Bere: A Novel Video Recommender System for Virtual Reality Using Human Behavioral Signals

Huadi Zhu[*1,2], Chaowei Wang[1], Venkateshwar Reddy Darmanola[1], Hongbo Guo[3], Wenqiang Jin[4], Ming Li[1]

[1]The University of Texas at Arlington, [2]Boise State University, [3]Meta, [4]Hunan University

ABSTRACT

While video recommendation has been studied extensively in regular PC and smartphone settings, such a topic has been rarely discussed in the virtual reality (VR) context so far. On the other hand, as the popularity of VR videos continues to soar, its recommendation will play a crucial part in providing suggestions and guiding users through a deluge of available content. Given this unmet need, in this work, we present *Bere*, a video recommender system tailored for VR. Our approach leverages viewers' behavioral responses as they engage with VR videos to infer their preferences and thus make future recommendations. We integrate these new behavioral user-video interaction measures into the mainstream recommendation framework and renovate the graph learning-based paradigm to accommodate the new changes. The recommender system is further empowered with a novel domain adaptation approach named CMCCDA to address the data scarcity problem for model training. We also develop an energy-efficient adaptive encoding scheme to reduce the energy consumption on the VR device. We collect a behavioral dataset for video recommendation in VR and demonstrate through extensive evaluation that *Bere* significantly outperforms state-of-the-art schemes by up to 68.0% in precision and up to 28.8% in ranking quality.

CCS CONCEPTS

• **Human-centered computing** → **Ubiquitous and mobile computing systems and tools**; • **Computing methodologies** → **Modeling methodologies**.

[*]This work was completed when the author was a Ph.D. student at The University of Texas at Arlington.

ACM MobiCom '24, November 18–22, 2024, Washington D.C., DC, USA

ACM ISBN 979-8-4007-0489-5/24/11

https://doi.org/10.1145/3636534.3690660

1 INTRODUCTION

1.1 Background

In recent years, the integration of virtual reality (VR) technology has revolutionized individuals' encounters with digital content, casting a notable impact on video consumption. Watching videos in VR offers an immersive and 3-degrees-of-freedom interactive experience, allowing users to enter a three-dimensional virtual environment. As advancements in VR technology have made such experiences more accessible, a substantial growth in video consumption in VR can be expected in the near future.

As one of the core tasks in video services, video recommendation plays an essential role in providing accurate suggestions for watchers, enabling them to navigate through the overwhelming content and efficiently discover videos that truly capture their interest. With the continuously rising popularity of VR videos, the need for effective video recommendations in this context becomes increasingly crucial. However, to our knowledge, *there is no video recommender system specially tailored for VR users.* Currently, video recommendation schemes in VR are directly borrowed from existing frameworks adopted by traditional platforms such as YouTube; these frameworks are used for conventional computing terminals such as PCs and smartphones. In comparison, recommending videos in VR presents the following uniqueness and potential opportunities. First, unlike traditional 2D videos, VR videos bring to its viewers' unique perceptive feelings, such as cybersickness, immersiveness, and presence [19, 30, 31, 54, 64]. These unique attributes potentially introduce a new set of factors influencing viewer preferences in VR settings. Second, many VR headsets nowadays are equipped with a variety of onboard sensors, from IMU to eye trackers. They can capture novel types of user-video interactions, providing valuable insights that can be utilized to enhance video recommendation in VR contexts.

Recently, human behavioral signals has emerged as a new sensing modality to measure user preference during video watching [3, 8, 9, 13, 23, 29, 34, 38, 50, 60]. For example, Christoforou *et al.* [8] employed eye-tracking data to quantify the impact of narrative-based video stimuli to the preferences of large audiences. Lee *et al.* [34] studied the link between users' head movement data and their preference on

VR videos. This evidence motivates us to leverage users' behavioral responses when engaging with videos to infer their preferences and exploit such information to make future video recommendations. As an initial effort in this research topic, we start by examining two commonly accessible behavioral measures from VR headsets: eye gaze and head movement. Through an extensive measurement study, we validate that these two measures can serve as effective indicators of whether a user enjoys watching a video. Encouraged by this promising finding, we propose incorporating them existing frameworks to enhance recommendation performance.

1.2 Challenges

Despite the appeal of this concept, its implementation poses the following non-trivial challenges.

C_1: **Coping with new user-video interaction metrics.** First, with new kinds of user-video interaction metrics, a new data structure and a novel learning model are needed to effectively extract prominent features from the complex raw readings for video recommendation.

C_2: **Lack of training datasets.** Second, to train the recommender model properly, it is essential to acquire a sizable and diverse labeled behavioral dataset. This typically involves data from thousands of users and videos, encompassing up to a million interactions. As an initial effort to utilize behavioral signals to refine video recommendations, we face the challenge of a lack of existing annotated datasets. Consequently, assembling a comprehensive dataset of a meaningful magnitude and scope presents a significant hurdle.

