IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 12 January 2024, accepted 30 January 2024, date of publication 2 February 2024, date of current version 12 February 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3361680

RESEARCH ARTICLE

Self-Supervised Hypergraph Learning for Enhanced Multimodal Representation

HONGJI SHU[1], CHAOJUN MENG[1], PASQUALE DE MEO[2], QING WANG[1], (Graduate Student Member, IEEE), AND JIA ZHU[3], (Member, IEEE)

[1]School of Computer Science and Technology, Zhejiang Normal University, Jinhua, Zhejiang 321004, China
[2]Department of Computer Science, University of Messina, 98122 Messina, Italy
[3]College of Education, Zhejiang Normal University, Jinhua, Zhejiang 321004, China

Corresponding author: Jia Zhu (jiazhu@zjnu.edu.cn)

This work was supported in part by the Natural Science Foundation of Zhejiang Province under Grant LY23F020010; in part by the Key Research and Development Program of Zhejiang Province under Grant 2022C03106; and in part by the Key Laboratory of Intelligent Education Technology and Application of Zhejiang Province, Zhejiang Normal University, Zhejiang, China.

ABSTRACT Hypergraph neural networks have gained substantial popularity in capturing complex correlations between data items in multimodal datasets. In this study, we propose a novel approach called the self-supervised hypergraph learning (SHL) framework that focuses on extracting hypergraph features to improve multimodal representation. Our method utilizes a dual embedding strategy and leverages SHL to improve the accuracy and robustness of the model. To achieve this, we employ a hypergraph learning framework to extract global context effectively by capturing rich inter-modal dependencies. Additionally, we introduce a novel self-supervised learning (SSL) component that utilizes the interaction graph data, thereby strengthening the robustness of the model. By jointly optimizing hypergraph feature extraction and SSL, SHL significantly improves the performance of multimodal representation tasks. To validate the effectiveness of our approach, we construct two comprehensive multimodal micro-video recommendation datasets using publicly available data (TikTok and MovieLens-10M). Prior to dataset creation, we meticulously handle invalid entries and outliers and complete missing mode information using external auxiliary sources, such as YouTube. These datasets are made publicly available to the research community for evaluation purposes. Experimental results on the above recommendation datasets demonstrate that the proposed SHL approach outperforms state-of-the-art baselines, highlighting its superior performance in multimodal representation tasks.

INDEX TERMS Multimodal, micro-video, self-supervised learning, hypergraph neural networks.

I. INTRODUCTION

Recommendation services have emerged as a fundamental element in various business domains, including, but not limited to, popular e-commerce platforms, such as Amazon and Taobao; social media platforms, such as Facebook and WeChat; and short video-sharing platforms, such as TikTok and Kwai.

The aforementioned platforms are *heterogeneous* in nature, that is, they are populated by *users* as well as a broad range of *entities* (also known as *items*) such as video clips, text snippets, audio tracks, and metadata describing available content or auxiliary information, such as ratings or likes. Therefore, users are allowed to interact in several ways, and it is crucial to accurately predict whether a target user likes or dislikes a specific entity.

Most previous studies use historical interactions between users and entities to generate predictions, and they focus on two key points: *presentation learning* and *interaction modeling* [1], [2]. In existing models, the quality of the representation directly affects the accuracy of the model's predictions.

An increasing number of researchers have applied graph neural networks (GNN) [3] to more effectively describe

The associate editor coordinating the review of this manuscript and approving it for publication was Jerry Chun-Wei Lin.

associations between users and items, and some studies have experimentally demonstrated the high predictive accuracy of GNNs [4], [5].

Moreover, inspired by the concept of GNN information propagation, researchers have adopted a similar approach to enhance the feature representation of users and entities.

For example, neural graph collaborative filtering (NGCF) [6] and light graph convolutional network (LightGCN) [7] convert collaborative filtering (CF) signals into feature representations using this propagation concept and achieve good results.

Standard GNN approaches to making recommendations have two main limitations.

First, existing approaches mainly consider *direct interactions* between a user and an entity; user-item interactions can be visually represented as a *bipartite graph* (see Figure 1) in which the first group of nodes corresponds to users and the second group of nodes corresponds to entities (In our example, entities coincide with movies.). In graph theory, we call these direct interactions *first-order connectivity*.

However, interactions between users and entities can be more complex and rich than direct interactions. A proper representation of these interactions can provide valuable insights into the association between users and entities. To gain a better understanding, let us consider Figure 1 again, the upper part: Here, we disclose a complex (but frequent) chain of interactions in which a user u_1 liked a movie m_1 viewed by a user u_2 who positively evaluated a movie m_2 and so on.

We colloquially call the *highway message* the chain of interactions above and observe that the existence of such a chain suggests that u_1 is likely to appreciate the movie m_3. Bipartite graphs are effective in modeling direct interactions, but they fail to capture highway messages in which nodes associated with users are arbitrarily connected with nodes corresponding to entities. Hereinafter, we call *higher-order connectivity* the interactions between users and entities that resemble the highway message depicted in Figure 1. Equivalently, first-order connectivity can be classified as a type of *local feature*. In contrast, higher-order connectivity relations can combine nodes that potentially reside in far regions of the user-entity graph. Thus, higher-order connectivity is a type of *global feature*.

Second, entities are often associated with a range of *modalities*, such as texts, images, audio, and videos. Thus, graph mapping interactions between users and entities are *multimodal*. Existing approaches [6], instead, consider only a single modality feature.

Recent studies [8], [9], [10], [11] have shown that feature modeling of multimodal information can capture fine-grained preferences between users and entities. However, the traditional graph structure (where only one edge exists between two nodes) significantly limits the representation of features between users and entities.

