Heterogeneous Graph Neural Network with Personalized and Adaptive Diversity for News Recommendation

GUANGPING ZHANG, Fudan University, Shanghai, China
DONGSHENG LI, Microsoft Research Asia, Shanghai, China
HANSU GU*, Seattle, Washington, USA
TUN LU*, Fudan University, Shanghai, China
NING GU, Fudan University, Shanghai, China

The emergence of online media has facilitated the dissemination of news, but has also introduced the problem of information overload. To address this issue, providing users with accurate and diverse news recommendations has become increasingly important. News possesses rich and heterogeneous content, and the factors that attract users to news reading are varied. Consequently, accurate news recommendation requires modeling of both the heterogeneous content of news and the heterogeneous user-news relationships. Furthermore, users' news consumption is highly dynamic, which is reflected in the differences in topic concentration among different users and in the real-time changes in user interests. To this end, we propose a Heterogeneous Graph Neural Network with Personalized and Adaptive Diversity for News Recommendation (DivHGNN). DivHGNN first represents the heterogeneous content of news and the heterogeneous user-news relationships as an attributed heterogeneous graph. Then, through a heterogeneous node content adapter, it models the heterogeneous node attributes into aligned and fused node representations. With the proposed attributed heterogeneous graph neural network, DivHGNN integrates the heterogeneous relationships to enhance node representation for accurate news recommendations. We also discuss relation pruning, model deployment, and cold-start issues to further improve model efficiency. In terms of diversity, DivHGNN simultaneously models the variance of nodes through variational representation learning for providing personalized diversity. Additionally, a time-continuous exponentially decaying distribution cache is proposed to model the temporal dynamics of user real-time interests for providing adaptive diversity. Extensive experiments on real-world news datasets demonstrate the effectiveness of the proposed method.

CCS Concepts: • **Information systems** → **Recommender systems**.

Additional Key Words and Phrases: news recommendation, graph neural network, heterogeneous information network, recommendation diversity

1 INTRODUCTION

The recent proliferation of social media has significantly changed the way in which people publish and acquire news. However, the convenience and openness of news publishing in various web applications and online social networks often cause a serious information overload problem [28]. For instance, millions of news are published on online news platforms such as MSN News and Google News everyday [1] and it is impossible for a user to glance

*Corresponding author.

Authors' addresses: Guangping Zhang, Fudan University, Shanghai, China, gpzhang20@fudan.edu.cn; Dongsheng Li, Microsoft Research Asia, Shanghai, China, dongshengli@fudan.edu.cn; Hansu Gu, Seattle, Washington, USA, hansug@acm.org; Tun Lu, Fudan University, Shanghai, China, lutun@fudan.edu.cn; Ning Gu, Fudan University, Shanghai, China, ninggu@fudan.edu.cn.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM 1559-1131/2024/3-ART
https://doi.org/10.1145/3649886

Just Accepted

through in a timely manner. Therefore, building effective news recommender systems to provide users with personalized information filtering is very important with both challenges and opportunities [50, 62]. Recently, deep learning-based approaches have been proposed to provide personalized news recommendations [16, 43, 47, 51]. However, these methods present limited performance on recommendation accuracy and diversity.

From the perspective of accuracy, the integrated modeling of heterogeneous news content and heterogeneous user-news relationships is crucial to providing accurate news recommendations. Considering that textual content takes up a large part of the news, early works achieved news recommendations by integrating natural language processing techniques into collaborative filtering [47, 48, 51]. However, the homogenous textual information is not sufficient for news recommendations. News essentially contains multiple types of information, including categorical information, relevant knowledge, and topological information. All types of information are important in news understanding as they focus on different aspects of the news [22, 31, 36, 43]. For instance, "Michael Jordan" as an entity could be connected to additional knowledge such as "six-time NBA champions", while the plain language itself does not necessarily contain the extra information. Besides, some other works attempt to model the high-order relationships based on the user-news bipartite graph to enhance node representation [8, 16, 17]. These methods improve the accuracy of news recommendations to a certain extent, but there are many factors that attract users to read news, which is difficult to fully describe using a single "user clicks on news" relation [49, 56]. For example, a sports fan may click on a piece of sports-related news due to the news's high-quality content, its relevance to their favorite team, or their friends' clicks on the news. Thus modeling the heterogeneous relationships with a decomposed heterogeneous information network (HIN) and identifying the critical relations are necessary for capturing the complicated reasons motivating users to click on news. Moreover, heterogeneous elements in the HIN, such as word tokens and knowledge entities, could provide additional high-order connectivity, which is beneficial in solving the cold-start challenge.

From the perspective of diversity, most existing works define recommendation diversity as the average in-list diversity of all the users at all times, ignoring the personalized and adaptive demands of users for news diversity [30, 31, 33]. Users have varying news reading habits, leading to differing expectations for news diversity. For example, users who primarily read news from subscribed publishers and have specific topic preferences may have a lower requirement for diversity in their news recommendations, while those who enjoy browsing news-feeding streams for leisure may have a higher need for diverse news recommendations. This personalizing of diversity in news recommendations can be referred to as "personalized diversity". Besides, Users often present real-time changing needs for news diversity. When searching for news, they prefer a diverse range of recommendations to increase the chances of finding interesting articles, but when engaged with a particular story, they prefer recommendations for related articles instead. We define this need for news recommendations to adaptively adjust recommendation diversity to automatically zoom in and out based on the users' real-time interests as "adaptive diversity".

In this paper, we propose a **H**eterogeneous **G**raph **N**eural **N**etwork with Personalized and Adaptive **Div**ersity for News Recommendation (**DivHGNN**). It provides a unified architecture for modeling the heterogeneous content of news and the heterogeneous user-news relationships to achieve accurate news recommendations. We first build an attributed heterogeneous graph that organizes the heterogeneous nodes, relations, and attributes. Secondly, to bridge the semantic gap of the heterogeneous node contents, we propose an attribute-level attentive heterogeneous node content adapter, in which we align and fuse the heterogeneous node attributes into unified node representations. For heterogeneous relationship modeling, we further design a relational heterogeneous graph neural network to learn the user and news representations from both the graph structural information and the heterogeneous node contents. To improve the efficiency and performance of the proposed DivHGNN, a relation pruning method is proposed for identifying critical relations. DivHGNN also provides new features of personalized and adaptive diversity. For personalized diversity, it simultaneously models the variance of nodes through variational representation learning. For adaptive diversity, we design an exponentially decaying

Just Accepted

cache, which stores the variational distributions of clicked news with timestamps for each user and applies exponential decaying to characterize the temporal dynamics. DivHGNN is built in a hierarchical manner and adopts a functional and modular architecture, further improving the efficiency of model updates and inference in deployment. Experimental results demonstrate that DivHGNN could substantially improve the accuracy and diversity, even for the cold-start news recommendation scenario, and recommend news with personalized and adaptive diversity. The further discussion explains how the functional and modular architecture benefits DivHGNN in providing more prompt recommendations with lower computational cost, the potential of building a user-controllable news recommender system based on DivHGNN, and limitations of deployment complexity and multi-modal news content support.

The major contributions of this work are summarized as follows:

- We propose an attributed heterogeneous graph representation learning method for news recommendation. Equipped with the heterogeneous node content adapter, the relational heterogeneous graph neural network, and the relation pruning method, DivHGNN can efficiently learn the user and news representations from both the heterogeneous news content and the heterogeneous user-news relationships. Besides, we emphasize the architecture designing principles of functionality and modularity in dealing with the cold-start news and improving computational efficiency.
- We innovatively define personalized diversity and adaptive diversity for news recommendations. We further devise a variational representation learning method that models the variances of users and news for personalized diversity, and design an exponentially decaying cache to model the temporal dynamics of users' real-time interests for adaptive diversity.
- Extensive experiments on real-world datasets demonstrate that DivHGNN can outperform the state-of-the-art news recommendation algorithms. Besides, our analyses show that DivHGNN can support efficient online deployment and presents the potential to build user-controllable news recommender systems.

2 RELATED WORK

2.1 News Recommendation

News recommendation not only involves mining the collaborative nature of user behavior but also requires a fine-grained understanding of the news content to achieve accurate recommendations and address specific challenges such as cold-start news [50]. Therefore, traditional collaborative filtering algorithms present a limited performance in this context [13, 53, 54]. Some methods address these challenges by combining natural language processing techniques with collaborative filtering techniques [25, 47, 48]. A common approach of these methods is to build a textual content encoding model initialized by pretrained language models, and then model user preference by aggregating the click records [51]. However, both the news content and the user-news relationship modeled by these methods are homogeneous, without fully leveraging the heterogeneous nature of news recommendation.

First, since news contains various content information, such as graph topological information in the heterogeneous graph, knowledge entities, and categorical information, encoding news based on a single textual perspective may lead to insufficient and biased content understanding. Indeed, as demonstrated by Arora *et al.* [2], the performance of language models such as Glove [27] and BERT [4] on NLP tasks, are easily influenced by the text complexity, word ambiguity, and prevalent unseen words, etc. Thus, it is important to comprehensively understand the rich news content with heterogeneous attributes. Following this idea, some methods improve news content modeling by introducing external information, such as knowledge graph [22, 36, 43]. However, these methods ignore the high-order relationships between user and news in the heterogeneous graph.

Second, user behaviors are influenced by various factors, and the complicated relationships between the users and news cannot be directly captured from the sparse click records [49], and as a result, it's necessary to introduce heterogeneous high-order information to support the user-news relationship modeling. Following this idea, some methods build relationship networks based on the user-news interaction records and model the representation of users and news from the collaborative topological information [8, 16, 17, 23, 56]. These methods exploited the heterogeneous high-order relationships, but failed at integrating heterogeneous node attributes, resulting in limited model performance.

2.2 Attributed Heterogeneous Graph Neural Network

Graph neural network [10, 40], which aims at learning low-dimensional node representations by preserving the structural properties, has shown superior performance on various machine learning tasks, such as node classification, node clustering, and link prediction.

As a considerable number of real-world scenarios are inherently heterogeneous, involving multiple node and relation types, heterogeneous graph neural networks (HGNN) are proposed by introducing advanced graph sampling algorithm [45, 59] or relation-aware architecture [38, 63]. Although these techniques enhanced the capability of graph neural networks to handle heterogeneity, they are more inclined to model network structure rather than node attributes, and thus may yield sub-optimal representations when attributes contain important discriminative features.

On the other side, attributed network embedding (ANE) focuses on learning node representations from both the topological structure and the node attributes. These models construct neighborhood attribute sequences through random walk or message passing, and are trained based on tasks such as attribute reconstruction and connection prediction [7, 26, 58]. These methods achieve the fusion of high-order topological information and node attribute information, but ignore the heterogeneity of nodes, relations, and attributes.

Combining the advantages of the above methods, attributed heterogeneous graph neural network (AHGNN) learns heterogeneity-aware and attribute-aware node representations. Different from the homogeneous GNNs which could directly fuse the attributes to update node representations, AHGNNs need to overcome the attribute heterogeneity and design effective fusion mechanisms to utilize the node content information [34, 61]. Key stages in the processing flow include neighbor sampling, homogeneous information aggregation, heterogeneous information alignment and fusion, etc. Table 1 compares the supported features of the above four types of GNN. The fully supported features of AHGNN match the requirements of utilizing heterogeneous news content and heterogeneous user-news relationships for news recommendation. However, AHGNNs are faced with the challenge of providing computational-efficient personalized service for news recommendation scenarios, in which millions of cold-start news are published every day and user interests may vary in real-time.

In this work, we extend the AHGNN by emphasizing the architecture design principles of functionality and modularity, to deal with the cold-start news challenge and improve computational efficiency. Heterogeneous graph neural networks present the ability of inductive learning, i.e., learning representations for the out-of-sample nodes and combating adversarial attacks [15]. Considering that cold-start news could build initial connections with the heterogeneous information networks through its content, it could be regarded as a special case that is attacked by masking its connections with the users. Based on the above two conditions, we propose to implement the GNN as

Just Accepted

Table 1. Feature comparison of four types of GNN methods.

	High-order Relationship	Attributed Nodes	Heterogenous Nodes	Heterogenous Relationship
Traditional GNN	✓	✗	✗	✗
HGNN	✓	✗	✓	✓
ANE	✓	✓	✗	✗
AHGNN	✓	✓	✓	✓

a mapping function rather than static embeddings for supporting the incremental node representation, and rely on the model robustness when missing partial connections to provide cold-start news recommendations. Besides, our approach utilizes a modularized architecture, which could achieve higher update efficiency through model reuse and hierarchical deployment, and enables further modular extensions that provide additional properties such as real-time recommendation, diversity improvement, etc.

2.3 Parameter Pruning

The growing size of neural networks brings with it higher model accuracy and, at the same time, an increasing computational expense. As a result, many approaches such as down-sizing models [6], operator factorization [64], value compression [20], parameter sharing [29], and sparsification [5] have been proposed to minimize the computational cost and improve the generalization and robustness of the models while ensuring their accuracy. Among all the above approaches, sparsification (also referenced as parameter pruning) is one of the most powerful methods and has attracted a great deal of recent research attention [14]. According to the pruning object, parameter pruning can be divided into model pruning [46], which lightens the model by pruning its weights, neurons, blocks, etc., and ephemeral pruning [35], which constructs a sub-network for each example by means of dropout, conditional computation, etc. In this paper, we propose a relation pruning method for heterogeneous graph neural networks, which follows the methodology of model pruning on neuron-like structures, with the 1st-order Taylor expansion of the training loss as the pruning criteria.