C_3: **Energy consumption overhead.** Lastly, continuously uploading the new user-video interaction metrics to a server where the recommendation is performed is energy-consuming and may quickly deplete the battery of standalone VR headsets. Hence, how to achieve energy efficiency for VR terminals is another critical aspect to consider in *Bere*.

1.3 Our Solution: *Bere*

In this paper, we propose *Bere*, a novel video recommender system for VR enhanced by behavioral signals. It aims to exploit the correlation between behavioral measures and user-video preferences to enhance VR video recommendation. During a video session, the user's behavioral responses are collected by the VR device's built-in sensors and uploaded to the server to infer the user preference. A recommender system on the server takes these signals in all user-video interactions as the input, extracts their intrinsic and collaborative information, and makes recommendations accordingly.

To tackle the challenges (C_1-C_3), we make the following technical contributions. To address C_1, we propose to formulate users, videos, and their interactions as a graph, where behavioral signals are modeled as node and edge embeddings. Graph convolutional network (GCN) is applied over the graph for feature extraction. To accommodate the new property of the constructed graph, we renovate the conventional message passing function in the convolutional layers of the GCN, improving its capability to learn from behavioral signals and extract collaborative information.

To solve C_2, we adopt the concept of *domain adaptation*, which takes the traditional user-video interaction data as the source domain, the behavioral signals as the target domain, and adapts the model pre-trained on the source domain to the target domain via fine-tuning. Compared with training from scratch, only a small amount of data from the target domain is required. Considering the non-negligible gap between the two domains in our case, we propose a novel *cross-modality cross-context domain adaptation* (*CMCCDA*) scheme to fill the gap by introducing an extra "bridge domain". The adaption is then performed in two incremental steps.

Finally, we address C_3 by developing an energy-efficient adaptive encoding scheme. It adaptively encodes behavioral signals in accordance with their entropy to massively reduce data size and thus the energy overhead for data transmission.

We highlight our contributions of the paper as follows:

- We introduce *Bere*, a behavioral-signal-enhanced video recommender system for VR. To our knowledge, this is the first video recommender system tailored for VR utilizing behavioral signals.
- We integrate behavioral signals into the mainstream recommendation framework and renovate the GCN learning paradigm to accommodate the new property of the user-video interaction graph. A novel domain adaptation approach is developed to address the data scarcity problem. Additionally, an energy-efficient adaptive encoding scheme is proposed to reduce the energy consumption of VR devices.
- We collect a behavioral dataset for video recommendation in VR. It involves a total of 3,000 video sessions within 60 participants and 400 videos. This dataset will be open-sourced to the research community.
- We demonstrate through extensive evaluation that *Bere* outperforms state-of-the-art schemes for video recommendation in VR by up to 68.0% in recommendation precision and up to 28.8% in the ranking quality.

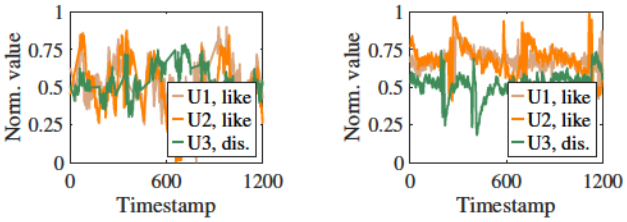
2 RELATED WORK

Taxonomy of recommender systems. Over the past few decades, recommender systems have become a crucial technique in diverse domains, including e-commerce, social media, and content streaming [10, 40, 44, 46, 59, 61]. These

systems are designed to predict and suggest items potentially liked by target users based on their historical preferences and behaviors. Based on how information is filtered for recommendations, these systems can be classified into three categories, namely *content-based filtering* [4, 39, 41], *collaborative filtering* [24, 33, 43, 45], and *hybrid filtering* [1, 5, 7, 51]. Among the above categories, collaborative filtering can capture complex and subtle patterns in user behavior and excels in its large-scale performance without requiring domain knowledge *a priori* [27]. As a sub-category of this direction, graph-based collaboration filtering techniques first structure user-item interactions as *graphs*, then employ graph learning models to extract collaborative information, which is further utilized to predict the preference of a target user on various items and make recommendations accordingly [15, 21, 56, 57, 59]. A representative state-of-the-art is PinSage [59], a recommendation framework for Pinterest utilizing GCN for personalized recommendations. Prior works [15, 49, 56, 57] also fall into this category. *Bere* also adopts the graph-based collaborative filtering framework.