In addition, studies have [12], [13] shown that the modeling of complex relationships in hypergraphs is superior to that

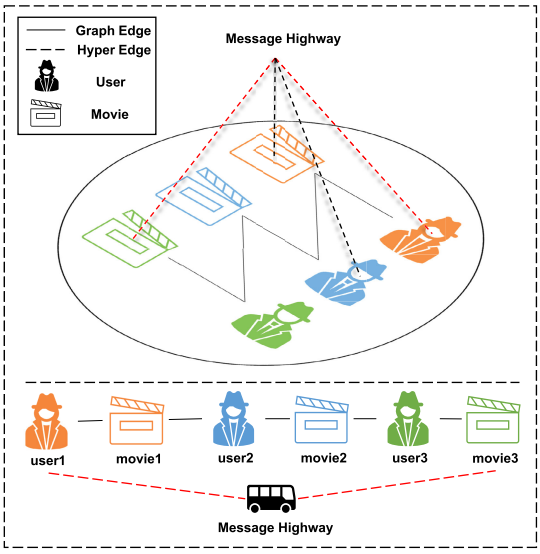

FIGURE 1. Hyperedge-based message highway. Herein, we describe a complex chain of interactions. The solid line represents the direct interaction between the user and the item, and the dashed line represents the hyperedge used for high-speed information transmission.

of traditional GNN networks. Hypergraphs [14] provide a flexible and natural modeling tool for modeling such complex relationships. In many real networks, such complex relationships are common, thus stimulating the problem of using hypergraphs for learning. Concurrently, studies using self-supervised learning (SSL) [15] have shown that it can effectively improve the robustness of models.

However, in the aforementioned studies, their focus is on examining the relationships between adjacent nodes in the graph and uncovering the profound connections between nodes. Nodes that are farther apart can rely only on multiple aggregations of GNNs to achieve this. However, owing to excessive smoothing and the presence of noise, a GNN with too many layers will affect the overall performance of the model. This study aims to use hypergraph-based learning structures to capture global feature information and enhance local feature information based on traditional GNNs. Additionally, it captures user preferences for different modes on a multimodal interaction graph. The result is a new architecture called *multimodal self-supervised hypergraph learning (SHL)*.

The SHL architecture replaces *edges* with *hyperedges*. Thus, we propose to map the user-entity interaction graph as a *hypergraph*. In general, a hyperedge connects an arbitrary number of nodes associated with users with an arbitrary number of nodes representing entities (see Figure 2 for a practical illustration). Hyperedges are an easy-to-understand tool for describing higher-order connectivity; thus, they extend the traditional methods that consider only first-order connectivity. However, the methods developed to train GNNs can be easily extended to manage hypergraphs, as shown in the subsequent sections.

In summary, the key contributions of this study are as follows.

- We propose a new method, SHL, that uses dynamic hypergraph structure information to capture the feature representation on the global scale between different modalities. Our SHL architecture incorporates an attention-based mechanism to represent local features. Global- and local-level features are then properly merged to obtain a more accurate representation of any entity.
- We propose an efficient technique to learn the structure of hypergraphs. In particular, we assume that the incidence matrix of a hypergraph (that is, the matrix containing, for each hyperedge, nodes such as hyperedge connects) can be factorized as the product of two low-rank matrices. Thus, we significantly reduce the number of parameters required to learn and avoid overfitting.
- We construct two complete multimodal micro-video datasets that can be reused by the research community for evaluation purposes. We started with two public datasets, TikTok and MovieLens-10M [16], and eliminated invalid data and outliers; we also completed missing mode information. We compared our architecture with that of five state-of-the-art competitors and conducted experiments.

II. RELATED WORK

A. COLLABORATIVE FILTERING-BASED METHOD

Previous studies on personalized recommendations have used the CF technique. Recently, studies have been conducted by combining neural networks and CF-based methods. In detail, matrix factorization (MF) methods [17] compute the inner product in the latent space of the user item to predict ratings, but this choice cannot capture complex interactions. To this end, He et al. [18] introduced the neural collaborative filtering (NCF) method, which replaces the inner product with nonlinear functions obtained using a multilayer perceptron to model interactions between the user and item features. Chen et al. [19] proposed an *attention mechanism* called attentive collaborative filtering (ACF) to properly capture implicit feedback in multimedia recommendation.

B. GRAPH NEURAL NETWORK-BASED METHOD

Methods such as NCF or ACF learn the embeddings of the user and item from descriptive functions such as IDs or attributes. In particular, information about the interaction between the user and item is only relevant to defining the loss function used to train the model. However, GNNs can fully use user-item interaction information; thus, they have been extensively exploited in the design of modern recommender systems. A major breakthrough is due to the NGCF approach described by Wang et al. [6], which refines the embedding of a user (respectively, an item) by aggregating the embeddings of the interacted users/items. This mechanism resembles the well-known message-passing procedure used in graph convolutional neural networks. LightGCN [7] is built on NGCF, but it significantly simplifies the architecture of NGCF by incorporating only a neighborhood aggregation module.

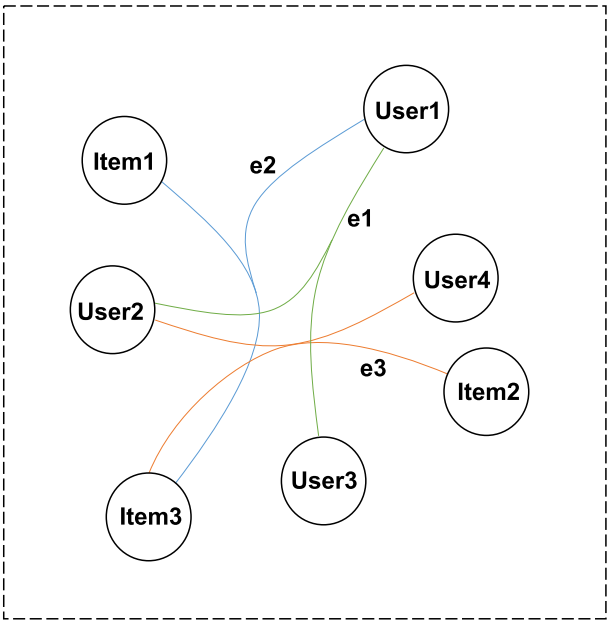

FIGURE 2. A hypergraph connecting users and items on a Social Web platform. Here, e_1, e_2, and e_3 are the hyperedges.