2.4 Diversity Modeling

Recommendation motivates users' long-term participation, but there is always a trade-off between accuracy and diversity [18]. Early works [65] utilized a two-stage strategy, in which the second stage was used to promote in-list diversity by re-ranking and re-evaluating. More recent approaches attempt to deal with the diversity challenge from the perspective of model optimization. Some works formulize the in-list diversity as a regularization term in the optimizing objective, to achieve simultaneous optimization of both the accuracy and diversity [9, 32]. Other works attempt to improve the recommendation diversity via targeted model structure designs, such as hierarchical interest modeling [31]. Some data-level attempts have also successfully improved recommendation diversity, such as multi-field data [55], popularity data [24, 30], similarity-aware neighbor sampling [55, 57]. Although these approaches enhanced the diversity of news recommendations, treating diversity simply as an overall metric

Just Accepted

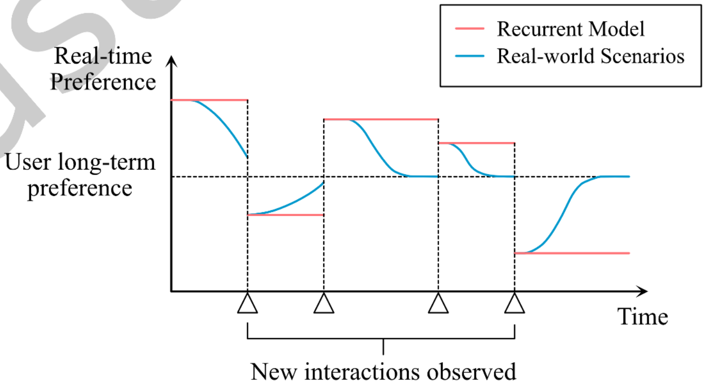


Fig. 1. The difference in real-time user preference between recurrent models and real-world scenarios.

ignored the dynamic need for news diversity across users and scenarios, resulting in an unsatisfactory user experience. Different from existing approaches, we propose to model the dynamic user needs with personalized and adaptive diversity, and further deal with them using variance modeling and exponentially decaying user interest cache.

2.5 Temporal Dynamics

Another related research area is temporal dynamics, which refers to the phenomenon that user interest changes over time. There are several news recommendation methods considering the temporal dynamics by separately modeling the long-term interests which represent the stable preference and the short-term interests which reflect the real-time demand. Okura *et al.* [25] discussed different user preference models, including word-based models, decaying models, and recurrent models, and reported that the recurrent models outperformed the others. This conclusion is widely adopted by the follow-up works [1, 16, 19, 36]. These methods improve recommendation accuracy and overall diversity by modeling user interest shifting, but struggle in achieving automatic zooming in and out, and fail at meeting the real-time nature of adaptive diversity. Utilizing recurrent models, these methods learn the long and short-term user interests from historical clicked sequences, and the captured temporal dynamics, which are reflected in the tail of the sequence, are refreshed when and only when new clicks happen. As illustrated in Figure 1, recurrent model-based approaches fail at handling the real-time decay of short-term interest intensity, and the modeled short-term interests between two observed click records are discrete in the time dimension. For example, for a user who visits the platform after a considerable interval, her/his short-term interest from the perspective of the recurrent model stays unchanged, as there is no new interaction observed in the middle. However, the user's short-term interest has very likely shifted already during the long visiting interval. It is thus necessary to model the user interest temporal dynamics in a real-time manner. In this regard, some sequential product recommendation approaches [42] introduced Hawkes Process [12] with exponentially decaying temporal kernel functions to model the real-time user demand and deal with the repeat consumption challenge [41].

3 PRELIMINARY

This section gives formal definitions of key terminologies. Table 2 summarizes frequently used notations in this paper for quick reference.

Definition 1 (Attributed Heterogeneous Graph). *An attributed heterogeneous graph with multiple attributes is defined as a graph $G = (V, E, T_V, T_E, A_V, A_E)$ with multiple types of nodes V and links E. T_V and T_E represent the set of node and link types, with the property that $|T_V| + |T_E| > 2$. A_V denotes the heterogeneous node attributes, which is defined as a collection of triples $A_V = \{(t, v, a)\}$ where t denotes the attribute type, $v \subseteq V$ is the affiliated node, and a refers to the attribute value. Similarly, the heterogeneous edge attributes A_E is defined as $A_E = \{(t, e, a)\}$, where e is the affiliated link.*

Definition 2 (Attributed Heterogeneous Graph Representation Learning). *Given an attributed heterogeneous graph $G = (V, E, T_V, T_E, A_V, A_E)$ as the input, the representation learning task on G is to learn the low dimensional node representation $H_V \in \mathbb{R}^{|V| \times d}$ and link representation $H_E \in \mathbb{R}^{|E| \times d}$ as output that contain both structural information of G and heterogeneous content information of A_V and A_E.*

The output representation H_V and H_E could be applied to a variety of downstream tasks, such as node classification and link prediction. In this work, we focus on learning the representations of the users and news for recommendation from A_V, which is a link prediction task.

Just Accepted

Table 2. Notations used in this paper. Indices are used as superscripts or subscripts to indicate the specific object of the operation.

Notation	Form	Description
v	Index	Nodes
t	Index	Node attribute type
a	Vector	The pre-trained model encoded node attributes
C	Matrix	The node content matrix, each row represents one attribute
c	Vector	The node content representation
M	Dictionary	The node attributes position map
r	Index	Relation
l	Index	GNN layer
k	Index	Attention head
h	Vector	Intermediate node representation in GNN hidden layers
W	Matrix	GNN transformation matrix
$\mathcal{N}$	Set	Node neighbors
μ	Vector	The mean vector of node variational representation
σ	Vector	The variance vector of node variational representation
$\mathcal{L}$	Scalar	The training loss
ϑ	Set	All trainable model parameters
u	Index	User
n	Index	News
B	Set	Pruning block
ψ	Scalar	Pruning criteria
m	Boolean	Pruning mask

Personalized diversity characterizes the unique interest ranges of users when browsing news articles. Therefore, the diversity of recommended news lists should align with the diverse browsing behaviors of users. Formally, the following conditions should hold to ensure personalized diversity.

Definition 3 (Personalized Diversity). *For any two users u_1 and u_2 in the user set U, given an inner-list news diversity metric $Div(\cdot)$* [1]*, when the historical clicking records L'_{u_1} and L'_{u_2} satisfy $Div(L'_{u_1}) > Div(L'_{u_2})$, the recommendation lists L_{u_1} and L_{u_2} also satisfy $Div(L_{u_1}) > Div(L_{u_2})$.*

Adaptive diversity characterizes the changing interests of users when browsing news articles. Therefore, the diversity of recommended news lists should align with the temporal user interest changes that decay from newly clicked news to long-term interests. Formally, the following conditions should hold to ensure adaptive diversity.

Definition 4 (Adaptive Diversity). *Given that user u clicked on news n on time t_0, for recommendation lists L_{t_1} and L_{t_2} generated on time t_1 and t_2 with $t_0 < t_1 < t_2 < t'_0$, where t'_0 refers to the time of next clicking record, we have $Div(L_{t_1}, L_u) > Div(L_{t_2}, L_u)$ and $Div(L_{t_1}, L_n) < Div(L_{t_2}, L_n)$, in which $Div(\cdot)$ is an inter-list news diversity metric, L_u is the recommendation list generated based on user representation H_u, and L_n is the recommendation list generated based on news representation H_n.*

Just Accepted

[1] Inter-list news diversity, such as inter-list average distance, measures the dissimilarity between two news recommendation lists. Inner-list news diversity is a special case when the two lists are the same one, and measures the dissimilarity within the list.

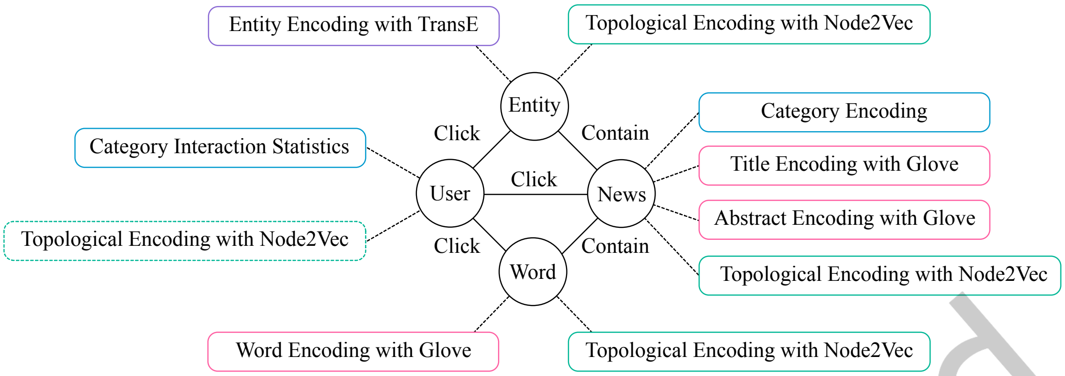

Entity Encoding with TransE
Topological Encoding with Node2Vec
Category Interaction Statistics
Topological Encoding with Node2Vec
Entity
Click
Contain
User
Click
News
Click
Contain
Word
Category Encoding
Title Encoding with Glove
Abstract Encoding with Glove
Topological Encoding with Node2Vec
Word Encoding with Glove
Topological Encoding with Node2Vec

Fig. 2. The data structure of the attributed heterogeneous graph in DivHGNN.

Just Accepted

4 METHOD

In this section, we introduce the **H**eterogeneous **G**raph **N**eural **N**etwork with Personalized and Adaptive **Div**ersity for News Recommendation (**DivHGNN**), which effectively learns the user and news representations from both heterogeneous attributes and informative high-order relationships, and provide personalized and adaptive news diversity via variational representation learning and exponentially decaying cache.

4.1 Attributed Heterogeneous Graph

Users and news are the two fundamental types of nodes in news recommendation, and many approaches [25, 48, 51] model a user-news bipartite graph with homogeneous attributes and learnable embeddings. However, homogeneous attributes are insufficient for the user and news representation learning due to their inherent bias [2]. Besides, user-news clicks happen for various reasons that may not be explained by sparse historical click records directly. We therefore construct an attributed heterogeneous graph to organize multiple information sources in news recommendation scenarios.

Following Definition 1, the attributed heterogeneous graph illustrated in Figure 2 includes node types T_V of user, news, word, and entity. The heterogeneous relationships T_E include user-click-news, user-click-word, user-click-entity, news-contain-word, and news-contain-entity. Besides, we add self-loops to each node type for self-message-passing. The heterogeneous node attributes A_V include textual, knowledge, categorical, and topological attributes. Specifically, the textual attributes, which are colored with pink in Figure 2, are encoded by the pretrained Glove [27]; the knowledge attributes, which are colored with purple in Figure 2, are encoded by the TransE [3] pretrained on knowledge tuples extracted from WikiData; the topological attributes, which are colored with green in Figure 2, are encoded by the Node2Vec [10]; and the categorical attributes, which are and colored with blue in Figure 2, are encoded by statistic-weighted averaging Glove encodings of the category titles.

4.2 Heterogeneous Node Content Adapter

Since the attributed heterogeneous graph contains one or more attributes for each node type, there will be semantic gaps among all the attributes due to different representation learning techniques. Therefore directly aggregating heterogeneous neighborhood information in GNN may be ineffective and inferior.

To tackle this challenge, we extend the Transformer [39] to accommodate the heterogeneous attributes and further propose a heterogeneous node content adapter for attribute alignment and node content fusion, which contains three attribute type-specific aligners corresponding to the textual&categorical [2], knowledge and topological attributes, respectively; and a shared attribute fusion module. Since different nodes contain different attribute combinations, we apply position mapping to ensure the consistency of the input of the shared attribute fusion module. Figure 3 (a) shows the network architecture of the heterogeneous node content adapter.

More specifically, consider node $v \in V$ with content $C^v = \{a_t^v\}$, where $a_t^v \in \mathbb{R}^{d_t}$ denotes the encoded heterogeneous attribute of node v with attribute-type t. The heterogeneous attribute a_t^v is aligned through the attribute type-corresponding aligner implemented by a Multi-Layer Perceptron (MLP) formulated as follows:

$$\tilde{a}_t^v = \text{MLP}_t\left(a_t^v\right). \tag{1}$$

Then, the aligned node content matrix $\tilde{C}^v = [\tilde{a}_t^v \in \mathbb{R}^{d'}]$ is aggregated for node content fusion via the shared fusion module. Firstly, we model the cross-attribute dependencies based on the self-attention mechanism as follows:

$$\hat{C}^v = \text{softmax}\left(\frac{\tilde{C}^v[\tilde{C}^v]^T}{\sqrt{d'}}\right)\tilde{C}^v. \tag{2}$$

The shared fusion module is implemented by an MLP, and each attribute is assigned a fixed position in its input. To construct the input, we map each crossed attribute $\hat{a}^v \in \hat{C}^v$ to the corresponding position, and for attributes which node v is not affiliated with, we pad the corresponding position in the input with zeros as follows:

$$c^v = \text{MLP}_{fusion}\left(\text{mapping\&padding}(\hat{C}^v, M, \mathbf{0})\right), \tag{3}$$

where $\hat{C}^v$ is the crossed node content and M refers to the node attributes position map. c^v is the fused node representation, which is also regarded as the primitive node representation for the following graph neural network.

By applying alignment and fusion, the heterogeneous node content adapter addresses the semantic difference among node attributes. This adapter reduces the complexity of the user and news representation learning, and enables the rest of the GNN model to focus on heterogeneous high-order relationships.

4.3 Attributed Heterogeneous Graph Representation Learning

After performing node semantic alignment, we need to further aggregate the neighborhood information to enhance node representation for accurate recommendation. Different node relations often present unique influence patterns that are difficult to model through a homogeneous model. Therefore, a major challenge in neighborhood information aggregation is how to build a unified model for heterogeneous relations. In addition, preserving the semantic information carried by nodes also further increases the difficulty. Following Definition 2, we utilize high-order relationships on top of the attributed heterogeneous graph for news recommendation, as shown in Figure 3 (b). To obtain the enhanced node representations, we aggregate heterogeneous attributes and relationships from their neighborhood N_V based on the primitive node representation c^V. Before aggregation, we perform neighbor sampling to obtain a fixed amount of neighbors as the size of the node neighborhood varies in a wide range. Besides, considering that the attributed heterogeneous graph G has multiple relationships and each relationship presents a unique impact, we leverage a relational graph neural network with the message-passing framework to learn the node representations.

[2] Textual and categorical attributes share the same aligner as they utilize the same encoding technique (Glove).