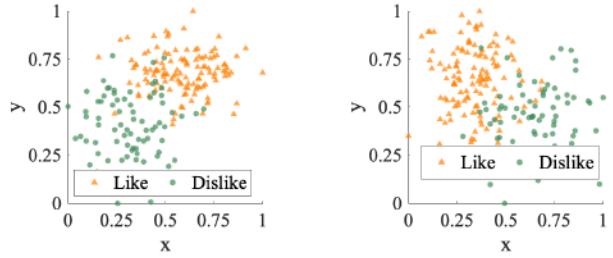
Video recommendation. Video recommendation is an important application of recommender systems. Related techniques have been widely employed by various video streaming platforms such as YouTube and TikTok [11, 16, 53]. Compared with the other tasks, video recommendation is unique due to its rich content and temporal dynamics [16, 21]. Extensive existing efforts have been devoted to tackling these characteristics [21, 26, 28, 37]. For example, Huang *et al.* [26] utilized video types and temporal factors to identify similar videos for recommendation. Jiang *et al.* [28] created fine-grained user interest groups based on users' interaction sequences and made recommendations based on the preferences of others from the same group. Recently, Han *et al.* [21] developed MTHGNN, a micro-video recommender system that considers the temporal and dynamic changes in users' preferences. None of the above works utilizes behavioral signals to understand user preference for recommendations.

Human-induced data and recommender systems. The idea of involving human-induced data in recommendation systems research has been explored in prior works. Over the past decade, researchers have attempted to link user preference/ratings with behavioral and physiological data, such as gaze [8, 23], head [34], brainwave [13, 38, 60], heart rate [50], and multi-modality data [3, 9]. However, these works focus on quantifying correlations between these measures and user preference and content rating, rather than forecasting user preferences over unseen videos from user-video interaction histories. Based on this, other researchers further explored the idea of involving human-induced measures in recommender systems [6, 12, 14, 36, 48, 58, 62], which only end up with predicting individual user preferences [6, 12, 14], inferring video genres [36], quantifying viewers' attention



(a) Gaze in x-axis (b) Head movement in yaw

Figure 1: Signal patterns of *likes* vs *dislikes*.



(a) Gaze (b) Head movement

Figure 2: Normalized distribution of 2D features of gaze and head movement.

[58], and predicting/clustering gazes [48, 62]. These efforts result in closed small-scale recommendation schemes that do not see practicality in the real-world settings. How to utilize such data to generate recommendation results has rarely been investigated in a systematic way. Training a comprehensive recommender systems generally requires a very large dataset in a range of thousands of users and millions of interactions. Acquiring such a dataset with human-induced data in lab settings is prohibitively infeasible. This lack of public dataset serves as the bottleneck of developing a comprehensive human-data-based recommender system. More importantly, none of the above systems is designed for VR settings, which present significant difference from traditional platforms and require delicate novel designs. In this work, we make an initial effort to bridge this gap.

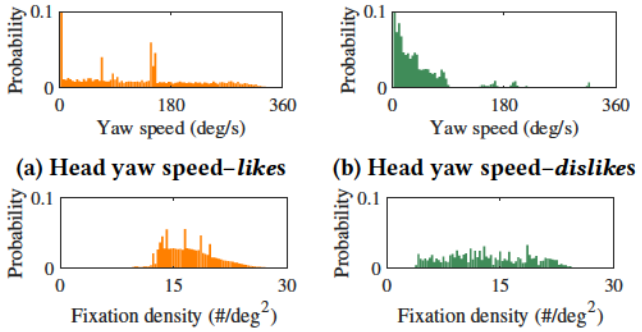
3 MEASUREMENT STUDY

Measurement setup. To validate that the correlation under the VR setting, we carry out an IRB-approved measurement study at a university lab. 10 subjects are recruited. Each is asked to watch a set of 20 videos wearing an HTC Vive Focus 3 VR device. After watching each video, subjects indicate whether they *like* or *dislike* it. During the entire process, their behavioral signals are recorded by the onboard sensors and locally stored on the VR device. The analysis is performed over the collected dataset from 200 video-watching traces. We focus on two kinds of behavioral signals: gaze and head movement, which are captured by the onboard eye tracker and inertial measurement unit (IMU), respectively.

Results. Figure 1 illustrates exemplary gaze (x-axis) and head movement (yaw) signal patterns of three users, the first two (U1, U2) liking the video whereas the third user (U3) dislikes it. We observe that U1 and U2, who have the same preference (*like*) to the video, display similar signal patterns to each other, while there are distinctive patterns between users with different preferences (*like* vs. *dislike*). To further illustrate this, we visualize two-dimensional latent features extracted from the behavioral signals in Figure 2, by applying an autoencoder to the raw readings and casting the derived multi-dimensional embedding vectors onto two dimensions for visualization. Note that the original multi-dimensional features exhibit even greater distinctiveness across different classes compared to these reduced two-dimensional features. Nevertheless, the features marked with *like*s and those with *dislike*s are still distributed in two distinctive clusters. Take gaze as an example: The mean values of its two-dimensional features are [0.63, 0.57] and [0.35, 0.34] (in [x, y]) for *like* and *dislike*, respectively. Their standard deviations are [0.18, 0.12] and [0.16, 0.22], respectively. Note that the original multi-dimensional features exhibit even greater distinctiveness across different classes compared to the reduced two-dimensional features shown in Figure 2.