C. GRAPH CONVOLUTION NETWORK-BASED METHOD

In the field of multimedia recommender systems, Wei et al. [9] introduced the *multimodal graph convolutional network (MMGCN) system*. The MMGCN system constructs a bipartite user-item interaction graph for each modality. Then, it combines signals from the relevant contents (such as frames) of interactive objects for each modality (such as videos). The output of such an aggregation step is used to learn a better user representation.

D. GRAPH ATTENTION NETWORK-BASED METHOD

In GNN models, noise information can increase because of the continuous iteration of message transmission and aggregation [7]. In contrast, graph attention network (GAT) architectures can better control the spread of noise information in the message transmission process [10]. However, owing to the limitations of embedding vector smoothness in GAT architectures, it is difficult to spread node information over long distances. This means that two users with similar interests cannot correlate with each other when they correspond to nodes located in distant regions of the user-item interaction graph. Hypergraphs are elegant tools used to solve this problem [20]. As shown in Figure 2, a hyperedge can link any number of vertices in a hypergraph [21]; thus, it can connect nodes in distant regions of the user-item interaction graph.

E. CONTRASTIVE LEARNING-BASED METHOD

Moreover, numerous studies [22], [23], [24] have shown that SSL can significantly enhance the robustness of the model.

For example, self-supervised graph learning (SGL) [15] leverages edge and node dropout, along with the utilization of random walk techniques. Another example is multimodal graph contrastive learning (MMGCL) [25], where SSL techniques are integrated into multimodal learning, accompanied by a proposed negative sampling strategy that facilitates the augmentation of the targeted graph data.

In this study, we aim to advance the current state-of-the-art by introducing a novel approach that combines GNN architectures with hypergraphs in the context of multimodal data. Our primary objective is to enhance the robustness of the model by incorporating SSL methods. This unique combination allows us to harness the inherent noise reduction capabilities of GNNs while enabling the flexible aggregation of related nodes in arbitrary configurations. Using these synergistic techniques, we strive to achieve superior performance and address the challenges posed by complex and diverse data environments.

III. PROPOSED METHOD

In this section, we describe the overall framework of the multimodal SHL model.

As illustrated in Figure 3, the proposed model comprises five components: a *multimodal embedding* layer, a *local encoding* layer, a *global encoding* layer, a *multimodal feature fusion* layer, and a *prediction* layer. First, through the multimodal embedding layer, we convert a video into a visual vector, an audio vector, and a text vector as input for the model. Next, different modes of data flow through the local and global feature coding layers concurrently and finally converge at the fusion layer. The data in the same mode share the weight. Different colors in the diagram represent different modes. Finally, multimodal feature fusion and prediction of the interaction between each user and each entity are performed through the multimodal feature fusion layer and the prediction layer. In the following sections, we provide a detailed explanation of each component of the model.

A. PROBLEM STATEMENT

Let U, I, and M denote a set of users, entities, and modalities (with sizes $|U|$, $|I|$, and $|M|$, respectively). The goal of SHL is to generate new users U_{new} and entities I_{pos} so that a user has a high degree of matching with the appropriate entity while ensuring that the new user U_{new} and entity I_{neg} have a low degree of matching when they do not match. Here, pos and neg represent positive samples that users like and negative samples that users do not like, respectively.

B. MULTIMODAL EMBEDDING LAYER

In line with previous studies [18], [19], we assume that each user/entity is endowed with an ID; thus, we learn an embedding e_u (respectively, e_i) for each user $u \in U$ (respectively, entity $i \in I$) from such an ID. For each modality $m \in M$ and each user $u \in U$ (respectively, entity $i \in I$), we learn an embedding $e_{m,u}$ (respectively, $e_{m,u}$). For example, in the case of text, we apply the popular Doc2Vec algorithm [26]. We denote the number of features used to encode raw visual, acoustic, and text data as *Video*, *Audio*, and *Text*, respectively.

For each modality, we manage the graph recording interactions between users and entities (see Figure 3).

C. LOCAL ENCODING LAYER

Inspired by previous studies [27], [28], the GAT network extracts local graph features. To study the embedding of a node h, the local encoding layer combines three types of embeddings: the ID embedding of h, the embedding of h, and the embeddings of the neighbors of h. We then illustrate how the gate/attention blocks manage the embeddings of the neighbors of h.

For a fixed modality $m \in M$, we apply the following equation to learn the embedding of the neighbors N_h of h as follows:

$$e_{m,N_h} = R\left(\sum_{t \in N_h} f_a(h,t) f_g(h,t) \mathbf{W}_m^N e_{m,t}\right), \tag{1}$$

where (and hereinafter) R denotes the *LeakyReLU* function. Functions $f_a(h,t)$ and $f_g(h,t)$ are the gating and attention components, respectively. Finally, $\mathbf{W}_m^N$ denotes the trainable parameter matrix, and $e_{m,t}$ denotes the embedding of node t in modality m.

We implement the gating mechanism through an inner product, as follows:

$$f_g(h,t) = \delta\left(\frac{e_{m,h}^\top e_{m,t}}{\sqrt{d_t}}\right), \tag{2}$$

where $\delta(\cdot)$ denotes the *sigmoid* function, and d_t denotes the out-degree of the node t. We recall that the gating mechanism controls the information flow in the propagation process.

After obtaining the gating weights, we compute the attention weights as follows [29]:

$$f_a(h,t) = \left(\mathbf{W}_{m,h} e_{m,h}\right)^\top tanh(\mathbf{W}_{m,t} e_{m,t}), \tag{3}$$

where *tanh* function is utilized as a nonlinear activation function and m denotes different modal spaces. The matrices $\mathbf{W}m,h$ and $\mathbf{W}m,t$ are the learnable transformation matrices.