Just Accepted

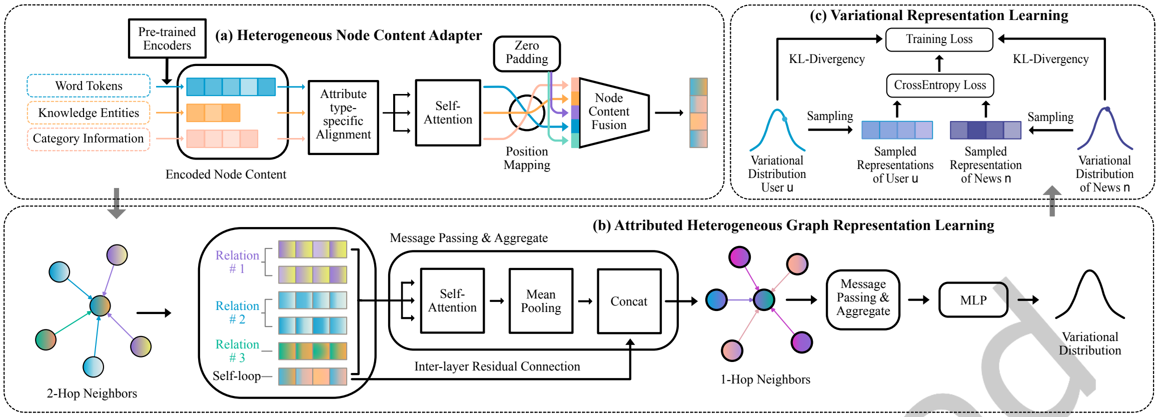

(a) Heterogeneous Node Content Adapter
Pre-trained Encoders
Word Tokens
Knowledge Entities
Category Information
Encoded Node Content
Attribute type-specific Alignment
Self-Attention
Zero Padding
Position Mapping
Node Content Fusion
(c) Variational Representation Learning
Training Loss
KL-Divergency
KL-Divergency
CrossEntropy Loss
Sampling
Sampling
Variational Distribution User u
Sampled Representations of User u
Sampled Representation of News n
Variational Distribution of News n
(b) Attributed Heterogeneous Graph Representation Learning
2-Hop Neighbors
Relation # 1
Relation # 2
Relation # 3
Self-loop
Message Passing & Aggregate
Self-Attention
Mean Pooling
Concat
Inter-layer Residual Connection
1-Hop Neighbors
Message Passing & Aggregate
MLP
Variational Distribution

Fig. 3. The model architecture of DivHGNN. (a) The proposed heterogeneous node content adapter, which aligns and fuses the heterogeneous node content to form the primitive node representation. (b) Attributed heterogeneous graph representation learning, which aggregates the neighborhood information of nodes to enhance node representation. (c) Variational representation learning, which refines the node representation and meanwhile models the variance of the representation.

The relational graph neural network consists of multiple graph convolution layers. In the l-th layer, messages are aggregated with a relation-specific multi-head graph attention layer, which is formulated as follows:

$$e_{vi}^{r\,(l)(k)} = \phi^{r\,(l)(k)}\left(\mathbf{W}^{r\,(l)(k)}h_v^{(l)}, \mathbf{W}^{r\,(l)(k)}h_i^{(l)}\right), \tag{4}$$

$$\alpha_{vi}^{r\,(l)(k)} = \text{softmax}_j(e_{vi}^{r\,(l)(k)}) = \frac{\exp\left(e_{vi}^{r\,(l)(k)}\right)}{\sum_{j\in\mathcal{N}_v^r}\exp\left(e_{vj}^{r\,(l)(k)}\right)}, \tag{5}$$

$$H_v^{r\,(l+1)} = \|_{k=1}^{K} sigmoid\left(\sum_{i\in\mathcal{N}_v^r}\alpha_{vi}^{r\,(l)(k)}\mathbf{W}^{r\,(l)(k)}h_i^{(l)}\right), \tag{6}$$

where the upper corner markers r, l, k refer to the relation, layer, attention head, respectively. $h_v^{(l)}$ denotes the representation of node v in the l-th layer and h_v^0 denotes the primitive node representation c^v. $\mathbf{W} \in \mathbb{R}^{d^{(l+1)}\times d^{(l)}}$ is the projection matrix, and $\phi : \mathbb{R}^{d^{(l+1)}} \times \mathbb{R}^{d^{(l+1)}} \rightarrow \mathbb{R}$ is the coefficient function, which is implemented with a one-layer feed-forward neural network. $\mathcal{N}_v^r$ denotes the nodes in relation r with the source node v. The operator $\|$ represents concatenation. Then, the messages from multiple relations are crossed via self-attention, and considering that each graph convolution layer models different level of high-order connectivities, we utilize inter-layer residual connections, which is formulated as follows:

$$H_v{}^{(l+1)} = \|_{r=1}^{R} H_v^{r\,(l+1)}, \tag{7}$$

$$\hat{H}_v^{(l+1)} = \text{softmax}\left(\frac{H_v{}^{(l+1)}[H_v{}^{(l+1)}]^T}{\sqrt{d^{(l+1)}}}\right)H_v{}^{(l+1)}, \tag{8}$$

$$h_v{}^{(l+1)} = \|\left(\text{mean}(\hat{H}_v^{(l+1)}), h_v{}^{(l)}\right). \tag{9}$$

Just Accepted

Neighborhood information is aggregated iteratively, and the output of the last layer, denote as ${h_v}^{(L)}$, models the heterogenous relationships and attributes with multiple levels of high-order connectivities. We feed ${h_v}^{(L)}$ into an $MLP : \mathbb{R}^{d^{(L)}} \rightarrow \mathbb{R}^{d''}$ for further feature extraction to generate the ultimate node representation h_v as follows:

$$h_v = \text{MLP}_{dense}\left({h_v}^{(L)}\right). \tag{10}$$

4.4 Variational Representation Learning

Traditional news recommendation algorithms, relying on point estimation, fail to accommodate the diverse interests of users and the wide reach of news audiences. This oversight leads to a suboptimal experience due to a lack of personalized content diversity. DivHGNN models user and news representations as distributions, embracing a generative perspective to enhance personalization and diversity in recommendations, directly addressing these limitations. We assume that the process of aggregating neighborhood information to represent node v can be formulated as the conditional probability given its neighbors, denoted as $p(h_v|N_v)$. In this way, the final node representations can be sampled from $p(h_v|N_v)$. However, the ground-truth distribution of node representation $p(h_v|N_v)$ can not be approximated directly. Inspired by [21], we leverage the variational distribution $q(h_v|N_v)$ with a standard factorized Gaussian prior $p(h_v) = \mathcal{N}(\mathbf{0}, \mathbf{1})$ to approximate $p(h_v|N_v)$. Correspondingly, the variational posterior for node v can be denoted as $q(h_v|N_v) = \mathcal{N}(\mu_v, \sigma_v)$, where the mean μ_v and the variance σ_v can be obtained by modifying Equation 10 as follows:

$$[\mu_v, log(\sigma_v)] = \text{MLP}_{dense}\left({h_v}^{(L)}\right). \tag{11}$$

For user/news representation, we select S samples from the variational posterior via reparametrizing and concatenate the samples as its representation, denoted as $\hat{h}_v$.

Finally, regarding the news recommendation on the attributed heterogeneous graph as a user-news link prediction task, the learning objective is to minimize the cross-entropy loss between the predicted score and the true label, subject to the constraints of the prior distribution, formulated as follows:

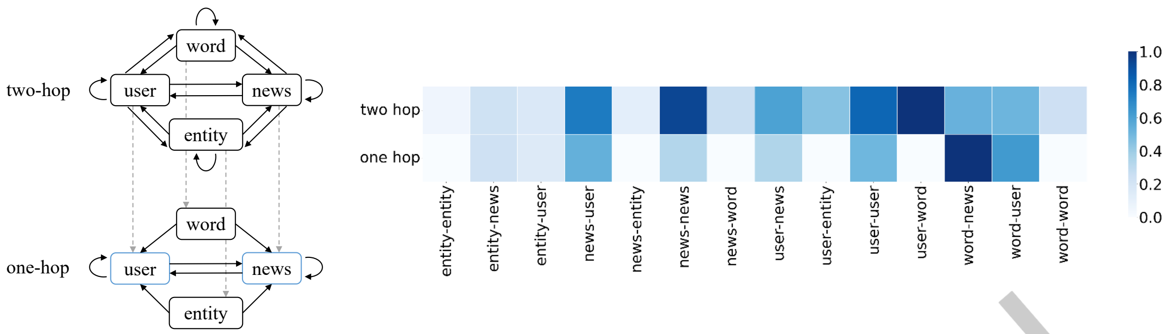
$$\begin{aligned} \underset{\vartheta}{\text{argmin}}\, \mathcal{L} = &\sum_{(u,n,y)\in I} \text{CrossEntropy}(y, sigmoid(\hat{h}_u \cdot \hat{h}_n)) \\ &+\kappa \sum_{v\in V_U \cup V_N} \mathbb{KL}(q_\vartheta(h_v|N_v)||p(h_v)), \end{aligned} \tag{12}$$

where $(u, n, y) \in I$ is a tuple of interaction records. Specifically, the label y is 1 when the user u clicked the news n, and 0 when news n is presented to u but not clicked. V_U and V_N denote all user and news nodes and N_v refers to the heterogeneous neighbor of node v. ϑ denotes the parameters of the proposed graph neural network, and $p(h_v)$ denotes the prior distribution. The first term can be interpreted as the prediction error, where the clicking possibility between the user and the news is calculated by sigmoid over the dot product of the sampled representations. The second term denotes the Kullback–Leibler divergence between the variational posterior and its prior, which serves as regularizations. Figure 3 (c) illustrates the above process.

4.5 Relation Pruning

The information flow of the proposed graph representation learning method can be summarized as the relational meta graph shown in Figure 4 (a) (in the case of a two-layer graph neural network). Neighborhood information aggregates hierarchically towards the seed nodes along the relations depicted by the black solid arrows. Inside each black solid arrow, the information is processed and aggregated by the corresponding graph convolution layer sub-module with parameters $\phi^{r(l)}$ and $\mathbf{W}^{r(l)}$.

Just Accepted



(a) Two-hop relational meta graph

(b) The importance of each relation in the two-hop relational meta graph measured by gradient

Fig. 4. (a) The two-hop relational meta graph used for heterogeneous graph representation learning. Blue boxes represent seed nodes. The black solid arrows depict relations through which messages are processed, passed, and aggregated. Each black solid arrow is associated with a corresponding neural module. The grey dashed arrows denote the assignment operations connecting different layers. (b) The average value of the absolute gradient of each relation-associated neural module, which measures its importance to node representation learning. At each layer, the values are normalized to the 0 to 1 interval by dividing by the maximum value. The darker the color, the higher the importance.

Algorithm 1 Relation Pruning Algorithm for Heterogeneous Graph Neural Network.

```
Input: Training set $\mathcal{D}$, untrained model $\mathcal{M}$, number of relations $R$, number of layers $L$
Initialize: $\psi^{r(l)} = 0$ and $m^{r(l)} = 1$, for $r$ in 1...$R$, for $l$ in 1...$L$
while not converged do
    $\Delta\psi^{r(l)} = 0$, for $r$ in 1...$R$, for $l$ in 1...$L$        ▷Cache for inner epoch gradient accumulation
    for each batch in $\mathcal{D}$ do
        Calculate $\mathcal{L}$ and update $\mathcal{M}$ according to Equation 12        ▷ Calculate training loss
        for $l$ in 1...$L$ do
            for $r$ in 1...$R$ do
                $\Delta\psi^{r(l)} \leftarrow \Delta\psi^{r(l)} + \left|\frac{\partial \mathcal{L}}{\partial B^{r(l)}}\right|$        ▷Accumulate absolute gradient in the cache
            end for
        end for
    end for
    $\psi^{r(l)} \leftarrow \gamma \cdot \psi^{r(l)} + \Delta\psi^{r(l)}$        ▷Update criteria via inter epoch gradient accumulation
    for $l$ in 1...$L$ do
        for $r$ in 1...$R$ do
            $m^{r(l)} = 1$ if $\psi^{r(l)} \geq \varrho \cdot \mathrm{mean}(\mathrm{S}^{(l)})$ , else 0        ▷Calculate the pruning mask
        end for
    end for
end while
Output: Trained model $\mathcal{M}$, pruning masks $\{m\}$
```

However, not all relations are equally important in modeling users and news, which can be illustrated by a motivating example. Figure 4 (b) presents the average value of the absolute gradient of each relation-associated

Just Accepted

graph convolution layer sub-module. In general, if the module parameters do not change much from their initial random values during the learning process, then they are likely to be less important. As shown in Figure 4 (b), the changes in module parameters corresponding to different relations present a large variation, which confirms the difference in importance. We can improve the computational efficiency and accuracy of the model by pruning some unimportant relations. Similarly, some heterogeneous graph neural networks aggregate neighborhood information through pre-defined meta-paths. These approaches are equivalent to pruning by introducing expert knowledge. The superior performance of these approaches confirms the positive impacts of identifying informative relations in providing accurate recommendations.

Inspired by structured pruning, we propose a relation pruning method for heterogeneous graph neural networks to optimize the relational meta-graph in a data- and learning-driven manner. We define each relation-associated graph convolution layer sub-module with parameters $\phi^{r(l)}$ and $\mathbf{W}^{r(l)}$ as a pruning block $B^{r(l)}$, where r refers to the relation and l refers to the layer. We leverage the 1st-order Taylor expansion of the training loss to identify unimportant relations. More specifically, after the training process of each epoch, we accumulate the absolute value of the gradient of the pruning blocks as the criteria, which could be formulated as:

$$\psi_{i+1}^{r(l)} \leftarrow \gamma \cdot \psi_i^{r(l)} + \left| \frac{\partial \mathcal{L}}{\partial B^{r(l)}} \right|, \tag{13}$$

where $\psi_i^{r(l)}$ denotes the criteria of pruning block $B^{r(l)}$ at the i-th epoch, γ is a decaying factor with a value from 0 to 1, and $\mathcal{L}$ denotes the training loss defined in Equation 12. Given a pruning threshold ratio ϱ, we prune blocks that of significantly lower criteria than other blocks in the same set, which could be formulated as:

$$m^{r(l)} = \begin{cases} 1 & \text{if } \psi^{r(l)} \geq \varrho \cdot \text{mean}(\mathrm{S}^{(l)}) \, , \\ 0 & \text{otherwise .} \end{cases} \tag{14}$$

$\mathrm{S}^{(l)}$ is the pruning set consisting of all the criteria of pruning blocks in the l-th layer. $m^{r(l)}$ is the pruning mask, whose value is 1 when the corresponding module is preserved and 0 when pruned. Correspondingly, $\phi^{r(l)}$ and $\mathbf{W}^{r(l)}$ in Equation 6 are replaced by $m^{r(l)} \cdot \phi^{r(l)}$ and $m^{r(l)} \cdot \mathbf{W}^{r(l)}$ respectively. By default, we apply relation pruning for all the graph convolution layers during the full training process.