We further extract two calibrated features, namely head yaw speed and fixation density. Head yaw speed, drawn from the IMU readings, denotes the average speed of a subject's head movement on the yaw axis. As shown in Figure 3a and 3b, values of this feature are more evenly distributed when subjects like the video, whereas those are more concentrated on the lower end otherwise. Fixation density is the average number of gazes in a unit area for all fixations. We observe from Figure 3c and 3d that fixation density approximately ranges between 12 and 24 gazes per unit area when the user likes the video; this value is more scattered otherwise.

The results are promising: Gaze and head movement exhibit patterns highly correlated with user interest in video content. They serve as evidence that such behavioral signals can be used as effective indicators of user preference, which will be exploited for video recommendation in this work.



(a) Head yaw speed–*like*s **(b) Head yaw speed–*dislike*s**

(c) Fixation density–*like*s **(d) Fixation density–*dislike*s**

Figure 3: Exemplary features distributions extracted from gaze and head rotation measures.

4 SYSTEM OVERVIEW

In this work, we propose *Bere*, a novel video recommender system for VR by exploring viewers' behavioral signals. *Bere* harnesses the intrinsic correlation between viewers' preferences and behavioral signals to enhance video recommendation. Figure 4 depicts the overall system architecture, which consists of four major components: adaptive encoding, graph construction, GCN-based recommendation, and CMCCDA. As a user watches a video, her behavioral signals, i.e., gaze and head movement, are recorded and encoded by the VR device. The encoded embeddings are uploaded to the cloud server, where the user-video interactions are constructed into graphs (§5.1). Then, a GCN model is employed to learn representations from the graph, based on which the top-K videos are derived and recommended to the target user (§5.2). To train the model with the limited annotated behavioral measures, we propose a novel domain adaption strategy CMCCDA to deal with the non-negligible inter-domain distances (§6). An adaptive encoding algorithm is also developed to compress the raw behavioral signals and reduce energy overhead for data communications at VR terminals (§7).

5 GRAPH-BASED RECOMMENDATION

Our task is to recommend a list of videos from a given video pool to a target user. The list consists of videos the user has not encountered and will likely align with her preference. Motivated by key observations from the measurement study, we propose to introduce behavioral signals (i.e., gaze and head movement) as a new kind of user-video interaction metric to facilitate VR video recommendations. In the following, we first model user-video interactions into graphs. Then, we employ GCN as a graph learning tool, upon which a recommender system is built.

5.1 Graph Construction

We construct the entire dataset of all users, videos, and their interactions as a graph $\mathcal{G} = \{\mathcal{U}, \mathcal{V}, C\}$, where the set of users are represented as graph nodes $\mathcal{U}$, the set of videos as nodes $\mathcal{V}$, and their connections as graph edges C. An edge $c_{ij} \in C$ connects a user node $u_i \in \mathcal{U}$ and a video node $v_j \in \mathcal{V}$; $c_{ij} = 1$ if u_i has watched v_j, and $c_{ij} = 0$ otherwise. We define the attribute of each edge as the *embedding*. To derive the embedding, an encoder is applied to the time-series behavioral signals recorded during the video-watching session; the encoder's design is detailed in §7. This embedding is a vector of features extracted to describe the user preference from the video watching session. Our definitions of edge

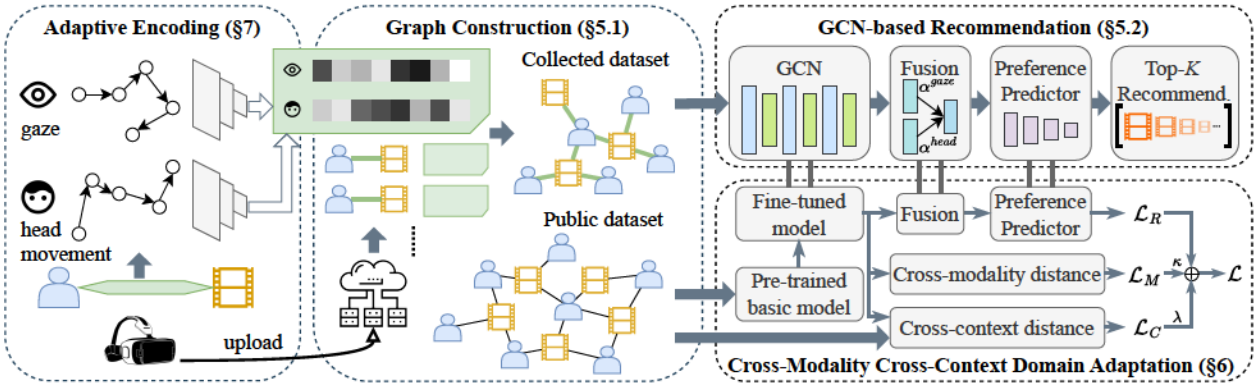


Figure 4: System architecture of *Bere*.

attributes are intuitive: Behavioral signals can reflect user-video interactions, as demonstrated in §3. We further define the node embedding $\boldsymbol{n}_i$ by taking the average of embeddings of all edges connected to that node: $\boldsymbol{n}_i = \frac{1}{|\mathcal{N}_i|}\sum_j \boldsymbol{e}_{ij}$, where $\mathcal{N}_i$ represents u_i's neighbor set and $\boldsymbol{e}_{ij}$ is the embedding of the edge between u_i and its neighbor v_j. The node's attribute is defined in such a way as behavioral signals contain rich information regarding both the user and her watched video. For example, the signal may reveal the user preferences and watching habits, which can be used to profile the user.