To determine the attention weights that represent the affinity between two nodes, we use the inner product and then normalize the attention weights across all neighbors using the softmax function [30]. This allows the final attention scores to distinguish the varying importance scores of the neighbors:

$$f_a(h,t) = \frac{exp^{f_a(h,t)}}{\sum_{t' \in N_h} exp^{f_a(h,t')}}. \tag{4}$$

Finally, we modify the feature representation of h using the embedding e_{m,N_h}. In particular, the ID embedding e_h of h is used as the anchor between modalities and consequently acts

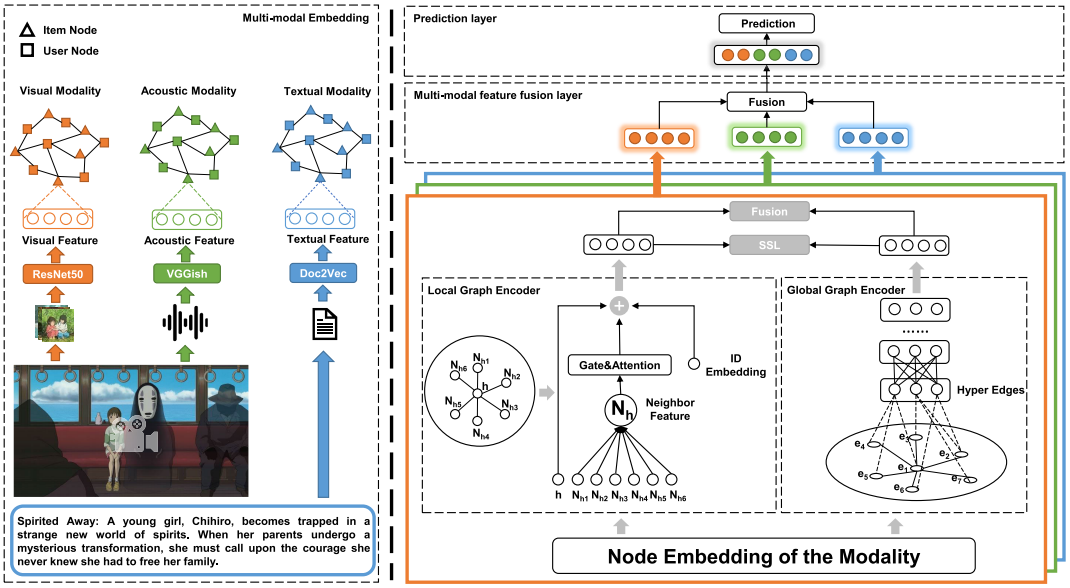

FIGURE 3. Architecture of our SHL model containing multimodal embedding layer, local encoding layer, global encoding layer, multimodal feature fusion layer and prediction layer.

as the propagation highway. The formula is as follows:

$$\tilde{e}_{m,h} = R(\mathbf{W}_m^h e_{m,h}) + e_h, \tag{5}$$

$$e_{m,local} = R(\mathbf{W}_m^l e_{m,N_h}) + \tilde{e}_{m,h}, \tag{6}$$

where $\mathbf{W}_m^h$ and $\mathbf{W}_m^l$ denote the trainable parameter matrices, respectively, and $e_{m,local}$ denotes the local embedding in the modality m.

D. GLOBAL ENCODING LAYER

Owing to the limitations of the local feature representation, it is challenging to transfer information between distant nodes. Therefore, we introduce the concept of *global features* to relate nodes in distant regions of the multimodal user-entity interaction graph and use hypergraph structure learning (HSL) to build the global feature representation of both users and entities.

1) HYPERGRAPH MESSAGE PASSING

According to Jiang et al. [20], messages passing across the nodes of a hypergraph are relevant to capture deeper higher-order relationships and thus enable us to construct global features.

A hypergraph comprises both nodes and hyperedges, and each hyperedge can link any number of nodes [31]. Each hyperedge establishes a direct information channel between the user and item. More formally, we define a hypergraph $\mathcal{G}_H$ in which N is the set of nodes and H is the set of hyperedges. We also assume that the size $|H|$ of the available hyperedges is a fixed hyperparameter in our model. We introduce two matrices, that is, $\mathbf{H} \in \mathbb{R}^{|N|\times|H|}$ (which is a trainable hypergraph dependency matrix) and $\mathbf{E} \in \mathbb{R}^{|N|\times d}$ (which is the matrix collecting node embeddings) under the assumption that each embedding has size d. Thus, we compute

$$e_{m,global} = R(\mathbf{H}\mathbf{H}^\top \mathbf{E}). \tag{7}$$

2) HYPERGRAPH STRUCTURE LEARNING

Because of the addition of the hypergraph dependency matrix $\mathbf{H}$, the entire model depends on numerous parameters. Consequently, the space required to store all parameters increases significantly, as does the complexity of the training phase. To maintain the computational complexity of our approach without sacrificing the expressiveness of our model, we decompose $\mathbf{H}$ as follows:

$$\mathbf{H} = \mathbf{E}\cdot\mathbf{W}, \tag{8}$$

where $\mathbf{E} \in \mathbb{R}^{|N|\times d}$ and $\mathbf{W} \in \mathbb{R}^{d\times|H|}$. Thus, we can reduce the number of parameters to $d\times|H|$, resulting in a significant decrease in the complexity of the training phase and allowing the model to run smoothly without sacrificing its expressiveness.