The proposed relation pruning algorithm is an extension of the standard model training workflow, as illustrated in Algorithm 1. It dynamically adjusts the values of pruning masks during the model training, masking out parts with limited contribution to accelerate the training process. Prior to training, we initialize the pruning criteria $\psi^{r(l)}$ and pruning mask $m^{r(l)}$. During each training epoch before convergence, the absolute gradient of each pruning block is accumulated after each model iteration, as depicted from line 5 to 12. Subsequently, in line 13, the decayed pruning criteria are updated by adding the accumulated changes within the epoch. Finally, at the end of each training epoch, as delineated from line 14 to 18, the mask for each pruning block is computed based on whether the pruning criteria exceed a fixed ratio of the mean value of their corresponding layers. Blocks with a mask value of 0 are subsequently masked out in further training.

4.6 Exponentially Decaying Distribution Cache

In real-world news recommendation scenarios, the user interests are not constant but could be easily attracted by various unexpected events. Therefore, in addition to accurately modeling users' long-term reading interests, we also need to dynamically model users' real-time interests and provide news recommendations with adaptive diversity. However, existing approaches are discrete in the time dimension and difficult to provide real-time service. Thus, we propose the exponentially decaying distribution cache (EDC), in which we save the variational distributions of clicked news together with the interaction timestamp for each user, and apply time-continuous exponentially distribution decaying. Figure 5 shows the architecture of the EDC.

Just Accepted

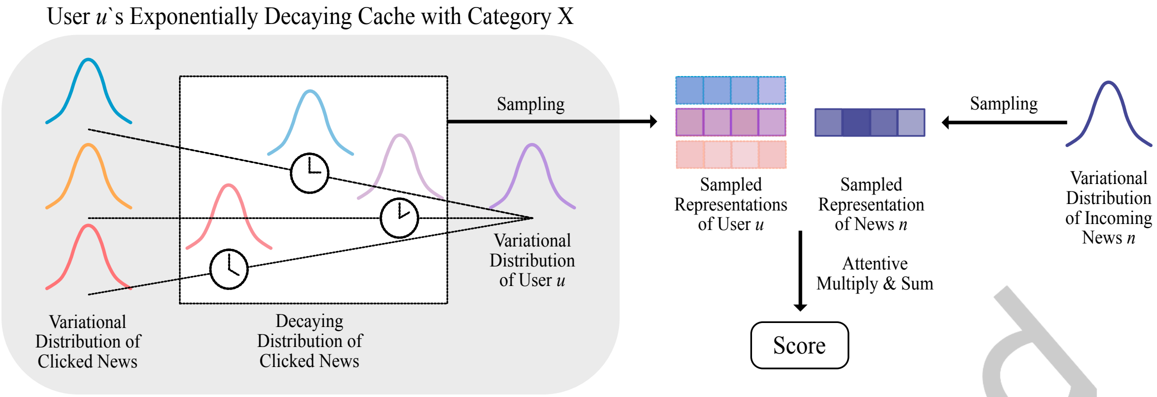

User u`s Exponentially Decaying Cache with Category X
Sampling
Sampling
Sampled Representations of User u
Sampled Representation of News n
Variational Distribution of Incoming News n
Variational Distribution of User u
Attentive Multiply & Sum
Score
Variational Distribution of Clicked News
Decaying Distribution of Clicked News

Fig. 5. The proposed exponentially decaying distribution cache. The incoming news is of category X (satisfying that $\mathbf{R}(n) = X$). If the incoming news n is recommended and clicked by user u, it would replace the cached news which of the earliest click time (the orange one in this case).

The EDC is set along with the categories, as there is a continuity in the news reading behaviors under the same topic. Each EDC possesses a tiny piece of memory which can cache up to M variational distributions along with the timestamp. When processing one incoming news n with timestamp t for user u, we firstly compute the decaying factors of the news in corresponding cache, formulated as:

$$\lambda^u_{\mathbf{R}(n)} = \|_{i=1}^{M} e^{-\beta f(t-\tau^u_{\mathbf{R}(n),i})}, \tag{15}$$

where $\mathbf{R}$ is a cache router, which maps the incoming news n to the corresponding cache. $\tau^u_{\mathbf{R}(n),i}$ and $\lambda^u_{\mathbf{R}(n)}$ denotes the timestamp of the i-th news and the concatenated decaying factors of all the news in the corresponding cache. β is a positive hyper-parameter controlling the decaying speed, and f is a non-negative monotonically increasing function that defines the decaying pattern. In this work, f is implemented as a power function denoted as $f = x^{\rho}$, where ρ is a positive hyper-parameter [37]. With a higher ρ, the cached distribution would be retained for a period and then decay steeply. When ρ is small, the cached distribution would start decaying immediately but with a slower pace.

Controlled by the decaying weights, the variational distribution of cached news would decay towards the variational distribution of the user u by weighted average, which is formulated as:

$$\begin{aligned}\hat{\mu}^u_{\mathbf{R}(n)} &= \lambda^u_{\mathbf{R}(n)} \cdot \mu^u_{\mathbf{R}(n)} + (1 - \lambda^u_{\mathbf{R}(n)}) \cdot \mu_u,\\ \hat{\sigma}^u_{\mathbf{R}(n)} &= \lambda^u_{\mathbf{R}(n)} \cdot \sigma^u_{\mathbf{R}(n)} + (1 - \lambda^u_{\mathbf{R}(n)}) \cdot \sigma_u,\end{aligned} \tag{16}$$

where $\mu^u_{\mathbf{R}(n)}, \sigma^u_{\mathbf{R}(n)}$ and $\hat{\mu}^u_{\mathbf{R}(n)}, \hat{\sigma}^u_{\mathbf{R}(n)}$ denote the concatenated mean and variance of the cached news before and after decaying. We select S samples from the decayed variational distributions and concatenate the samples as the representation of the cached news, denoted as $\hat{h}^u_{\mathbf{R}(n),i}$. We further utilize the attention mechanism to aggregate the representations by their importance in the cache for the ultimate scoring as follows:

$$score_{u,n} = \hat{h}_n \cdot \text{softmax}\left(\frac{\hat{h}_n [\hat{H}^u_{\mathbf{R}(n)}]^T}{\sqrt{S \cdot d''}}\right) \hat{H}^u_{\mathbf{R}(n)}, \tag{17}$$

Just Accepted

where $\hat{H}^{u}_{\mathbf{R}(n)} = \|_{i=1}^{M} \hat{h}^{u}_{\mathbf{R}(n),i}$ is the concatenated news representations of the cache, $\hat{h}_n$ denotes the sampled representation for the incoming news n, S refers to the times of sampling, and d'' is the latent space dimension.

When a new positive interaction is observed, the corresponding cache update its saved news according to FIFO (first-in-first-out), which is formulated as follows:

$$\left(\mu^{u}_{\mathbf{R}(n),i}, \sigma^{u}_{\mathbf{R}(n),i}, \tau^{u}_{\mathbf{R}(n),i}\right)_{\underset{\tau}{\operatorname{argmin}}(i)} \leftarrow (\mu_n, \sigma_n, t)\,, \tag{18}$$

where μ, σ, and τ are initialized as $\mathbf{0}$, $\mathbf{1}$, and 0, respectively.

For cold-start and restart scenarios, the decaying factor λ would be approximately equal to 0, and the prediction would lean on $\hat{h}_u$ and $\hat{h}_n$ adaptively. For an accessing peak, every click would be cached in the EDC, and be considered when making recommendations. As time grows, the real-time interest captured by the EDC gradually decays towards the long-term interest modeled by the graph neural network, which presents the temporal dynamics of the user interests.

5 EXPERIMENTS

5.1 Datasets and Experimental Settings

Datasets. We conduct experiments on two widely used real-world datasets: Microsoft News Dataset (MIND) [52] and Adressa News Dataset (Adressa) [11]. Specially, we use the small version of MIND, which is collected from the user behavior logs of Microsoft News from October 12 to November 22, 2019, and the 1-week version of Adressa [3], which covers one week of the web traffic from February 2017 on the *www.adresseavisen.no* website. Detailed statistics of the two datasets are summarized in Table 3.

Table 3. Statistics of the two news recommendation datasets.

	# News	# Users	# Clicks	# Categories
MIND	65,238	94,057	347,727	18
Adressa	14,661	133,765	1,060,341	19

For the MIND dataset, we construct the news click history with records in the first four weeks, construct the training set and the validation set with records in the fifth week, and construct the test set with records in the last week. For the Adressa dataset, we construct the news click history with records in the first five days, construct the train set and the validation set with records on the sixth day, and construct the test set with records on the last day. Following previous work [36], we process the knowledge entities in the Adressa dataset with *Wikidata query* [4]. Negative samples for model training are collected from exposed but not clicked news in history for the MIND dataset, and randomly sampled from all candidate news uniformly for the Adressa dataset.

Implementation Details [5]. For attributed heterogeneous graph representation learning, we utilize two relational graph attentive layers, and the number of attention heads is set as 4. In each layer, we sample at most 15 neighbors for each relation without replacement. Note that only click records in the historical set are used for neighborhood sampling. The output dimensions of the heterogeneous node content adapter and the variational distribution are 128 and 64, respectively. For relation pruning, we set the pruning threshold ratio ϱ as 0.2 for all epochs and set the decaying factor γ as 0.9. For model optimization, we set the weight of KL divergence κ as $1e-4$. The number of variational distribution sampling is set as 3. The learning rate is set as $1e-4$. The ratio of negative sampling is set as 4. All models are trained for up to 50 epoches. For exponentially decaying caches, we build a

[3] Non-subscriber users and news without any keyword or entity are filtered out.

[4] https://query.wikidata.org/

[5] Our code will be publicly available upon acceptance of this paper: https://github.com/aSeriousCoder/DivHGNN

Just Accepted

cache for each user on each category with $M = 5$, $\beta = 0.2$, and $\rho = 0.25$. Following previous works [8, 31, 33, 49], we leverage **AUC**, **MRR**, **nDCG@5** and **nDCG@10** to evaluate the recommendation accuracy and normalized in-list average distance (**ILAD@5** and **ILAD@10**) to evaluate the recommendation diversity. Besides, we adopt the trade-off metric (**TO**) [33] to evaluate the overall recommendation performance based on nDCG and ILAD, which is defined as

$$\text{trade-off} = 2 * \text{accuracy} * \text{diversity}/(\text{accuracy} + \text{diversity}).$$

Higher trade-off values indicate better performance. For the sake of clarity, all the above metrics are expressed as percentages in the following tables. Each experiment is repeated 5 times and the average result is reported.

Just Accepted

5.2 Baseline Methods

To comprehensively evaluate the proposed method, we consider four types of state-of-the-art news recommendation methods: NLP-based methods (NAML [47], NRMS [48], LSTUR [1], and Tiny-NewsRec [60]), KG-based methods (DKN [43], KRED [22], and KOPRA [36]), GNN-based methods (GNewsRec [16], GERL [8], GNUD [17], User-as-Graph [49], and DIGAT [23]), and diversity-aware methods (D2NN [33], PP-Rec [30], HieRec [31], and GLORY [56]) in the comparison.

- NAML [47] extends the single-view news representation to multi-view setting and proposes an attentive learning model to learn unified news representation from titles, bodies and topic categories.
- NRMS [48] uses multi-head self-attentions to model word dependence for news representation and captures news relatedness for user representation.
- LSTUR [1] captures news representation from titles and topic categories and learns both long- and short-term user representation from IDs and recently browsed news.
- Tiny-NewsRec [60] attempts to improve both the effectiveness and the efficiency of PLM-based news recommendation models via self-supervised domain-specific post-training and two-stage knowledge distillation.
- DKN [43] incorporates knowledge graph representation into news recommendation. Especially, they fuse semantic-level and knowledge-level representations of news, and aggregate user clicking histories with current candidate news for user representation.
- KRED [22] incorporates knowledge entities for better news understanding. In detail, they aggregate information from entity neighborhoods to enrich embeddings.
- KOPRA [36] utilizes user interest-aware knowledge pruning to augment news content, and captures user interest through recurrent graph convolution.
- GNewsRec [16] constructs a heterogeneous user-news-topic graph and learns user/news embeddings from heterogeneous structural information as long-term interest. Short-term interests are captured by an attentive RNN.
- GERL [8] constructs a user-news bipartite graph by historical user clicks and models user-news relatedness in a graph setting. Especially, they utilize a transformer for news representation, and represent users via aggregating historically clicked news and high-order neighbor users.
- GNUD [17] regarded the user-news interactions as a bipartite graph, and learned disentangled user/news embeddings through graph convolution with neighborhood routing mechanism.
- User-as-Graph [49] models each user as a personalized attributed heterogeneous graph and aggregates neighborhood information via graph pooling, while the news is processed by a news encoder.
- DIGAT [23] augments the candidate news by incorporating the semantically relevant news in a semantic augmented graph (SAG) and captures multi-level user interests in a news-topic graph.

Table 4. Performance comparison between DivHGNN and sixteen state-of-the-art news recommendation methods on the MIND dataset. Boldface indicates the best overall performance. The results are expressed in percentages. TO is short for trade-off.