With the constructed user-video interaction graph, we further derive a *node attribute array*, an *edge attribute array*, and an *adjacency matrix*, which will be used in the graph learning presented soon. A node attribute array $X \in \mathbb{R}^{N\times D}$ represents all node embeddings $X = \{\boldsymbol{n}_i | \forall i \in [1, N]\}$. An edge attribute array $E \in \mathbb{R}^{N\times N\times D}$ represents all edge embeddings $E = \{\boldsymbol{e}_{ij} | \forall i, j \in [1, N]\}$. An adjacency matrix $A \in \mathbb{R}^{N\times N}$ represents the connectivity between two arbitrary nodes $A = \{c_{ij} \in \{0, 1\} | \forall i, j \in [1, N]\}$. In the above definitions, N denotes the number of nodes in $\mathcal{G}$ and D is the cardinality of the embedding vector.

Discussions. Most existing video recommendation frameworks only consider traditional user-video interactions, such as video-watching duration or if a user *likes* that video. Due to their simple data format (i.e., binary or real values), these edge weights are conveniently formulated into the adjacency matrix to feed into the graph learning models. In contrast, (embeddings of) behavioral measures here are high-dimension vectors, which cannot be represented as simple-value edge weights. Therefore, we incorporate them into the graph as edge embeddings E. These edge embeddings provide richer information than simple-value edge weights and thus offer more valuable insights into users' preferences. The main job of the rest of this work is to develop suitable learning techniques to extract latent features from the new graph with a sophisticated structure.

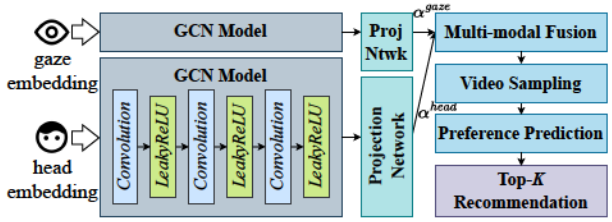


Figure 5: GCN-based video recommendation workflow.

5.2 GCN-based Video Recommendation

5.2.1 Overview. *Bere* is built on a classic graph-based recommendation framework, which consists of two stages: GCN learning and recommendation. *In the stage of GCN learning*, to mine the complex relationships among users and videos from the constructed graph, we apply a GCN model, which takes as input the graph, i.e., X, A, E as derived above, and produces the output as *node representations*. The representations in different modalities are then projected into the same space and fused. *In the stage of recommendation*, a subset of candidate videos is first sampled from the entire video pool. These are videos that the target user has not previously watched but may be interested in. Then, a preference predictor takes the target user's and each candidate video's *node representations* as an input pair and predicts the preference score. Finally, top-K videos with the highest scores are recommended to the user. Figure 5 illustrates this workflow.

To fit into our scenario, we make several renovations to the classic GCN-based recommendation framework, including modifying the message passing mechanism and some key calculations. Next, we will introduce each step of our GCN-based recommender system in detail.

5.2.2 GCN Learning. The first step of GCN applies convolution over the graph. It consists of the message passing and the aggregation steps – in these steps, the embeddings of each node's neighbors are propagated through connecting edges and integrated with its own embedding. As there are

two different modalities, i.e., gaze and head movement data, we start by considering an arbitrary modality $m \in \mathcal{M}$.

Message passing. As the first step of the convolution operation, message passing propagates the information from each node to all neighbors through their connecting edges

$$\boldsymbol{\mu}_{j \leftarrow i}^{m} = \frac{1}{\sqrt{|\mathcal{N}_i|\,|\mathcal{N}_j|}} \left(W_1^m \boldsymbol{n}_i^m + W_2^m \left(\beta \boldsymbol{e}_{i,j}^m + \boldsymbol{n}_i^m \odot \boldsymbol{n}_j^m \right) \right) \quad (1)$$

where $\boldsymbol{n}_i^m$ represents node u_i's embeddings of modality m, $\odot$ stands for the element-wise production, W_1^m and W_2^m are learnable matrices, and β is the weight. The conventional function allows each node to collect information from its immediate neighbors (the first term) [35, 57], thereby integrating local neighborhood information into its representation.

