109–121

Hristovski D, Friedman C, Rindflesch TC, et al. (2006) Exploiting semantic relations for literature-based discovery. In: AMIA Annual Symposium Proceedings, 349

Jha K, Xun G, Wang Y, et al. (2019) Hypothesis generation from text based on co-evolution of biomedical concepts. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, ACM, 843–851

Kazemi SM, Goel R, Eghbali S, et al. (2019) Time2vec: Learning a vector representation of time. arXiv preprint arXiv:1907.05321

King RD, Whelan KE, Jones FM et al. (2004) Functional genomic hypothesis generation and experimentation by a robot scientist. Nature 427(6971):247–252

King RD, Rowland J, Oliver SG et al. (2009) The automation of science. Science 324(5923):85–89

Kirsch A, van Amersfoort J, Gal Y (2019) BatchBALD: efficient and diverse batch acquisition for deep Bayesian active learning. NeurIPS, Denver

Kitano H (2021) Nobel turing challenge: creating the engine for scientific discovery. npj Syst Biol Appl 7(1):29

Klein MT, Hou G, Quann RJ et al. (2002) Biomol: a computer-assisted biological modeling tool for complex chemical mixtures and biological processes at the molecular level. Environ Health Perspect 110(suppl 6):1025–1029

Krenn M, Buffoni L, Coutinho B et al. (2023) Forecasting the future of artificial intelligence with machine learning-based link prediction in an exponentially growing knowledge network. Nat Machine Intell 5(11):1326–1335

Kumari P, Goru R, Chaudhuri S et al. (2020) Batch decorrelation for active metric learning. IJCAI-PRICAI, Jeju Island

Kumar S, Zhang X, Leskovec J (2019) Predicting dynamic embedding trajectory in temporal interaction networks. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 1269–1278

Liu Y, Liu Y, Li P et al. (2022) Antibacterial properties of cyclolinopeptides from flaxseed oil and their application on beef. Food Chem 385:132715

Lönnerdal B, Sandberg AS, Sandström B et al. (1989) Inhibitory effects of phytic acid and other inositol phosphates on zinc and calcium absorption in suckling rats. J Nutr 119(2):211–214

Loshchilov I, Hutter F (2017) Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101

Milani Fard A, Bagheri E, Wang K (2019) Relationship prediction in dynamic heterogeneous information networks. In: Advances in Information Retrieval: 41st European Conference on IR Research, ECIR 2019, Cologne, Germany, April 14–18, 2019, Proceedings, Part I 41, Springer, 19–34
Nguyen GH, Lee JB, Rossi RA et al. (2018) Continuous-time dynamic network embeddings. Companion Proc Web Conf 2018:969–976
Pareja A, Domeniconi G, Chen J, et al. (2020) Evolvegcn: Evolving graph convolutional networks for dynamic graphs. In: Proceedings of the AAAI conference on artificial intelligence, 5363–5370
Pinsler R, Gordon J, Nalisnick E et al. (2019) Bayesian batch active learning as sparse subset approximation. NeurIPS, Denver
Priyadarshini K, Chaudhuri S, Borkar V, et al. (2021) A unified batch selection policy for active metric learning. In: Machine Learning and Knowledge Discovery in Databases. Research Track: European Conference, ECML PKDD 2021, Bilbao, Spain, September 13–17, 2021, Proceedings, Part II 21, Springer, 599–616
Rossi E, Chamberlain B, Frasca F, et al. (2020) Temporal graph networks for deep learning on dynamic graphs. arXiv preprint arXiv:2006.10637
Schullehner K, Dick R, Vitzthum F et al. (2008) Benzoxazinoid biosynthesis in dicot plants. Phytochemistry 69(15):2668–2677
Settles B (2012) Active learning. SLAIML, Shimla
Shi F, Foster JG, Evans JA (2015) Weaving the fabric of science: dynamic network models of science's unfolding structure. Soc Networks 43:73–85
Singer U, Guy I, Radinsky K (2019) Node embedding over temporal graphs. arXiv preprint arXiv:1903.08889
Smalheiser NR, Swanson DR (1998) Using Arrowsmith: a computer-assisted approach to formulating and assessing scientific hypotheses. Comput Methods Prog Biomed 57(3):149–153
Spangler S (2015) Accelerating discovery: mining unstructured information for hypothesis generation. Chapman and Hall/CRC, Boca Raton
Spangler S, Wilkins AD, Bachman BJ, et al. (2014) Automated hypothesis generation based on mining scientific literature. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 1877–1886
Srihari RK, Xu L, Saxena T (2007) Use of ranked cross document evidence trails for hypothesis generation. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 677–686
Srivastava N, Hinton G, Krizhevsky A et al. (2014) Dropout: a simple way to prevent neural networks from overfitting. J Machine Learn Res 15(1):1929–1958
Swanson DR (1986) Fish oil, Raynaud's syndrome, and undiscovered public knowledge. Perspect Biol Med 30(1):7–18
Swanson DR, Smalheiser NR (1997) An interactive system for finding complementary literatures: a stimulus to scientific discovery. Artif Intell 91(2):183–203
Sybrandt J, Shtutman M, Safro I (2017) Moliere: Automatic biomedical hypothesis generation system. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 1633–1642
Sybrandt J, Tyagin I, Shtutman M, et al. (2020) Agatha: automatic graph mining and transformer based hypothesis generation approach. In: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, 2757–2764
Tabachnick BG, Fidell LS (2000) Computer-assisted research design and analysis. Allyn & Bacon Inc, Boston
Trautman A (2022) Nutritive knowledge based discovery: Enhancing precision nutrition hypothesis generation. PhD thesis, The University of North Carolina at Charlotte
Trivedi R, Farajtabar M, Biswal P, et al. (2019) Dyrep: Learning representations over dynamic graphs. In: International Conference on Learning Representations
Van der Maaten L, Hinton G (2008) Visualizing data using t-sne. J Machine Learn Res 9(11):2579–2605
Vaswani A, Shazeer N, Parmar N et al. (2017) Attention is all you need. Adv Neural Inform Proc Syst. https://doi.org/10.48550/arXiv.1706.03762
Wang Y, Wang W, Liang Y et al. (2021) Curgraph: curriculum learning for graph classification. Proc Web Conf 2021:1238–1248
Wang Z, Li Q, Yu D et al. (2022) Temporal graph transformer for dynamic network. In: Part II (ed) Artificial Neural Networks and Machine Learning-ICANN 2022: 31st International Conference on Artificial Neural Networks, Bristol, UK, September 6–9, 2022, Proceedings. Springer, Cham, pp 694–705
Wang L, Chang X, Li S, et al. (2021a) Tcl: Transformer-based dynamic graph modelling via contrastive learning. arXiv preprint arXiv:2105.07944