		MIND						
Model	Category	AUC	MRR	nDCG5	nDCG10	ILAD5	ILAD10	TO
NAML	NLP-based Models	64.34	29.8	32.61	39.1	38.3	42.17	37.92
NRMS		65.33	30.41	33.15	39.69	36.65	42.04	37.83
LSTUR		65.48	30.25	33.41	39.81	37.11	41.96	38.02
Tiny-NewsRec		67.53	32.15	36.06	41.22	48.54	52.48	43.79
DKN	Knowledge-based Models	64.02	29.03	31.7	38.42	41.97	47.73	39.36
KRED		65.76	30.62	33.57	40.25	44.82	49.22	41.35
KOPRA		65.98	31.17	34.24	40.71	42.52	48.83	41.17
GNewsRec	GNN-based Models	65.95	30.48	33.47	40.11	39.42	43.88	39.07
GERL		66.53	31.55	34.63	41.18	38.96	44.16	39.65
GNUD		66.5	31.42	34.48	41.03	39.74	44.37	39.79
User-as-Graph		67.47	32.41	35.72	42.07	39.35	44.85	40.43
DIGAT		67.91	32.06	36.39	42.75	41.69	46.82	41.78
D2NN	Diversity-aware Models	63.53	29.35	32.56	38.81	44.6	49.37	40.56
PP-Rec		66.36	31.25	34.04	40.77	47.86	52.59	42.88
HieRec		67.22	31.91	35.85	41.09	45.13	50.26	42.59
GLORY		67.24	31.77	35.19	41.40	52.69	54.48	44.67
DivHGNN	Ours	**68.41**	**34.04**	**37.02**	**43.03**	**56.03**	**61.62**	**47.64**

- D2NN [33] utilizes LSTM with orthogonality constraint and diversity-aware attention mechanism to learn distinctive user interests for news recommendation.
- PP-Rec [30] incorporates news popularity information and proposes to score candidate news with both personalized matching scores and news popularity scores.
- HieRec [31] represents each user with a hierarchical interest tree to capture their diverse and multi-grained interests and utilizes a hierarchical user interest matching framework to match candidate news with different levels of user interests.
- GLORY [56] builds a global-aware historical news encoder that combines the user's global representation with local representation based on a gated graph neural network to enhance the accuracy and diversity of personalized news recommendations.

All baseline models are trained using the Adam optimizer for up to 50 epochs. The learning rate is set to $1e-4$. Validation is performed every 1000 training iterations. When better model performance is not observed for 5 consecutive epochs in validation set, the model is regarded as converged and we stop the training. The configurable number of layers and the size of hidden dimensions are set to 2 and 128 for all models. Other complex configurations follow the default configuration of the open-source code repository. All models are trained for 5 times and the average results are reported.

5.3 Performance Evaluation

Table 4 and Table 5 compares the performance between DivHGNN and sixteen state-of-the-art news recommendation methods on both MIND and Adressa. We can observe from the results that:

Just Accepted

Table 5. Performance comparison between DivHGNN and sixteen state-of-the-art news recommendation methods on the Adressa dataset. Boldface indicates the best overall performance.

		Adressa						
Model	Category	AUC	MRR	nDCG5	nDCG10	ILAD5	ILAD10	TO
NAML	NLP-based Models	68.94	62.92	62.08	68.76	47.82	54.81	57.51
NRMS		68.47	62.41	61.9	68.62	46.29	54.44	56.85
LSTUR		67.8	61.75	61.34	68.27	46.83	54.79	56.96
Tiny-NewsRec		70.92	66.01	63.07	69.26	57.6	63.65	63.27
DKN	Knowledge-based Models	65.16	60.52	60.65	67.65	51.86	59.01	59.47
KRED		69.33	63.41	62.29	68.82	54.78	60.6	61.37
KOPRA		69.92	64.74	63.02	69.12	52.45	60.55	60.91
GNewsRec	GNN-based Models	69.54	63.82	62.35	68.84	49.39	56.86	58.70
GERL		70.49	64.67	62.74	68.93	49.03	57.31	58.83
GNUD		70.63	65.24	62.87	69.01	49.98	57.78	59.30
User-as-Graph		71.31	66.3	63.43	69.42	49.55	58.29	59.52
DIGAT		71.58	66.47	63.67	**70.15**	51.91	60.19	61.00
D2NN	Diversity-aware Models	67.25	62.88	60.82	67.63	53.83	61.86	60.87
PP-Rec		70.09	64.56	62.02	68.89	58.24	65.89	63.71
HieRec		70.91	65.31	62.91	69.28	55.41	64.16	62.78
GLORY		71.06	65.27	62.68	69.21	59.38	68.44	64.91
DivHGNN	Ours	**72.14**	**67.07**	**63.79**	70.11	**64.26**	**71.08**	**67.31**

- DivHGNN consistently outperforms all the compared methods, specifically, DivHGNN achieves the highest trade-off of 47.64% and 67.31% on MIND and Adressa, respectively. The reason for this high performance is that DivHGNN can integrate heterogeneous attributes and efficiently aggregate heterogeneous neighborhood information from most informative relations. Meanwhile, the proposed variational modeling and exponentially decaying cache could provide personalized and adaptive diversity, which further optimizes the trade-off for users in different browsing scenarios.
- The NLP-based methods rely on textual content for news encoding and model user preference by aggregating user click histories. Limited by the homogeneous content and relationship modeling, most of these methods present an inferior performance on both accuracy and diversity. But we can also find that with a fine-grained adaptation of the large language models (LLMs), NLP-based approaches could achieve exciting performance.
- Incorporating external information into the modeling process could enhance model performance on diversity. The knowledge-based methods introduce supplementary information from external knowledge graphs, while Tiny-NewsRec leverages the semantic information learned by the LLMs from the large-scale corpus. Both of them have achieved significant improvements in diversity.
- Along with diversified relationships, the model performance, especially in accuracy, is gradually enhanced. At the same time, the aggregation of higher-order neighborhood information could smooth the distribution of node representations and thus enhance the diversity of representation modeling. Our method extends this approach to the attributed heterogeneous information network, which benefits from learning both graph structural information and heterogeneous node content, achieving superior performance on both accuracy and diversity. Besides, it is noteworthy that DIGAT, which is based on semantic information for neighborhood enhancement, achieves superior accuracy compared to the topological neighborhood-based baselines, which indicates that the direct topological neighborhood has some noisy information with a negative impact, i.e., not all relations are

Just Accepted

useful. To deal with this challenge, we propose a novel relation pruning strategy, which helps identify and block neighborhood noise.

- Among all baselines, LSTUR, KOPRA, and GNewsRec take efforts to model long- and short-term interests for user representation. However, these methods require additional model training, and fail at modeling the fine-grained temporal dynamics between two click records. Our approach incorporates the parameter-free exponentially decaying cache to model the real-time user preference drifts, achieving improved performance on news recommendation.
- D2NN performs a relatively independent optimization for diversity, which compromises the accuracy to some extent. In contrast, our approach embeds the optimization of diversity into the optimizing process for accuracy, or implements as a plug-in that does not affect the optimizing process, which allows our model to achieve a better accuracy-diversity trade-off.
- PP-Rec and HieRec propose customized model designs that fit users' habits (leveraging popular news, inner-topic interest cohesiveness) through deep insights into users' news reading practices, which in turn improves model performance, especially in diversity. Similarly, the design of proposed exponentially decaying cache comes from an in-depth modeling of two main reading modes (casual browsing and digging out) and the transformation paradigm between them, which well matches the requirements for adaptive diversity.
- Among all baselines, GLORY's performance in diversity is outstanding. GLORY integrates global information into the user and news modeling process. Considering that users often present personalized preferences, the local information drawing from interaction history is often distributed differently from the global information. This difference brings diverse data inputs for user and news modeling, and ultimately produces more diverse recommendation results.
- Tiny-NewsRec presents higher recommendation diversity even than some diversity-aware models. This is because ILAD could be affected by the model training to a large extent. If the representation space is more dispersed, the ILAD would be higher even for the same list. Tiny-NewsRec fine-tunes and distills the pretrained Transformer, which has a more dispersed representation distribution than the pretrained Glove model used by diversity-aware baselines, resulting in a higher ILAD of Tiny-NewsRec. The baselines for the comparison of diversity-aware models are NAML, NRMS, and LSTUR. Based on these methods, diversity-aware models improve accuracy and diversity through model design (popularity-aware, hierarchical structure, etc.).
- All methods present a superior performance on Adressa over MIND, due to the negative samples of the two datasets were collected with different strategies. Negative samples of MIND come from exposed but not clicked news, which have already been filtered by the online news recommendation system, and thus are more semantically similar and more difficult to distinguish. In contrast, negative samples of Adressa come from random sampling, which are more differentiated. The conclusions drawn from the two datasets are essentially the same, and for the sake of brevity, we subsequently report the experimental results on the MIND dataset only.

5.4 Ablation Study

In this section, we compare the ablative variants of DivHGNN to illustrate the necessity of considering both heterogeneous contents and relationships in news recommendation and the importance of providing personalized and adaptive news diversity. We build seven ablative variants as follows: 1) DivHGNN without heterogeneous attributes (w/o H.A.), which replaces the multiple pretrained model-encoded node attributes and the heterogeneous node content adapter with trainable node embeddings, and learns the model parameters along with the embeddings; 2) DivHGNN without heterogeneous relationships (w/o H.R.), which ignores the difference among all relationships, and utilizes shared graph neural network layers to process neighborhood information from all kinds of relationships. 3) DivHGNN without the proposed heterogeneous node content adapter (w/o

Just Accepted

Table 6. Performance comparison among ablative variants on the MIND dataset.

Model	AUC	MRR	nDCG5	nDCG10	ILAD5	ILAD10	TO
DivHGNN w/o H.A.	67.09	32.05	35.05	41.32	53.27	56.34	45.01
DivHGNN w/o H.R.	67.21	32.2	35.12	41.46	51.35	53.97	44.34
DivHGNN w/o N.C.A.	67.78	32.15	35.42	41.71	53.43	56.76	45.37
DivHGNN FT	67.47	32.34	35.69	42.1	54.65	57.83	45.99
DivHGNN w/o P.D.	67.83	32.96	36.09	42.55	53.96	57.65	46.13
DivHGNN w/o A.D.	67.96	33.09	36.37	42.57	55.94	61.58	47.22
DivHGNN w/o Pr.	67.66	32.94	36.31	41.67	**56.36**	**62.51**	47.09
DivHGNN	**68.41**	**34.04**	**37.02**	**43.03**	56.03	61.62	**47.64**

N.C.A.), which replaces the adapter with a set of node-specific MLPs; 4) DivHGNN with fine-tuning (FT), which regards the multiple pretrained model-encoded node attributes as trainable embeddings. 5) DivHGNN without personalized diversity (w/o P.D.), which removes the proposed variational representation learning. 6) DivHGNN without adaptive diversity (w/o A.D.), which removes the proposed exponential decaying cache. 7) DivHGNN without pruning (w/o Pr.), which removes the proposed matapath pruning strategy.

From Table 6, we can observe that the performance of DivHGNN w/o H.A. and DivHGNN w/o H.R. decreases dramatically compared to DivHGNN, which illustrates the importance of modeling heterogeneous contents and heterogeneous relationships. More specifically, heterogeneous contents make an important contribution to improving diversity, while heterogeneous relationships play a greater role in improving accuracy.

DivHGNN FT presents a superior performance compared to DivHGNN w/o H.A., which illustrates the essential role of external information (LLM, KG, etc.) in improving the accuracy and diversity of news recommendations. Indeed, as Wu *et al.* mentioned in [51], exploiting the massive external information learned by pretrained models can improve news encoding, which has been widely adopted in comparing baseline models in various forms, such as parameter initialization and connectivity enhancement. However, the implementation schema of heterogeneous external information is decisive for performance improvement. Due to the difference in optimization goals between the pretraining and recommendation tasks, directly fine-tuning the final embeddings of pretrained models may be counterproductive. In the absence of tailored processing, the semantic gaps between different pretrained models could also make them constrain each other. As a consequence, compared with the fine-tuning schema in DivHGNN FT and the direct fusion schema in DivHGNN w/o N.C.A., regarding the pretrained models as independent encoders and utilizing a multi-modal adapter to align and fuse the heterogeneous content encodings, presents a superior performance.

Performance reduction is also observed on DivHGNN w/o P.D.. Without the variance modeling, the interest range of users and the audience range of news are considered as uniform, resulting in some inappropriate news recommendations, such as consistently pushing news on a single topic to high-variance users or pushing other non-interesting topics to low variance users. As a parameter-free plug-in, the exponential decaying cache does not affect the distribution of representation space, so its absence presents few impact on the diversity of DivHGNN w/o A.D.. But its auto-zooming feature well models the users' news reading habits and could obviously improve recommendation accuracy. Compared with DivHGNN w/o Pr., the proposed pruning strategy cuts out some of the low-importance relations, reducing data noise and thus improving accuracy. It also weakens the diversity of data sampling and negatively affects recommendation diversity. From a trade-off perspective, the overall benefits outweigh the drawbacks.

Just Accepted

5.5 Model Complexity Analysis

In this section, we theoretically and experimentally analyze the complexity of DivHGNN.

5.5.1 Theoretical Analysis. We begin by discussing the algorithmic complexity of DivHGNN. Assume there are N nodes in the network, each with M attributes on average. DivHGNN initially encodes the multiple attributes of these heterogeneous nodes based on the pre-trained models, which manifest as representation queries with a complexity of $O(Q \cdot M \cdot N)$, where Q denotes the operational complexity of performing a single query. Next, DivHGNN performs alignment and fusion of these encoded features for all nodes. This process consists of two MLP modules and one self-attention module. For clarity, we assume all MLP submodules in DivHGNN have L layers with D dimensions. The two MLP modules operate on per-feature and per-node separately, with an overall complexity of $O((M+1) \cdot N \cdot L \cdot D^2) \approx O(M \cdot N \cdot L \cdot D^2)$. The complexity of one self-attention operation is $O(n \cdot d^2 + n^2 \cdot d)$, where n is the sequence length, and d is the feature dimension. In our context, n is the number of node features M, and d is the MLP dimension D. The self-attention is conducted at the node level, executed N times. Since M is much smaller than D, the complexity of the self-attention module is $O(M \cdot N \cdot D^2)$. Therefore, the overall complexity of alignment and fusion is $O(M \cdot N \cdot L \cdot D^2 + M \cdot N \cdot D^2) \approx O(M \cdot N \cdot L \cdot D^2)$. Thirdly, the graph neural network of DivHGNN consists of G layers of message passing. A single message passing first transforms each message through an MLP, followed by message aggregation using self-attention. Assuming the average number of relations per node is R, and the neighbor sampling scale is S, the overall complexity of the graph neural network is $O(G \cdot N \cdot (R \cdot S \cdot L \cdot D^2 + R \cdot S \cdot D^2)) \approx O(G \cdot N \cdot R \cdot S \cdot L \cdot D^2)$. Lastly, the computational complexity of the representation sampling and EDC module can be considered as querying operations at the node level, with algorithmic complexities of $O(Q \cdot N_{\text{User \& News}}) < O(Q \cdot N)$ and $O(Q \cdot N_{\text{user}} \cdot T) < O(Q \cdot N \cdot T)$, where T is the number of topics. Upon consolidating the complexity of each module, we obtain the overall algorithmic complexity of DivHGNN:

$$O(Q \cdot N \cdot (M+T) + (M + G \cdot R \cdot S) \cdot N \cdot L \cdot D^2).$$

5.5.2 Experimental Analysis. Table 7 presents a comparison between DivHGNN and some baseline models in terms of the size of trainable model parameters and the training time. In terms of model parameter size, trainable embeddings account for a large part. In comparison, considering the pretrained models as independent embedding lookup tables significantly reduces the scale of trainable parameters, and each word or entity only needs to be queried once at its first occurrence, improving the computation reusability. This efficiency could be further boosted via engineering efforts, such as encoding reuse and multi-line deployment, which we will further

Table 7. Number of trainable parameters and training time comparison.