Discussions. For effective feature extraction, we renovate the message-passing function to accommodate our unique graph structure, where behavioral responses, serving as edge embeddings, contain information from both end nodes (user and video). To preserve such information in every convolutional layer, we integrate them into our message passing function as the second term. In this way, edge information is passed into target nodes, which fits our scenario based on the intuitive fact that user-video interactions contain users'/videos' characteristics. The added term gathers such information and improves learning efficiency. In addition, we introduce the third term above to encourage passing more information between similar neighbors. For example, if user u_i has a similar embedding as video v_j, it indicates that the characteristics of video v_j align well with u_i's preferences. This similarity can be leveraged to enhance v_j's feature representation by integrating more common information shared with u_i. Note that the introduction of this term has been explored in prior work [56], and we adapt a similar concept to improve the classic approach and enhance our recommendation performance. Nevertheless, our primary contribution lies in the innovative integration of the second term, which intuitively addresses our unique problem and significantly enhances system efficiency. Overall, our proposed design substantially changes the message-passing workflow in traditional GCN, a design unexplored in prior works.

Aggregation. Upon receiving all neighbor information, node v_j performs the following aggregation function on each layer l to derive the *collaborative information*

$$\boldsymbol{n}_j^{(l)m} = LeakyReLU\left(W_3^m \boldsymbol{n}_j^{(l-1)m} + \sum_{i \in \mathcal{N}_j} \boldsymbol{\mu}_{j \leftarrow i}^{(l)m} \right) \quad (2)$$

where $\boldsymbol{n}_j^{(l-1)m}$ denotes the target node embedding in the previous convolutional layer. On the lth convolutional layer, node v_j is updated by its information on the $(l-1)$th layer and the lth-layer neighbors' embeddings, which reflect the collaborative information.

Multi-modality attentional fusion. After several convolutional layers of message passing and aggregation, the output embeddings of multiple modalities are fused to derive the final embedding of each node. Considering the heterogeneous embedding space of each modality, we employ a projection network that maps embeddings of both modalities into the common space, before these projection outputs are weighted by their modality-specific attention and fused

$$\boldsymbol{n}_j = \sum_{m \in \mathcal{M}} \alpha^m H\left(\boldsymbol{n}_j^m \right) \quad \textbf{where} \quad \alpha^m = \frac{e^{-r^m}}{\sum_{i \in \mathcal{M}} e^{-r^i}} \quad (3)$$

where $\boldsymbol{n}_j$ stands for the final node representation after fusion, $H(\cdot)$ represents the projection network that maps all modalities to the same latent space, α^m denotes the attention for modality m, and r^m is the data compression ratio in m.

5.2.3 Recommendation. After GCN learning, the final representations at all nodes possess sufficient information for video recommendations. A video sampler first samples a list of candidate videos from the video pool for the target user [21], from which a preference predictor estimates their preference scores, and recommends the candidate videos with the highest preference scores to the user.

The preference predictor consists of a few fully connected layers; given a target user u and a candidate video v, it takes the final representations of u and v as an input pair and produces the predicted preference score

$$\hat{y}_{uv} = F_\theta \left(\gamma \boldsymbol{n}_i \odot \boldsymbol{n}_j + \frac{1}{|\mathcal{N}_v|} \sum_{\boldsymbol{n}_i \in \mathcal{N}_v} \boldsymbol{n}_u \odot \boldsymbol{n}_i + \frac{1}{|\mathcal{N}_u|} \sum_{\boldsymbol{n}_j \in \mathcal{N}_u} \boldsymbol{n}_v \odot \boldsymbol{n}_j \right) \quad (4)$$

where $F_\theta(\cdot)$ is a multilayer perceptron (MLP) parametrized by θ, and γ is a weighting hyperparameter. In this way, we comprehensively formulate u's potential preference towards v by capturing 1) their direct similarity, 2) the similarity between u and users who have watched v, and 3) the similarity between v and videos watched by u, before feeding it to the MLP that maps the input vector to the final preference score.

Discussions. Compared with the preference prediction function in existing works, we propose to add the correlation between the users' and the videos' embeddings as the first term, leveraging the fact that they both fall into the same feature space. In contrast, they commonly reside in heterogeneous embedding spaces in previous works. The additional term directly encourages recommending videos to the target user that share similar embeddings with each other.

Finally, videos in the candidate list are ranked based on their predicted preference scores, and the top-K candidate videos are recommended to the target user.

5.2.4 Loss Function. The remaining piece is to decide the loss function for model training. To this end, we establish upon the classic Bayesian Personalized Ranking (BPR) loss [42], a commonly adopted loss function to train recommender systems, and propose our recommendation loss as follows

$$\mathcal{L}_R = -|y_{uv^p} - y_{uv^q}| \log \sigma\left(|\hat{y}_{uv^p} - \hat{y}_{uv^q}|\right) \tag{5}$$

where $\hat{y}_{uv^p}$ and $\hat{y}_{uv^q}$ are the predicted preference scores between the target user u and a random pair of two arbitrary videos v^p and v^q, respectively, with y_{uv^p} and y_{uv^q} representing their ground-truth preference scores.