3) HYPERGRAPH STRUCTURE MAPPING

To complete our SHL system, we stack several hypergraph neural layers (HNLs) to increase the ability of the hypergraph to capture global features. The specific formula is as follows:

$$e_{m,global} = R(\mathbf{H} f_{hnl}(\mathbf{H}^\top \mathbf{E})), \tag{9}$$

$$f_{hnl}(X) = R(\mathbf{W}^{hnl} X) + X, X = \mathbf{H}^\top \mathbf{E}, \tag{10}$$

where $f_{hnl}()$ represents the hypergraph neural layers, matrices $\mathbf{H}$ and $\mathbf{E}$ are defined before, and $\mathbf{W}^{hnl}$ represents the trainable parameter matrix.

E. MULTIMODAL FEATURE FUSION LAYER

Based on previous studies [32], [33], [34], once we acquire local and global feature representations of a modality m, the next step involves fusing them to generate a composite representation denoted by e_m. This fusion process is achieved by applying the following equation:

$$e_m = R\left(e_{m,local} + e_{m,global}\right). \tag{11}$$

The last step involves fusing each representation e_m to obtain a representation e. We adopted two strategies: *average feature fusion* and *gate feature fusion*. We detail these two strategies below.

1) AVERAGE FEATURE FUSION

In the average feature fusion strategy, we compute the mean of representations e_m for each modality $m \in M$:

$$e = \frac{1}{|M|}\sum_{m\in M} e_m. \tag{12}$$

2) GATE FEATURE FUSION

In the gate feature fusion strategy, we first use the *Concat* function to connect the features of the various modalities:

$$e^* = Concat\{e_1||\cdots||e_m|\ \forall\, m \in M\}. \tag{13}$$

Next, we calculate and obtain the weight information:

$$\begin{aligned} \mathbf{W}'_m &= \frac{R(e_m^T)\odot R(\mathbf{W}_m^* e^*)}{\sqrt{d}}, \\ \mathbf{W}_m &= \frac{exp^{\mathbf{W}'_m}}{\sum_{m\in M} exp^{\mathbf{W}'_m}}, \end{aligned} \tag{14}$$

where $\odot$ represents the *torch.mul* function.

Finally, we use the weight information to obtain the final fusion feature data:

$$e = \sum_{m\in M} \mathbf{W}_m \odot e_m. \tag{15}$$

F. CONTRASTIVE LEARNING

After obtaining the local and global features, we use the global feature based on a hypergraph as the supervisory signal to reduce the influence of noise on the local feature. Formally, we take Information Noise Contrastive Estimation (InfoNCE) [35] and label it as L_{ssl}:

$$\begin{aligned} pos &= \exp\left(\frac{\sum_i (global_i \cdot local_i)}{0.1}\right), \\ neg &= \sum_i \exp\left(\frac{local_i \cdot global_i^T}{0.1}\right), \\ L_{ssl} &= \frac{1}{N}\sum_i \left(-\log\left(\frac{pos_i}{neg_i + \epsilon} + \epsilon\right)\right), \end{aligned} \tag{16}$$

where i represents the index of the nodes, ϵ represents a hyperparameter, and N represents the dimension of the nodes.

G. PREDICTION AND OPTIMIZATION

To predict the match scores y_{ui} between the user representation e_u and the entity representation e_i, we compute their inner product as follows:

$$y_{ui} = e_u^\top e_i. \tag{17}$$

Following Rendle et al. [36], we employ Bayesian Personalized Ranking (BPR), which assumes that users favor previously used items over unused ones to improve the model parameters. The loss function L to minimize is

$$L = \sum_{(u,i,j)\in O} -\ln\left(\delta(y_{ui} - y_{uj})\right) + L_{ssl} + \lambda\|\theta\|_2^2, \tag{18}$$

where $O = \{(u,i,j)|(u,i) \in R^+, (u,j) \in R^-\}$, R^+ represents the positive samples in the dataset and R^- represents the negative samples that do not exist in the dataset. Here, λ represents the regularization weight and θ represents the regularization parameters.

IV. EXPERIMENTS

To demonstrate the superiority of SHL and elucidate the reasons for its effectiveness, we conducted comprehensive experiments and addressed the following research questions.

- RQ1: How does SHL compare to state-of-the-art models in terms of top-k recommendation performance?
- RQ2: What benefits can the incorporation of hypergraph learning provide to the model?
- RQ3: What impact do various configurations have on the effectiveness of the proposed SHL?

A. EXPERIMENTAL SETTINGS

We preprocessed two publicly available micro-video datasets by performing data cleaning. As is customary, we randomly partitioned each dataset into training, validation, and test sets. To compare the performance of our SHL method with the baselines, we reimplemented several popular methods as baseline models and conducted experiments.

1) DATASETS PREPARATION

To evaluate the performance of SHL, we employed two publicly available datasets: TikTok and MovieLens. In particular, Table 1 presents some statistics on input datasets.

- **TikTok**[1]**:** This dataset was first made available in a TikTok data mining competition. The TikTok dataset comprises brief movies ranging from 3 to 15 seconds, coupled with a text description made public by the video creator. We cleaned the data before using it; in detail, we removed data items with missing modalities and outliers.
- **MovieLens**[2]**:** This dataset contains rating data for multiple movies provided by multiple users. Movie metadata and user attribute information are available.

[1] http://ai-lab-challenge.bytedance.com/tce/vc/
[2] https://grouplens.org/datasets/movielens/

TABLE 1. Dataset Introduction: Statistics for the TikTok and MovieLens datasets. The variables *Video*, *Audio*, and *Text* represent the number of features used for the raw visual, acoustic, and textual data, respectively.

Datasets	#Interactions	#Items	#Users	*Video*	*Audio*	*Text*
TikTok	724329	27375	37024	128	128	128
MovieLens	1442586	5162	51852	2048	128	100

TABLE 2. Performance evaluation of the baselines and SHL.