Model	Param Size (MB)			Training Time		
	Total	Embedding	Network	Total	Epoch Time	Num Epoch
NRMS	26.40	25.31	1.09	1h54m35s	9m9s	12.53
DKN	9.55	8.44	1.11	3h41m38s	12m11s	18.17
DIGAT	8.86	4.60	4.26	8h10m45s	20m45s	23.65
HieRec	18.63	15.61	3.02	14h40m38s	52m46s	16.69
GLORY	61.34	51.49	9.85	4h24m34s	17m10s	15.41
DivHGNN FT	28.50	25.49	3.01	5h48m25s	16m19s	21.36
DivHGNN w/o Pr.	3.71	-	3.71	6h30m27s	14m55s	26.16
DivHGNN	3.01	-	3.01	5h35m31s	13m32s	24.78

Just Accepted

Table 8. Performance with different number of layers and neighbors on the MIND dataset.

Layers	Neighbors	AUC	MRR	nDCG5	nDCG10	ILAD5	ILAD10	TO
1	5	65.68	31.11	34.01	40.56	50.8	56.03	43.92
	10	66.52	31.87	34.78	41.26	53.44	59.15	45.39
	15	67.17	32.42	35.2	41.51	54.83	60.96	46.14
2	5	66.21	31.66	34.41	40.97	53.16	59.03	45.09
	10	67.69	32.89	35.99	41.95	55.35	60.75	46.63
	15	**68.41**	**34.04**	**37.02**	**43.03**	56.03	61.62	**47.64**
3	5	65.93	31.37	34.26	40.64	53.87	59.65	45.13
	10	67.2	32.35	35.03	41.38	55.85	62.01	46.36
	15	67.84	32.96	35.85	42.17	**56.37**	**63.04**	47.19

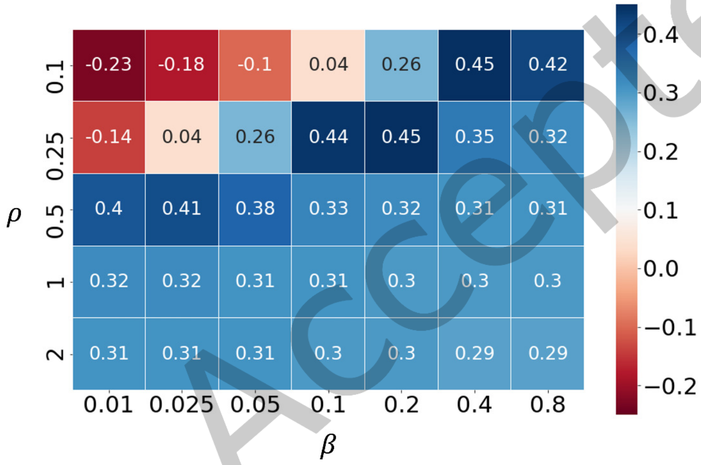


Fig. 6. Performance improvement with different β and ρ on the MIND dataset.

discuss in Section 6.1. We could also see from Table 7 that the neural network parameter scale of DivHGNN is comparable with baseline models, despite of implemented with advanced techniques. In terms of training time, DivHGNN also presents competitive performance. NRMS is a strong baseline model known for its compact design, which leads to high computational efficiency and fast convergence. Introducing features like heterogeneous information, complex relationships, and hierarchical structures does improve performance, but it also makes the model take longer to compute and slows down the convergence process. DivHGNN optimizes the utilization of these features, making it compute faster. However, the use of various advanced techniques inevitably brings challenges to convergence, leading to a higher average number of training epochs for DivHGNN. Balancing these two aspects, DivHGNN presents a decent convergence speed compared to other baseline models. We also compare two variants of DivHGNN. The DivHGNN FT transforms the attribute encodings as trainable embeddings, which makes training more complex and increases the time per epoch, but it also makes the model converge faster. The DivHGNN w.o. Pr. cuts down on computation by ignoring relationships with low contributions, which also makes the model converge faster.

Just Accepted

5.6 Parameter Sensitivity Analysis

This section explores how hyper-parameters affect the proposed DivHGNN and examines the impacts of two key hyper-parameters: the number of GNN layers and neighbors, and the setting of decaying controllers.

Table 8 presents the performance comparison of the DivHGNN variants with different numbers of GNN layers (from 1 to 3) and neighbors (5, 10, 15 per sampling) on the MIND dataset. From the perspective of accuracy, DivHGNN With 2 GNN layers yields the best performance. DivHGNN with 1 GNN layer aggregates neighborhood information only from directly interacted nodes (one-hop neighbors) and thus can not exploit the high-order connectivity. As the result, it is presented with limited performance. On the other hand, DivHGNN with 3 GNN layers takes too much noisy information from distant neighbors and starts to suffer from the over-smoothing issue, resulting in an unsatisfying performance as well. From the perspective of diversity, increasing the number of GNN layers can enhance the data diversity of neighborhood information and thus improve the recommendation diversity. In contrast, when increasing the number of neighbors sampled, the model could learn from a more diverse and stable distribution, and as a result, both the performance on accuracy and diversity improve steadily. But the marginal gain of this improvement is diminishing as the number of samples rises.

We further explore how the hyper-parameters β and ρ of EDC influence the recommendation accuracy. Figure 6 shows the performance improvement of DivHGNN with varying β and ρ compared to DivHGNN w/o EDC (w/o A.D.), in terms of AUC (%) on the MIND dataset. By adjusting the values of β and ρ, we can approximately observe the real-time user interest decaying process, corresponding to the optimal performance ridge in Figure 6. Specially, DivHGNN with EDC ($\beta = 0.2$ and $\rho = 0.25$) outperforms DivHGNN w/o EDC by 0.45% in terms of AUC. However, the EDC presents a negative effect when the values of β and ρ are low, in which situations the cached short-term interest decays too slowly. This phenomenon indicates that user short-term interest modelling that are not updated in a timely manner may do harm to the model performance due to the out-of-date trouble, and also confirms the necessity to support adaptive diversity in real-time.

5.7 Relation Pruning Analysis

Parameter pruning is often trained with a pruning schedule. In this section, we compare the effects of different pruning schedules on the DivHGNN. Specifically, we compare the impact of the two most widely used pruning schedules, in terms of accuracy, diversity and efficiency, to illustrate the positive effects of applying relation pruning and the special features of relation pruning task compared to other parameter pruning tasks:

- Pruning-during-training, which starts model pruning before it has been trained to convergence. In detail, we set the pruning threshold ratio ϱ as 0.2 for all the 50 epochs.
- Train-then-pruning, which uses a standard training procedure that is run to convergence followed by a pruning procedure and a fine-tune procedure. In detail, we set the pruning threshold ratio ϱ as 0.2 for the 20-th, 30-th, and 40-th epoch, and as 0 for the other epochs.

Just Accepted

Table 9. Performance with different relation pruning schedules on the MIND dataset.

schedule	AUC	MRR	nDCG10	ILAD10	TO	Param Size	Training Efficiency	Inference Efficiency
pruning-during-training	**68.41**	**34.04**	**43.03**	61.62	**47.64**	3.01M	1.09x	1.14x
train-then-pruning	67.94	33.27	42.05	61.73	47.18	3.01M	1.07x	1.14x
DivHGNN w/o Pr.	67.66	32.94	41.67	**62.51**	47.09	3.71M	1x	1x

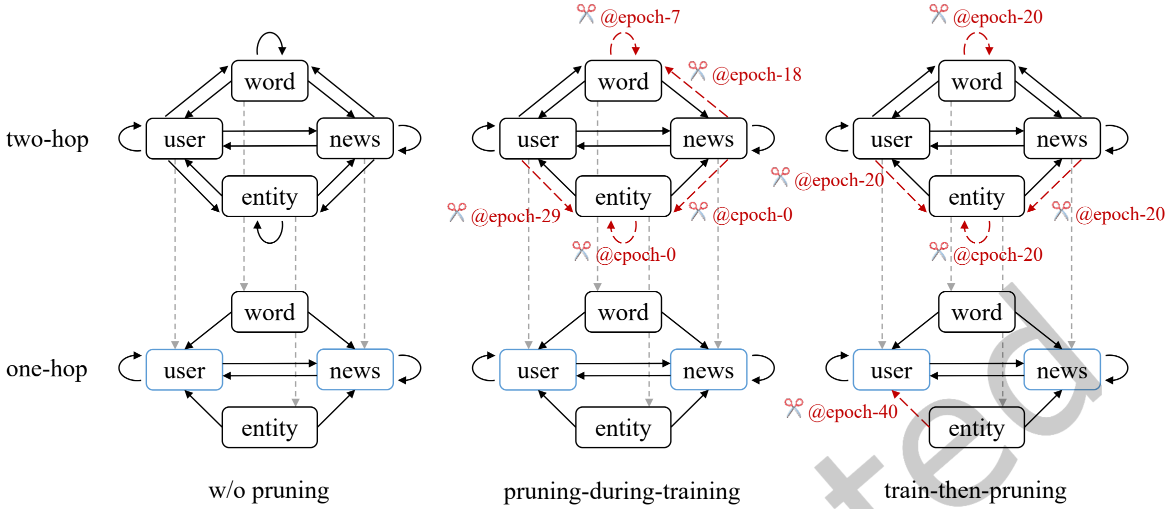


Fig. 7. Relational meta graph under different relation pruning schedules.

Table 10. Recommendation diversity for users of different variance.

	ILAD5(%)	ILAD10(%)	Category5	SubCategory5	Category10	SubCategory10
High variance users	66.58	66.97	6.53	11.37	9.68	19.87
Low variance users	63.73	64.25	6.07	10.03	8.73	18.14

Table 9 presents the performance comparison results. Compared to the non-pruned model, both schedules improve the model accuracy and efficiency, at the cost of a minor reduction in diversity. The pruning-during-training schedule outperforms the train-then-pruning schedule on both the accuracy-diversity trade-off and model efficiency, which is different from the conclusions under the other model pruning tasks. In general, the train-then-pruning schedule could generate pruned models with higher quality, because that it could more accurately identify the unimportant parameters as the pruning model has been converged. In contrast, the pruning-during-training schedule prunes before the model converges, which has a higher risk of misidentification, but it improves the pruning efficiency as it does not need to wait for the model to converge. There is an implicit assumption in the above conclusions that the importance of parameters is acquired gradually during the learning process. The basis of this assumption is that the structures to be pruned are replicas of each other and have no inherent differences. However, relation pruning does not satisfy this basic condition. As we define the pruning blocks as relation-associated neural modules, which are not interchangeable and of inherent importance, the learning process no longer grants importance to neurons, but merely models their importance. This change makes the results of importance identification more stable, and we can prune with less risk before the model converges as full structured, which in turn leads to better convergence and performance of the pruned model. Figure 7 illustrates the relational meta graph under different schedules. The pruning results based on the two schedules are basically the same, but the pruning-during-training schedule can identify unimportant relations much earlier.

5.8 Personalized Diversity Analysis

In this section, we analyze how DivHGNN provides news recommendations for users with personalized diversity.

Just Accepted

Table 11. Top-5 recommendation lists for high-variance user U24546.

ID	Category	SubCategory	Title
N5229	sports	football_nfl	Enemy Reaction 2019: Tampa Bay Buccaneers (Part 1)
N50123	finance	financenews	Gronk's CBD Company To Advertise At Patriot Place, ...
N64300	travel	travelarticle	Hiking to Havasupai: How far is it, what's Mooney Falls ...
N13859	lifestyle	halloween	How the world's largest candy company prepares for Halloween
N40249	foodanddrink	newstrends	This NC City Is Home To The Best Food Truck In State
N59922	sports	football_nfl	Grading all 32 NFL teams heading into the last eight weeks ...
N33730	weather	weathertopstories	Picturesque Southern Michigan Drone Footage Shows ...
N20455	foodanddrink	newstrends	This Halal buffet in the heart of the Highlands is ...
N15569	foodanddrink	recipes	18 Recipes That Start with a Bag of Peanut Butter Chips
N12456	lifestyle	lifestylebuzz	NC Mom of 4 Goes to Sheriff's Office to be Fingerprinted, ...
N14183	music	musicnews	Jaguars vs. Jets: Jawaan Taylor was a surprise musical guest ...
N62448	autos	autosnews	Hurricane Dorian Jeep makes an appearance at Pennzoil AutoFair
N57619	news	newsus	Top Seattle news: Poll: voters distrust city council
N11785	travel	travelnews	8 photos of Cori Copley, the adorable black lab who ...
N32836	sports	golf	Dominance: Tiger Woods ties Sam Snead's record with ...

Table 12. Top-5 recommendation lists for low-variance user U88457.