6 CROSS-MODALITY CROSS-CONTEXT DOMAIN ADAPTATION

To build the GCN-based recommendation model, it is crucial to gather a large labeled behavioral dataset of the necessary diversity and volume for proper model training. However, this is prohibitively infeasible as it involves thousands of users/videos and up to a million interactions. As a reference, *MovieLens-1M*, a widely used public dataset for training video recommender systems, consists of 1 million interactions from over 6 thousand users and over 3 thousand videos [18].

To address this data scarcity issue, we propose to adopt the concept of *domain adaptation*. It allows a deep learning model trained in one *source domain* (i.e., traditional user-video interaction data of 2D videos, denoted by $\mathcal{D}_S$) to adapt to a different but related *target domain* (i.e., viewers' behavioral signals in watching VR videos, denoted by $\mathcal{D}_T$) via fine-tuning. Typically, domain adaption needs a much smaller amount of data than training the whole model in the target domain from scratch. Nonetheless, the successful employment of domain adaptation requires the *distance* between the source and target domains to be within a certain threshold; otherwise, the performance will be degraded significantly. In our case, such distance is non-negligible as traditional user-video interaction (e.g., whether users finish watching videos, hit *like*s, etc.) and behavioral signals are two distinctive modalities. Additionally, they are across different contexts as $\mathcal{D}_S$ is for videos displayed on regular terminals, such as PCs and smartphones, whereas $\mathcal{D}_T$ is for VR videos. This brings significant discrepancy between the two domains due to the unique characteristics of VR, such as immersiveness and interactability, compared to regular terminals, as well as different spectra of resolution (360p - 4k on regular terminals vs. 4K - 8K in VR. *This discrepancy prohibits the direct adoption of traditional domain adaptation techniques.*

Yet, despite the heterogeneous modalities and contexts of user-video interactions from the two domains, they are both effective indicators of user preferences and can both serve the recommendation task. To reveal such hidden information, we harness them under a unified framework by projecting them into the same latent space before adaptation. Next, we will elaborate on our novel domain adaptation framework for efficient model training.

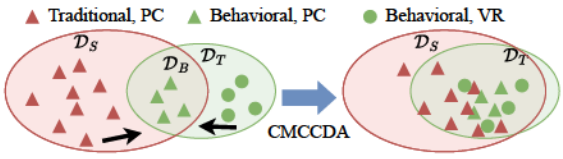


Figure 6: The illustration of the CMCCDA process; each point is a sample representation.

Bridge domain. To address the above challenge, we propose a novel domain adaptation framework called *cross-modality cross-context domain adaptation* (*CMCCDA*). The idea is to introduce a *bridge domain* $\mathcal{D}_B$, which connects the source and target domains. Rather than directly adapting the original graph learning model (trained over $\mathcal{D}_S$) to the target domain (using $\mathcal{D}_T$), we propose to fine-tune the model by minimizing the representations' distance between $\mathcal{D}_S$ and $\mathcal{D}_B$ and subsequently their distance between $\mathcal{D}_B$ and $\mathcal{D}_T$. In doing so, the substantial distance between $\mathcal{D}_S$ and $\mathcal{D}_T$ is broken down into two manageable distances that can be bridged in two incremental steps.

To perform CMCCDA, we create our own *hybrid* dataset: Each subject watches some randomly selected videos in VR headsets. During each video session, they freely hit the *like* or the *share* button, pause, fast-forward, rewind, or skip the current video play as they like. In the meantime, the VR headset records the subject's behavioral signals throughout the session. We call it a hybrid dataset because it involves both the traditional user-video interactions and the behavioral signals, which are aligned in each sample. The former shares the same modality with $\mathcal{D}_S$, while the latter shares the same context with $\mathcal{D}_T$. We denote this common information in the hybrid dataset, i.e., traditional interactions in the VR context, as $\mathcal{D}_B$, to facilitate bridging the two distinctive domains.

Cross-modality distance. We define the cross-modality distance as the distance between $\mathcal{D}_B$ and $\mathcal{D}_T$. It is computed as the average distance between the representation of each node in $\mathcal{D}_B$ and that of the corresponding node (i.e., same user/video) in $\mathcal{D}_T$. We call it "cross-modality" because nodes in $\mathcal{D}_B$ and those in $\mathcal{D}_T$ are associated with two modalities, the traditional interactions and behavioral signals, respectively. The distance between two corresponding nodes is denoted as $\Delta(z_i^B, H'(z_i^T))$, where z^B and z^T are the node representations, $\Delta(\cdot,\cdot)$ refers to any appropriate distance function, e.g., Euclidean distance, and $H'(\cdot)$ is a projection function to cast representations across modalities. This allows us to extract the common hidden information from the two modalities of interactions related to user preferences.

Cross-context distance. We define the cross-context distance as the distance between $\mathcal{D}_S$ and $\mathcal{D}_B$. Like the cross-modality distance, the cross-context distance involves the distance between representations of nodes from $\mathcal{D}_S$ and those from $\mathcal{D}_B$. Additionally, it also considers the difference in the graph structures of the two domains.