Weissenborn D, Schroeder M, Tsatsaronis G (2015) Discovering relations between indirectly connected biomedical concepts. J Biomed Semant 6(1):28
Wen Y, Zhang K, Li Z, et al. (2016) A discriminative feature learning approach for deep face recognition. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VII 14, Springer, 499–515
White K (2021) Publications Output: U.S. Trends and International Comparisons | NSF - National Science Foundation — ncses.nsf.gov. https://ncses.nsf.gov/pubs/nsb20214, [Accessed 25-May-2023]
Xun G, Jha K, Gopalakrishnan V, et al. (2017) Generating medical hypotheses based on evolutionary medical concepts. In: 2017 IEEE International Conference on Data Mining (ICDM), IEEE, 535–544
Zhang R, Wang Q, Yang Q et al. (2022) Temporal link prediction via adjusted sigmoid function and 2-simplex structure. Sci Rep 12(1):16585
Zhang Y, Pang J (2015) Distance and friendship: A distance-based model for link prediction in social networks. In: Asia-Pacific Web Conference, Springer, 55–66
Zhang Z, Wang J, Zhao L (2023) Relational curriculum learning for graph neural networks. https://openreview.net/forum?id=1bLT3dGNS0
Zhong Y, Huang C (2023) A dynamic graph representation learning based on temporal graph transformer. Alexandria Eng J 63:359–369
Zhou H, Jiang H, Yao W et al. (2022) Learning temporal difference embeddings for biomedical hypothesis generation. Bioinformatics 38(23):5253–5261
Zhou L, Yang Y, Ren X, et al. (2018) Dynamic network embedding by modeling triadic closure process. In: Proceedings of the AAAI Conference on Artificial Intelligence

Authors and Affiliations

Uchenna Akujuobi[1] · Priyadarshini Kumari[2] · Jihun Choi[3] · Samy Badreddine[1] · Kana Maruyama[3] · Sucheendra K. Palaniappan[4] · Tarek R. Besold[1]

✉ Uchenna Akujuobi
uchenna.akujuobi@sony.com

✉ Priyadarshini Kumari
priyadarshini.kumari@sony.com

Jihun Choi
jihun.a.choi@sony.com

Samy Badreddine
samy.badreddine@sony.com

Kana Maruyama
kana.maruyama@sony.com

Sucheendra K. Palaniappan
sucheendra@sbi.jp

Tarek R. Besold
tarek.besold@sony.com

[1] Sony AI, Barcelona, Spain

[2] Sony AI, Cupertino, USA

[3] Sony AI, Tokyo, Japan

[4] The Systems Biology Institute, Tokyo, Japan