ID	Category	SubCategory	Title
N52333	music	musicnews	20 Super Successful Musicians Who Got Their Start ...
N7804	sports	boxing	Julio Cesar Chavez on Canelo Alvarez: 'Many people ...'
N7313	health	wellness	Fitbit's Lead Sleep Research Scientist Shares ...
N43839	music	musicnews	John Legend Is Revamping Troublesome Christmas ...
N45149	sports	basketball_nba	Watch: Spurs guard Dejounte Murray hits ...
N34015	sports	football_nfl	Steelers re-sign RB Darrin Hall to the practice squad
N39027	video	animals	'Red tide' toxic algae bloom kills sea life and ...
N35058	news	newspolitics	Florida voter: Trump's helping our economy grow
N32225	sports	basketball_nba	Pacers Links: Pacers surviving and thriving thanks to ...
N58057	music	music-celebrity	Lady Gaga is in 'a lot of pain' following onstage fall
N54891	sports	football_nfl	Chargers owner Dean Spanos shoots down report of ...
N37817	news	newsscienceandtechnology	AI project to preserve people's voices in effort to ...
N61399	sports	football_nfl	OPEN THREAD: Bengals vs. Jaguars pregame
N60112	news	newscrime	NYPD: Armed Robbers Targeting People With ...
N52874	entertainment	entertainment-celebrity	John Legend clarifies his recent Kanye West comments

Just Accepted

Following Definition 3, we compare the inner-list diversity of the recommendation lists generated by DivHGNN for users with different levels of historical clicked news diversity. For each testing user in the MIND dataset whose variance of interests is among the highest 20% or the lowest 20%, we generate recommendation lists containing top-5/top-10 scored news for 3 times with exposed news filtering from all the testing news using DivHGNN. Table 10 presents the top-5/top-10 inter-list average distance (ILAD) among the 3 recommendation lists and the average number of categories/subcategories covered by the recommendation results. For the high-variance users, of whom the historically clicked news is more diverse, DivHGNN enhances the inter-list diversity by smoother representation sampling and achieves a higher topic coverage. And vice versa for the low-variance users.

As a more direct comparison, Table 11 and 12 present the detailed top-5 recommendation lists of two randomly sampled users (U24546 from the high-variance group and U88457 from the low-variance group). The recommendation lists of U24546 present a high topic diversity, with 15 news stories covering 9 topics and 13 sub-topics. In contrast, the recommendation lists of U88457 are more focused on specific topics, especially sports news, which appears up to 6 times.

5.9 Adaptive Diversity Analysis

In this section, we analyze how DivHGNN provides news recommendations for users with adaptive diversity. Following Definition 4, we discuss the real-time adaption of recommendation results generated by DivHGNN through a representative case in testing set: user U46312 clicks on news N63421 (Travel, "Australia's Qantas operates 19 ½ hour London-Sydney flight").

Table 13 presents the top-5 recommendation lists of user U46312 at four specific times: before click, on click, 1 minute after click, and 1hour after click. At the moment the user clicks on the news, DivHGNN makes a real-time update of the user interest modeling. ILAD@BC increases, reflecting the growing semantic difference between the real-time user modeling and the long-term user representation. ILAD@OC decreases, showing that the generated news recommendation list is more semantically similar to the clicked news. This process demonstrates the adaptive zooming in of DivHGNN from extensively browsing to targeted reading. Then as time goes on, if no new click behavior arises, The real-time user modeling gradually decays from relying on the clicked news to the long-term user representation, resulting in decreasing ILAD@BC and increasing ILAD@OC. If the user continues to click on some news, the zooming-in process would be refreshed and the decaying process would be reset.This process demonstrates the adaptive zooming out of DivHGNN from targeted reading back to extensively browsing. Table 13 also presents the topics of recommended news. After clicking on some travel news, the user who originally focus on sports news can also get plenty of travel-related recommendations from DivHGNN. If the user no longer clicks on travel news, DivHGNN would adaptively re-recommend sports news.

5.10 Cold-start News Recommendation

One of the biggest challenges of news recommendation is the cold-start problem, which typically refers to the limited ability to recommend news without any historical user interactions from the massive incoming news. To evaluate how DivHGNN performs in the cold-start situation, we compare DivHGNN with a cold start-supported

Table 13. Top-5 recommendation lists at different times for user U46312 who clicked on the news N63421. ILAD@BC and ILAD@OC refer to the inter-list average distance with the before-click recommendation lists and on-click recommendation lists, respectively.

Before Click		On Click		1 Minute After Click		1 Hour After Click	
ILAD@BC	ILAD@OC	ILAD@BC	ILAD@OC	ILAD@BC	ILAD@OC	ILAD@BC	ILAD@OC
51.58	71.24	71.24	50.99	70.17	66.39	63.74	70.38
NewsID	Category	NewsID	Category	NewsID	Category	NewsID	Category
N20516	sports	N50683	weather	N47434	news	N26527	sports
N54784	sports	N32840	travel	N49635	news	N35630	video
N19510	entertainment	N38704	news	N61259	sports	N4882	news
N27767	sports	N53268	sports	N8287	travel	N59524	sports
N8924	news	N6742	foodanddrink	N44908	foodanddrink	N223	music

Just Accepted

Table 14. Performance comparison on cold-start news on the MIND dataset

Model	AUC	MRR	nDCG5	nDCG10	ILAD5	ILAD10	TO
DIGAT (cold-start)	67.29	31.53	35.6	41.91	39.55	41.97	39.73
DIGAT	67.91	32.06	36.39	42.75	40.69	43.82	40.87
DivHGNN (cold-start)	67.83	32.87	36.15	42.41	51.83	56.46	45.53
DivHGNN	**68.41**	**34.04**	**37.02**	**43.03**	**56.03**	**61.62**	**47.64**

baseline, DIGAT, on the MIND dataset. Specifically, we introduce a new test set for the cold-start problem by masking the clicked history of news in the original test set.

Table 14 shows the performance comparisons with DIGAT in both cold-start and warm-start settings. From the table, we observe that the performance of DivHGNN (cold-start) yields weaker performance on both accuracy and diversity compared to DivHGNN, which occurs with DIGAT as well. DivHGNN shows superior performance compared with DIGAT (cold-start) and other baseline models in the warm-start setting. Differently, DivHGNN and DIGAT leverage different cold-start strategies. DIGAT performs neighborhood enhancement based on semantic similarity, which in turn embeds cold-start news into the existing representation space. DivHGNN establishes meaningful links between the cold-start news and the heterogeneous information network via words and knowledge entities, and then utilize the robustness of heterogeneous graph neural networks to complement the representations of cold start news. Furthermore, since the attributed heterogeneous graph representation can be deployed in NRT, cold-start news can quickly obtain recommendations rather than waiting for a new round of offline embedding learning.

6 DISCUSSION

6.1 Model Efficiency

In this section, we discuss model efficiency and considerations on how to deploy DivHGNN into production.

As shown in the Table 7, existing news recommendation approaches focus on embedding learning for both users and news. Given the size of the embeddings and the computational complexity, existing methods are often deployed in production offline, and the generated embeddings are served online regularly. However, such offline-online structures limit the ability to update user and news representations frequently and thus expose challenges to making news recommendations up-to-date. To improve upon existing limitations, each module of DivHGNN is deployed in production offline, near real-time (NRT), and online respectively, and the improved three-tier architecture help maximize the update frequency of representations and make fresh recommendations, with a minimized computational cost. Figure 8 illustrates the deployment architecture of DivHGNN.

Specifically, we export the embedding matrices of the pretrained models to deploy as lookup tables. Each word or entity only needs to be queried once at its first occurrence. The heterogeneous node content adapter and the attributed heterogeneous graph neural network are trained offline, considering the relatively high computational cost. As the node content is static, its encoding calls for an update only when the adapter is re-trained, and thus deployed offline as well. From offline training, the attributed heterogeneous graph neural network projects the node content encodings into the representation space via neighborhood sampling and aggregation, which is deployed as a service offline or NRT, depending on the object. The audience of existing news is relatively static, thus its representation is deployed offline, while the users and the cold-starting news vary dynamically, calling for a more prompt update in NRT. Mention that the adapter and the GNN provide service independently, invoking a representation service in NRT only activates the GNN. Finally, the computationally efficient EDC is deployed online to support real-time user interest modeling and adaptive diversity modeling. EDC can be deployed on the client side as well, if the computational workload of the server is a concern.

Just Accepted

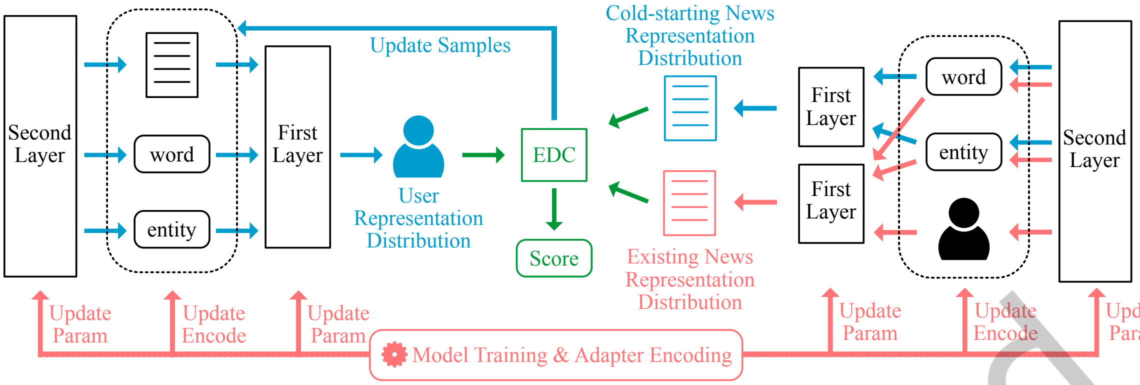


Fig. 8. The deployment architecture of DivHGNN. Data flows colored with red are deployed offline with a daily update frequency. Data flows colored with blue are deployed near real-time (NRT), which updates in hours or minutes. Data flows colored with green are deployed online, which would provide real-time services.

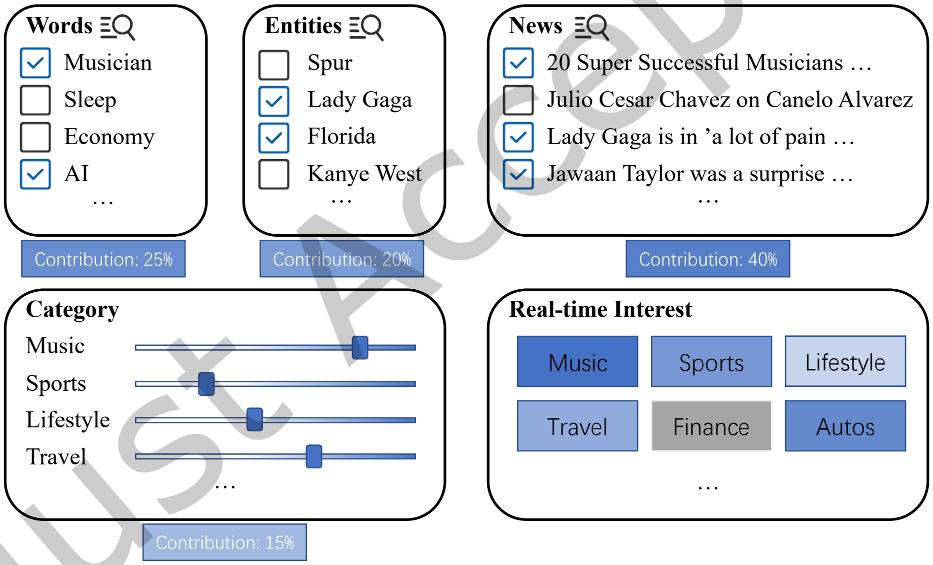


Fig. 9. Building user-controllable news recommender system based on DivHGNN.

It should be noted that one may be able to increase the training efficiency of RNN-based approaches by caching the latest hidden representations and only conducting incremental updates in the NRT environment. However, as discussed in Section 4.6, RNN-based approaches may not model the real-time shifting of user interest between two interactions explicitly and thus fail in providing adaptive news diversity, limiting their ability to effectively update user and news representation in reality.

Just Accepted

6.2 Building User-Controllable News Recommender System

This section discusses how to build a user-controllable news recommender system based on DivHGNN.

User-controllable news recommender systems attempt to enhance the user experience by providing users with a simple and easy-to-understand interface that allows them to adjust the recommender system in a participatory manner. Some online news platforms, such as TikTok, support users to control the frequency of recommended news from different topics via sliders. Compared to the coarse-grained control of existing approaches, DivHGNN allows the users to fine-tune the news recommendation from multiple views.

Figure 9 presents a demo interface of DivHGNN. At the data level, DivHGNN opens up the participatory neighborhood sampling and thus the users can freely select and mask the historically interacted words, entities, and news via a multi-select box, or actively add more candidates via search. Users could also configure their preferences for topics via the sliders. The configured preferences temporarily override the "Category Interaction Statistics" node attribute and are then integrated into the user modeling. At the model level, the interface presents the contribution (the average α value of the last graph attention layer in percentage) of each one-hop relation in the form of numbers and transparency. The users can manually adjust the model's dependency on a specific path by modifying the corresponding contribution value. The above two levels of control affect the processes of neighborhood sampling and the inference of heterogeneous graph representation respectively, whose computation could both be deployed in the NRT, thus supporting a fast response to the user adjustments. In addition, the interface makes real-time interest modeling transparent by showing the interest decaying process for each topic in the EDC through transparency. Users can intervene in the decaying process, for example by masking, to correct the gap between modeling and real interests.

6.3 Limitation

In this section, we discuss the limitation of the proposed DivHGNN model.

Although we have improved the efficiency of model updating and cold-start news recommendation by emphasizing the architecture design principles of functionality and modularity, we acknowledge that the architecture of DivHGNN is relatively complex and may require more engineering effort for deployment due to the utilization of the hierarchical service architecture. In particular, although it is beyond the scope of this paper, how to efficiently manage the large-scale graph data, including data I/O, graph operations, and neighbor sampling, involve crucial engineering challenges and can be critical bottlenecks to the system performance. In our implementation, we leveraged the Deep Graph Library (DGL, single machine version) [44] as the supporting graph engine. However, since the full graph is loaded into the memory, our experiments are limited by the scale of the graph data that the experimental machine can load at once. It affects our choice of the experimental dataset (MIND-small is used rather than the full version).