	TikTok			MovieLens		
Model	Precision	Recall	NDCG	Precision	Recall	NDCG
ACF	0.0913	0.2013	0.1065	0.0991	0.3905	0.2207
GraphSAGE	0.1074	0.2254	0.1187	0.1046	0.4115	0.2276
NGCF	0.1128	0.2276	0.1203	0.1063	0.4189	0.2356
MMGCN	0.1242	0.2523	0.1283	0.1122	0.4668	0.2648
MGAT	0.1566	0.3233	0.1672	0.1216	0.5099	0.2875
MMGCL	-	0.2695	0.1392	-	0.4926	0.2927
SHL	**0.1642**	**0.3352**	**0.1765**	**0.1267**	**0.5323**	**0.3085**
%Improv.	4.9%	3.7%	5.6%	4.2%	4.4%	5.4%

In this study, we used the MovieLens-10M dataset. For multimodal feature extraction, we first crawled the relevant video data from YouTube. After that, we extracted visual characteristics from video keyframes using the ResNet50 network [37]. The trained VGGish [38] network was applied to learn audio features. Finally, the doc2vec algorithm [26] is employed to learn text features, where the text contains the title and content.

2) EVALUATION METRICS

We randomly divided each dataset into three parts: training set (70% of input data), validation set (20% of input data), and test set (10% of input data). We used three metrics widely adopted in the recommender system literature: Precision@K, Recall@K, and Normalized Discounted Cumulative Gain (NDCG)@K with $K = 10$ [6], [39]. In the training process, we selected the learning rate from {1e-3, 1e-4, 1e-5}, adopted the warm-up learning strategy, trained for 1000 rounds, and finally reported the average performance obtained by all users in the test set.

3) BASELINES

We compared our approach with the following five baselines.

- ACF [19]. The ACF system leverages the attention mechanism to capture implicit interactions between items and users, thereby generating recommendations for multimedia content.
- GraphSAGE [40]. GraphSAGE computes node embeddings using node feature information on unseen nodes. It learns embeddings by sampling and aggregating the feature information of the local neighbors of a node using a feature function.
- NGCF [6]. The NGCF model implements a mechanism inspired by convolution in GNNs to explicitly model higher-order connectivity patterns.
- MMGCN [9]. The MMGCN builds separate bipartite graphs for each modality, connecting users to items in each modality, and then aggregates the representation information of these modalities to obtain the final user and item representation features for prediction.
- Multigraph attention and gating [10]. Multigraph attention and gating (MGAT) is an improvement of MMGCN and introduces attention and gating mechanisms to aggregate neighbor nodes to control the propagation of node information.
- MMGCL [25]. The MMGCL is an extension of SSL-based recommendation models that extend SGL. This method utilizes modal masking and edge loss techniques to achieve graph enhancement. However, the paper [25] presents some ambiguities, so the experimental evaluation of the model is based on a comprehensive comparison between the paper and our results.

B. PERFORMANCE COMPARISON (RQ1)

Table 2 presents the findings of the comparative analysis. The following conclusions can be drawn from this table.

- In every assessment measure, SHL surpasses all baseline models on the TikTok and MovieLens10K datasets. In particular, we report an improvement of up to 5.6% (respectively, 5.4%) between SHL and the second best-performing baseline in the TikTok (respectively, MovieLens) dataset. We attribute these improvements to the combination of features obtained by the local and global embedding layers. Thus, our model can capture the implicit relationships between nodes at a deeper level.
- The GNN-based model (that is, ACF) performs better than the traditional CF-based model. These advances are due to the graph convolution layer, which can effectively capture the relationships between nodes and effectively incorporate the features of the neighbors of a node to improve the representation learning stage.
- Our HSL makes up for the defects of the GNN-based and CF models. The HSL is suitable for determining the deep-level association information between nodes.

TABLE 3. Ablation study on key components of SHL.

Datasets	TikTok			MovieLens		
Variants	Precision	Recall	NDCG	Precision	Recall	NDCG
-HNL	0.1526	0.3160	0.1652	0.1242	0.5218	0.2983
-Hyper	0.1582	0.3266	0.1689	0.1229	0.5151	0.2905
-SSL	0.1640	0.3345	0.1759	0.1261	0.5310	0.3047
SHL	**0.1642**	**0.3352**	**0.1765**	**0.1267**	**0.5323**	**0.3085**

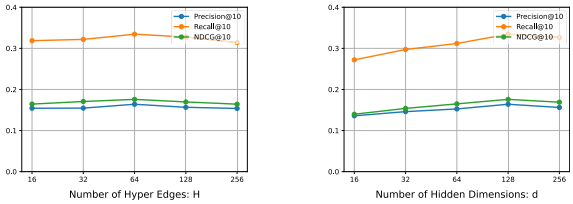

FIGURE 4. Hyperparameter research of the SHL.

C. ABLATION STUDY OF SHL (RQ2)

We investigate the impact of global hypergraph learning and hypergraph mapping on the performance of the proposed SHL model. Table 3 summarizes the evaluation results.

1) EFFECT OF GLOBAL HYPERGRAPH LEARNING

To investigate the overall effect of global hypergraph learning on the model, we conducted an experiment and evaluated its performance. We disabled this component from the model, termed -Hyper. Experiments show that once the hypergraph learning component was removed. The performance of the model decreased significantly. This implies that global collaboration can effectively capture the implicit features between nodes and refine the node feature representation through global message passing.

2) EFFECT OF HYPERGRAPH STRUCTURE MAPPING

We disabled the hypergraph structure mapping module in the model, termed -HNL. We found that the overall effect of the model shows a significant downward trend if we do not use deep hypergraph structure mapping. This shows that hypergraph structure mapping can help hypergraph learning capture more detailed feature representations.

3) EFFECT OF SELF-SUPERVISED LEARNING

After the SSL was removed, significant performance degradation occurred in all cases, confirming the positive effects of the enhanced global-to-local knowledge transfer.