In DivHGNN, heterogeneous information is encoded into node features through multiple pre-trained models. This approach enables DivHGNN to make full use of multi-modal news content. However, due to the lack of data and suitable general-purpose pre-trained models, this paper only discusses part of the information in the rich news content.

7 CONCLUSION

In this paper, we present a novel news recommendation method with personalized and adaptive diversity named DivHGNN, which models user and news representative distributions with both heterogeneous attributes and relationships. We propose a heterogeneous node content adapter that aligns and fuses various attributes for better news understanding. Through the proposed heterogeneous graph neural network with relation pruning, DivHGNN identifies the important relations and learns the variational distributions of user and news representations by considering user and news variances. Based on the variational distributions, DivHGNN could

Just Accepted

provide news recommendations with personalized diversity. We utilize an exponentially decaying distribution cache to model user real-time interests for adaptive news diversity. Extensive experiments on real-world datasets demonstrate the superiority of DivHGNN on both news recommendation accuracy and diversity.

8 ACKNOWLEDGEMENTS

This work was supported by the National Natural Science Foundation of China (NSFC) under Grants Nos. 62172106 and 61932007. We would like to thank the anonymous reviewers and appreciate their insightful comments that helped improve the quality of the paper. Tun Lu is also a faculty of Shanghai Key Laboratory of Data Science, Fudan Institute on Aging, MOE Laboratory for National Development and Intelligent Governance, and Shanghai Institute of Intelligent Electronics & Systems, Fudan University. Hansu Gu and Tun Lu are the corresponding authors.

REFERENCES

[1] Mingxiao An, Fangzhao Wu, Chuhan Wu, Kun Zhang, Zheng Liu, and Xing Xie. 2019. Neural news recommendation with long-and short-term user representations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 336–345.

[2] Simran Arora, Avner May, Jian Zhang, and Christopher Ré. 2020. Contextual embeddings: When are they worth it?. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 2650–2663.

[3] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems*, Vol. 26. 2787–2795.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 4171–4186.

[5] Xiaohan Ding, Xiangxin Zhou, Yuchen Guo, Jungong Han, Ji Liu, et al. 2019. Global sparse momentum sgd for pruning very deep neural networks. *Advances in Neural Information Processing Systems* 32 (2019).

[6] Thomas Elsken, Jan Hendrik Metzen, and Frank Hutter. 2019. Neural architecture search: A survey. *The Journal of Machine Learning Research* 20, 1 (2019), 1997–2017.

[7] Hongchang Gao and Heng Huang. 2018. Deep attributed network embedding. In *Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI))*.

[8] Suyu Ge, Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. 2020. Graph enhanced representation learning for news recommendation. In *Proceedings of the Web Conference 2020*. 2863–2869.

[9] Anupriya Gogna and Angshul Majumdar. 2017. DiABlO: Optimization based design for improving diversity in recommender system. *Information Sciences* 378 (2017), 59–74.

[10] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 855–864.

[11] Jon Atle Gulla, Lemei Zhang, Peng Liu, Özlem Özgöbek, and Xiaomeng Su. 2017. The adressa dataset for news recommendation. In *Proceedings of the 2017 International Conference on Web Intelligence*. 1042–1048.

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

[13] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 639–648.

[14] Torsten Hoefler, Dan Alistarh, Tal Ben-Nun, Nikoli Dryden, and Alexandra Peste. 2021. Sparsity in deep learning: Pruning and growth for efficient inference and training in neural networks. *The Journal of Machine Learning Research* 22, 1 (2021), 10882–11005.

[15] Shifu Hou, Yujie Fan, Yiming Zhang, Yanfang Ye, Jingwei Lei, Wenqiang Wan, Jiabin Wang, Qi Xiong, and Fudong Shao. 2019. αcyber: Enhancing robustness of android malware detection system against adversarial attacks on heterogeneous graph based model. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 609–618.

[16] Linmei Hu, Chen Li, Chuan Shi, Cheng Yang, and Chao Shao. 2020. Graph neural news recommendation with long-term and short-term interest modeling. *Information Processing & Management* 57, 2 (2020), 102142.

[17] Linmei Hu, Siyong Xu, Chen Li, Cheng Yang, Chuan Shi, Nan Duan, Xing Xie, and Ming Zhou. 2020. Graph neural news recommendation with unsupervised preference disentanglement. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 4255–4264.

[18] Elvin Isufi, Matteo Pocchiari, and Alan Hanjalic. 2021. Accuracy-diversity trade-off in recommender systems via graph convolutions. *Information Processing & Management* 58, 2 (2021), 102459.

Just Accepted

[19] Zhenyan Ji, Mengdan Wu, Hong Yang, and José Enrique Armendáriz Íñigo. 2021. Temporal sensitive heterogeneous graph neural network for news recommendation. *Future Generation Computer Systems* 125 (2021), 324–333.

[20] Sian Jin, Sheng Di, Xin Liang, Jiannan Tian, Dingwen Tao, and Franck Cappello. 2019. DeepSZ: A novel framework to compress deep neural networks by using error-bounded lossy compression. In *Proceedings of the 28th international symposium on high-performance parallel and distributed computing*. 159–170.

[21] Diederik P Kingma and Max Welling. 2014. Auto-encoding variational bayes. In *Proceedings of the 2nd International Conference on Learning Representations*.

[22] Danyang Liu, Jianxun Lian, Shiyin Wang, Ying Qiao, Jiun-Hung Chen, Guangzhong Sun, and Xing Xie. 2020. KRED: Knowledge-aware document representation for news recommendations. In *Proceedings of the 14th ACM Conference on Recommender Systems*. 200–209.

[23] Zhiming Mao, Jian Li, Hongru Wang, Xingshan Zeng, and Kam-Fai Wong. 2022. DIGAT: Modeling News Recommendation with Dual-Graph Interaction. In *Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*. Association for Computational Linguistics, 6595–6607.

[24] Ashutosh Nayak, Mayur Garg, and Rajasekhara Reddy Duvvuru Muni. 2023. News Popularity Beyond the Click-Through-Rate for Personalized Recommendations. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1396–1405.

[25] Shumpei Okura, Yukihiro Tagami, Shingo Ono, and Akira Tajima. 2017. Embedding-based news recommendation for millions of users. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1933–1942.

[26] Guosheng Pan, Yuan Yao, Hanghang Tong, Feng Xu, and Jian Lu. 2021. Unsupervised attributed network embedding via cross fusion. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. 797–805.

[27] Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*. 1532–1543.

[28] Owen Phelan, Kevin McCarthy, Mike Bennett, and Barry Smyth. 2011. Terms of a feather: Content-based news recommendation and discovery using twitter. In *Proceedings of the 2011 European Conference on Information Retrieval*. 448–459.

[29] Bryan A Plummer, Nikoli Dryden, Julius Frost, Torsten Hoefler, and Kate Saenko. 2020. Neural parameter allocation search. *arXiv preprint arXiv:2006.10598* (2020).

[30] Tao Qi, Fangzhao Wu, Chuhan Wu, and Yongfeng Huang. 2021. PP-Rec: News Recommendation with Personalized User Interest and Time-aware News Popularity. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 5457–5467.

[31] Tao Qi, Fangzhao Wu, Chuhan Wu, Peiru Yang, Yang Yu, Xing Xie, and Yongfeng Huang. 2021. HieRec: Hierarchical User Interest Modeling for Personalized News Recommendation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*. 5446–5456.

[32] Lijing Qin and Xiaoyan Zhu. 2013. Promoting diversity in recommendation by entropy regularizer. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*. 2698–2704.

[33] Shaina Raza and Chen Ding. 2021. Deep Neural Network to Tradeoff between Accuracy and Diversity in a News Recommender System. In *Proceedings of the 2021 IEEE International Conference on Big Data*. 5246–5256.

[34] Chuan Shi, Xiao Wang, and S Yu Philip. 2022. Heterogeneous Graph Representation Learning and Applications. , 318 pages.

[35] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* 15, 1 (2014), 1929–1958.

[36] Yu Tian, Yuhao Yang, Xudong Ren, Pengfei Wang, Fangzhao Wu, Qian Wang, and Chenliang Li. 2021. Joint Knowledge Pruning and Recurrent Graph Convolution for News Recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 51–60.

[37] Zhaopeng Tu, Yang Liu, Shuming Shi, and Tong Zhang. 2018. Learning to remember translation history with a continuous cache. *Transactions of the Association for Computational Linguistics* 6 (2018), 407–420.

[38] Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha P. Talukdar. 2020. Composition-based Multi-Relational Graph Convolutional Networks. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*.

[39] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, Vol. 30. 5998–6008.

[40] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2018. Graph attention networks. In *Proceedings of the 6th International Conference on Learning Representations*.

[41] Chenyang Wang, Min Zhang, Weizhi Ma, Yiqun Liu, and Shaoping Ma. 2019. Modeling item-specific temporal dynamics of repeat consumption for recommender systems. In *The World Wide Web Conference*. 1977–1987.

[42] Chenyang Wang, Min Zhang, Weizhi Ma, Yiqun Liu, and Shaoping Ma. 2020. Make it a chorus: knowledge-and time-aware item modeling for sequential recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 109–118.

Just Accepted

[43] Hongwei Wang, Fuzheng Zhang, Xing Xie, and Minyi Guo. 2018. DKN: Deep knowledge-aware network for news recommendation. In *Proceedings of the 2018 World Wide Web Conference*. 1835–1844.

[44] Minjie Wang, Da Zheng, Zihao Ye, Quan Gan, Mufei Li, Xiang Song, Jinjing Zhou, Chao Ma, Lingfan Yu, Yu Gai, Tianjun Xiao, Tong He, George Karypis, Jinyang Li, and Zheng Zhang. 2019. Deep Graph Library: A Graph-Centric, Highly-Performant Package for Graph Neural Networks. *arXiv preprint arXiv:1909.01315* (2019).

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

[46] Nikolas Wolfe, Aditya Sharma, Lukas Drude, and Bhiksha Raj. 2017. The incredible shrinking neural network: New perspectives on learning representations through the lens of pruning. (2017).

[47] Chuhan Wu, Fangzhao Wu, Mingxiao An, Jianqiang Huang, Yongfeng Huang, and Xing Xie. 2019. Neural news recommendation with attentive multi-view learning. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. 3863–3869.

[48] Chuhan Wu, Fangzhao Wu, Suyu Ge, Tao Qi, Yongfeng Huang, and Xing Xie. 2019. Neural news recommendation with multi-head self-attention. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. 6389–6394.

[49] Chuhan Wu, Fangzhao Wu, Yongfeng Huang, and Xing Xie. 2021. User-as-graph: User modeling with heterogeneous graph pooling for news recommendation. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence*. 1624–1630.

[50] Chuhan Wu, Fangzhao Wu, Yongfeng Huang, and Xing Xie. 2023. Personalized news recommendation: Methods and challenges. *ACM Transactions on Information Systems* 41, 1 (2023), 1–50.

[51] Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. 2021. Empowering News Recommendation with Pre-trained Language Models. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1652–1656.

[52] Fangzhao Wu, Ying Qiao, Jiun-Hung Chen, Chuhan Wu, Tao Qi, Jianxun Lian, Danyang Liu, Xing Xie, Jianfeng Gao, Winnie Wu, et al. 2020. Mind: A large-scale dataset for news recommendation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 3597–3606.

[53] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. 2021. Self-supervised graph learning for recommendation. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*. 726–735.

[54] Lianghao Xia, Chao Huang, Yong Xu, Jiashu Zhao, Dawei Yin, and Jimmy Huang. 2022. Hypergraph contrastive collaborative filtering. In *Proceedings of the 45th International ACM SIGIR conference on research and development in information retrieval*. 70–79.

[55] Ruobing Xie, Qi Liu, Shukai Liu, Ziwei Zhang, Peng Cui, Bo Zhang, and Leyu Lin. 2021. Improving accuracy and diversity in matching of recommendation with diversified preference network. *IEEE Transactions on Big Data* 8, 4 (2021), 955–967.

[56] Boming Yang, Dairui Liu, Toyotaro Suzumura, Ruihai Dong, and Irene Li. 2023. Going Beyond Local: Global Graph-Enhanced Personalized News Recommendations. In *Proceedings of the 17th ACM Conference on Recommender Systems*. 24–34.

[57] Liangwei Yang, Shengjie Wang, Yunzhe Tao, Jiankai Sun, Xiaolong Liu, Philip S Yu, and Taiqing Wang. 2023. DGRec: Graph Neural Network for Recommendation with Diversified Embedding Generation. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*. 661–669.

[58] Renchi Yang, Jieming Shi, Xiaokui Xiao, Yin Yang, Juncheng Liu, and Sourav S Bhowmick. 2020. Scaling attributed network embedding to massive graphs. *Proceedings of the VLDB Endowment* 14, 1 (2020), 37–49.

[59] Yaming Yang, Ziyu Guan, Jianxin Li, Wei Zhao, Jiangtao Cui, and Quan Wang. 2021. Interpretable and efficient heterogeneous graph convolutional network. *IEEE Transactions on Knowledge and Data Engineering* (2021).

[60] Yang Yu, Fangzhao Wu, Chuhan Wu, Jingwei Yi, and Qi Liu. 2022. Tiny-NewsRec: Effective and Efficient PLM-based News Recommendation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. 5478–5489.

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

[62] Guangping Zhang, Dongsheng Li, Hansu Gu, Tun Lu, Li Shang, and Ning Gu. 2023. Simulating News Recommendation Ecosystem for Fun and Profit. *arXiv preprint arXiv:2305.14103* (2023).

[63] Jun Zhao, Zhou Zhou, Ziyu Guan, Wei Zhao, Wei Ning, Guang Qiu, and Xiaofei He. 2019. Intentgc: a scalable graph convolution framework fusing heterogeneous information for recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2347–2357.

[64] Qibin Zhao, Masashi Sugiyama, Longhao Yuan, and Andrzej Cichocki. 2019. Learning efficient tensor representations with ring-structured networks. In *ICASSP 2019-2019 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 8608–8612.

[65] Cai-Nicolas Ziegler, Sean M McNee, Joseph A Konstan, and Georg Lausen. 2005. Improving recommendation lists through topic diversification. In *Proceedings of the 14th International Conference on World Wide Web*. 22–32.

Just Accepted