D. HYPERPARAMETER COMPARISON (RQ3)

Here, we examine how some key hyperparameters affect the proposed SHL model. The results are shown in Figure 4. As can be observed, when there are 64 hyperedges and 128 hidden dimensions, the proposed model produces the best results. Hyperedges can establish a high-speed channel for information transmission between disconnected nodes in the proposed model. Increasing the number of hyperedges and hidden dimensions yields more complex semantic information. Concurrently, the performance drop may lie in overfitting because of an increase in the number of hyperedges and hidden dimensions.

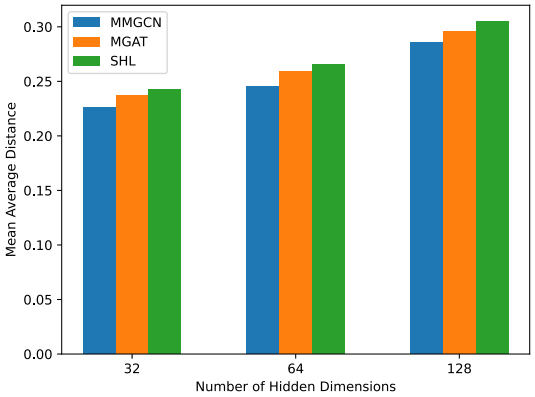

FIGURE 5. Comparison of the mean average distance (MAD) of the model with different hidden dimensions.

TABLE 4. Running time of each epoch.

Datasets	TikTok				
Number of Hyperedges	16	32	64	128	256
Running Time (sec.)	199	201	204	210	213

E. OVER-SMOOTHING EFFECT ANALYSIS

Over-smoothing is a critical problem that distinguishes the features of a complex of different nodes and affects the model performance. We used the mean average distance (MAD) as an evaluation index and conducted experiments with it. The results are shown in Figure 5. The results show that the proposed SHL model achieves the best results under different hidden dimensions.

F. RUNNING TIME ANALYSIS

We measured the time (in seconds) required to complete each epoch as a function of the number $|H|$ of hyperedges. Our tests were conducted on a Linux (Ubuntu 22.04) machine equipped with two Intel(R) Xeon(R) Gold 6254 CPUs @ 3.10GHz, four Nvidia GeForce RTX 3090 GPUs, and 256GB RAM. The results obtained are shown in Table 4. We observe that an increase in $|H|$ slightly affects the increase in running time, so our system is competitive even on large datasets.

V. CONCLUSION

This study introduces a multimodal recommendation method, SHL, based on hypergraph, which extends the existing

GNN technique. SHL introduces a global-based hypergraph dependency learning method that utilizes global high-order hidden features to compensate for the shortcomings of traditional GNNs in feature extraction. In particular, when learning user preferences, we use traditional GNNs to learn local features. Concurrently, we use the dynamic hypergraph learning method to obtain global features to enhance local feature information and compare the learned global features with local features to enhance the robustness of the model. Finally, we verify the validity of the model using two experiments with real datasets. However, some shortcomings are still present in this study, such as data noise. In subsequent works, we will focus on exploring and studying this problem further.

REFERENCES

[1] M. F. Aljunid and M. D. Huchaiah, "An efficient hybrid recommendation model based on collaborative filtering recommender systems," *CAAI Trans. Intell. Technol.*, vol. 6, no. 4, pp. 480–492, Dec. 2021.

[2] J. Xiao, H. Yang, K. Xie, J. Zhu, and J. Zhang, "Learning discriminative representation with global and fine-grained features for cross-view gait recognition," *CAAI Trans. Intell. Technol.*, vol. 7, no. 2, pp. 187–199, Jun. 2022.

[3] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 61–80, Jan. 2008.

[4] R. van den Berg, T. N. Kipf, and M. Welling, "Graph convolutional matrix completion," 2017, *arXiv:1706.02263*.

[5] L. Zheng, C.-T. Lu, F. Jiang, J. Zhang, and P. S. Yu, "Spectral collaborative filtering," in *Proc. 12th ACM Conf. Recommender Syst.*, 2018, pp. 311–319.

[6] X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua, "Neural graph collaborative filtering," in *Proc. 42nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2019, pp. 165–174.

[7] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "LightGCN: Simplifying and powering graph convolution network for recommendation," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2020, pp. 639–648.

[8] Q. Jin, J. Chen, S. Chen, Y. Xiong, and A. Hauptmann, "Describing videos using multi-modal fusion," in *Proc. 24th ACM Int. Conf. Multimedia*, 2016, pp. 1087–1091.

[9] Y. Wei, X. Wang, L. Nie, X. He, R. Hong, and T.-S. Chua, "MMGCN: Multi-modal graph convolution network for personalized recommendation of micro-video," in *Proc. 27th ACM Int. Conf. Multimedia*, Oct. 2019, pp. 1437–1445.

[10] Z. Tao, Y. Wei, X. Wang, X. He, X. Huang, and T.-S. Chua, "MGAT: Multimodal graph attention network for recommendation," *Inf. Process. Manage.*, vol. 57, no. 5, Sep. 2020, Art. no. 102277.

[11] C. Zhang and H. Li, "Low-rank constrained weighted discriminative regression for multi-view feature learning," *CAAI Trans. Intell. Technol.*, vol. 6, no. 4, pp. 471–479, Dec. 2021.

[12] S. Bai, F. Zhang, and P. H. S. Torr, "Hypergraph convolution and hypergraph attention," *Pattern Recognit.*, vol. 110, Feb. 2021, Art. no. 107637.

[13] Y. Gao, Z. Zhang, H. Lin, X. Zhao, S. Du, and C. Zou, "Hypergraph learning: Methods and practices," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 5, pp. 2548–2566, May 2022.

[14] S. Ji, Y. Feng, R. Ji, X. Zhao, W. Tang, and Y. Gao, "Dual channel hypergraph collaborative filtering," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2020, pp. 2020–2029.

[15] J. Wu, X. Wang, F. Feng, X. He, L. Chen, J. Lian, and X. Xie, "Self-supervised graph learning for recommendation," in *Proc. 44th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2021, pp. 726–735.

[16] F. M. Harper and J. A. Konstan, "The movieLens datasets: History and context," *ACM Trans. Interact. Intell. Syst.*, vol. 5, no. 4, pp. 1–19, 2015.

[17] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, Aug. 2009.

[18] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proc. 26th Int. Conf. World Wide Web*, 2017, pp. 173–182.

[19] J. Chen, H. Zhang, X. He, L. Nie, W. Liu, and T.-S. Chua, "Attentive collaborative filtering: Multimedia recommendation with item- and component-level attention," in *Proc. 40th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Aug. 2017, pp. 335–344.

[20] J. Jiang, Y. Wei, Y. Feng, J. Cao, and Y. Gao, "Dynamic hypergraph neural networks," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 2635–2641.

[21] Y. Feng, H. You, Z. Zhang, R. Ji, and Y. Gao, "Hypergraph neural networks," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 3558–3565.

[22] K. Zhou, H. Wang, W. X. Zhao, Y. Zhu, S. Wang, F. Zhang, Z. Wang, and J.-R. Wen, "S3-Rec: Self-supervised learning for sequential recommendation with mutual information maximization," in *Proc. 29th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2020, pp. 1893–1902.

[23] Z. Tao, X. Liu, Y. Xia, X. Wang, L. Yang, X. Huang, and T.-S. Chua, "Self-supervised learning for multimedia recommendation," *IEEE Trans. Multimedia*, vol. 25, pp. 5107–5116, 2022.

[24] J. Yu, H. Yin, X. Xia, T. Chen, J. Li, and Z. Huang, "Self-supervised learning for recommender systems: A survey," *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 1, pp. 335–355, Jan. 2024.

[25] Z. Yi, X. Wang, I. Ounis, and C. Macdonald, "Multi-modal graph contrastive learning for micro-video recommendation," in *Proc. 45th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2022, pp. 1807–1811.

[26] Q. V. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proc. 31st Int. Conf. Mach. Learn.*, vol. 32, Jan. 2014, pp. 1188–1196.

[27] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," 2014, *arXiv:1406.1078*.

[28] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel, "Gated graph sequence neural networks," 2015, *arXiv:1511.05493*.

[29] X. Wang, X. He, Y. Cao, M. Liu, and T.-S. Chua, "KGAT: Knowledge graph attention network for recommendation," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2019, pp. 950–958.

[30] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *Stat*, vol. 1050, no. 20, pp. 10–48550, 2017.

[31] X. Chen, K. Xiong, Y. Zhang, L. Xia, D. Yin, and J. X. Huang, "Neural feature-aware recommendation with signed hypergraph convolutional network," *ACM Trans. Inf. Syst.*, vol. 39, no. 1, pp. 1–22, Jan. 2021.

[32] D. Jin, Z. Qi, Y. Luo, and Y. Shan, "TransFusion: Multi-modal fusion for video tag inference via translation-based knowledge embedding," in *Proc. 29th ACM Int. Conf. Multimedia*, Oct. 2021, pp. 1093–1101.

[33] Y. Li, X. Zhang, F. Wang, B. Zhang, and F. Huang, "Fusing visual and textual content for knowledge graph embedding via dual-track model," *Appl. Soft Comput.*, vol. 128, Oct. 2022, Art. no. 109524.

[34] J. Zhu, C. Huang, and P. De Meo, "DFMKE: A dual fusion multi-modal knowledge graph embedding framework for entity alignment," *Inf. Fusion*, vol. 90, pp. 111–119, Feb. 2023.

[35] M. Gutmann and A. Hyvärinen, "Noise-contrastive estimation: A new estimation principle for unnormalized statistical models," in *Proc. 13th Int. Conf. Artif. Intell. Statist.*, in JMLR Workshop and Conference Proceedings, 2010, pp. 297–304.

[36] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: Bayesian personalized ranking from implicit feedback," 2012, *arXiv:1205.2618*.

[37] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[38] S. Hershey, S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. J. Weiss, and K. Wilson, "CNN architectures for large-scale audio classification," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2017, pp. 131–135.

[39] L. Xia, C. Huang, Y. Xu, P. Dai, X. Zhang, H. Yang, J. Pei, and L. Bo, "Knowledge-enhanced hierarchical graph transformer network for multi-behavior recommendation," in *Proc. AAAI Conf. Artif. Intell.*, 2021, vol. 35, no. 5, pp. 4486–4493.

[40] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1025–1035.

HONGJI SHU received the B.E. degree from Zhejiang Normal University, China, in 2018, where he is currently pursuing the degree with the College of Mathematics and Computer Science. His research interests include artificial intelligence and knowledge graph.

CHAOJUN MENG received the B.E. degree from Yanshan University, China, in 2021. He is currently pursuing the degree with the College of Computer Science, Zhejiang Normal University. His research interests include artificial intelligence and knowledge graph.

PASQUALE DE MEO received the Ph.D. degree in systems engineering and computer science from the University of Calabria. He has been a Marie Curie Fellow with Vrije Universiteit Amsterdam. He is currently an Associate Professor of computer science with the Department of Ancient and Modern Civilizations, University of Messina, Italy. His main research interests include social networks, recommender systems, and user profiling.

QING WANG (Graduate Student Member, IEEE) received the B.E. degree from Zhejiang Wanli University, China, in 2021. He is currently pursuing the degree with the College of Mathematics and Computer Science, Zhejiang Normal University. His research interests include data mining and artificial intelligence.

JIA ZHU (Member, IEEE) received the Ph.D. degree from The University of Queensland, Australia. He is currently a Distinguished Professor with the School of Teacher Education, Zhejiang Normal University, and the Deputy Director of the Key Laboratory of Intelligent Education Technology and Application of Zhejiang Province. His research interests include intelligent education, theoretical algorithms for database and data mining, federated learning, and blockchain with AI.