We first identify *common nodes* from $\mathcal{D}_S$ and $\mathcal{D}_B$. They are common videos in both the public dataset and the hybrid dataset. For each common node v_i, we identify its local graph covering its neighbor nodes in h hops and all edges involved, where h is an empirical value. Then, we calculate the distance between the common nodes' representations in two domains and weight it with the similarity between their local graphs $\Delta\left(z_i^S, z_i^B\right) \cdot \Gamma_i^{S,B}$, where $\Gamma_i^{S,B}$ is the similarity between the local graphs in $\mathcal{D}_S$ and $\mathcal{D}_B$, respectively.

To derive $\Delta\left(z_i^S, z_i^B\right) \cdot \Gamma_i^{S,B}$ efficiently, we propose to approximate this similarity by comparing their *graph representations*, which is defined as the weighted average of its node representations. To further enhance efficiency, given a common node v_o, we propose to approximate the learnable weight for each u_i (v_i) in its local graph using its *centrality* $\zeta_i = \frac{d_o d_i}{l_{o,i}^k} + \eta \sum_{j \in \mathcal{N}_i, j \neq o} \frac{d_i d_j}{l_{i,j}^k}$ where d represents the node degree and ι gives the length of the shortest path between two different nodes; η and k are hyperparameters. This centrality reflects how much the node contributes to the common node. The graph similarity is thus computed as follows.

$$\Gamma_i^{S,B} = C\left(\sum_{p \in \mathcal{G}_{i,h}^S} \zeta_p \boldsymbol{x}_p, \sum_{q \in \mathcal{G}_{i,h}^B} \zeta_q \boldsymbol{x}_q\right) \tag{6}$$

where $C(\cdot,\cdot)$ is an arbitrary similarity function, e.g., cosine similarity. $\mathcal{G}_{i,h}^S$ and $\mathcal{G}_{i,h}^B$ denote the local graphs of common node v_i in $\mathcal{D}_S$ and $\mathcal{D}_B$, respectively. $\boldsymbol{x}_p$ and $\boldsymbol{x}_q$ are the node embeddings before GCN. In doing so, we are able to eliminate the distraction of contexts and emphasize on preference-related features between two domains.

Bridging two domains. Finally, we formulate the loss functions based on the two distances discussed above

$$\mathcal{L}_M = \frac{1}{|\mathcal{S}|} \sum_{p \in \mathcal{S}} \Delta\left(z_p^B, H'\left(z_p^T\right)\right) \tag{7}$$

$$\mathcal{L}_C = \frac{1}{|\mathcal{O}|} \sum_{i \in \mathcal{O}} \Delta\left(z_i^S, z_i^B\right) \cdot \Gamma_i^{S,B} \tag{8}$$

where $\mathcal{S}$ is a set of randomly sampled nodes and $\mathcal{O}$ is the set of all common nodes. We derive two losses, $\mathcal{L}_M$ and $\mathcal{L}_C$, based on the cross-modality and cross-context distances, respectively. During CMCCDA, both losses are computed incrementally in each iteration and back-propagated to update the model parameters through optimization. Minimizing the former encourages extracting representations with a smaller distance between $\mathcal{D}_B$ and $\mathcal{D}_T$, adapting the model from the bridge domain to the target domain. Similarly, minimizing the latter rewards learning similar representations between $\mathcal{D}_S$ and $\mathcal{D}_B$, with an emphasis on common nodes that have more similar local graphs. These losses jointly guide the model to decrease the distance of representations in all domains through fine-tuning, adapting it from $\mathcal{D}_S$ to $\mathcal{D}_T$.

Given such, the final loss for fine-tuning is derived as

$$\mathcal{L} = \mathcal{L}_R + \kappa \mathcal{L}_M + \lambda \mathcal{L}_C \tag{9}$$

where κ and λ are tunable weights. Recall that $\mathcal{L}_R$ stands for our recommendation loss proposed in §5. $\mathcal{L}_M$ and $\mathcal{L}_C$ are defined above in Equation 7 and 8, respectively.

We summarize the steps of CMCCDA as follows. First, data from all three domains are sampled and fed into three identical, weight-sharing GCN models, respectively. Then, the outputs are utilized to compute the *cross-modality distance* between $\mathcal{D}_B$ and $\mathcal{D}_T$, and the *cross-domain distance* between $\mathcal{D}_S$ and $\mathcal{D}_B$. Finally, losses are derived based on these distances to fine-tune the GCN models.

7 ENERGY-EFFICIENT ADAPTIVE ENCODING

Recommender systems are typically deployed at cloud servers as their operations are resource-demanding. We thus adopt a similar strategy here. On the other hand, unlike traditional user-video interaction metrics (e.g., hitting *like* and watching duration), whose data size is very minimal, the size of the time-series multi-modal behavioral signals is enormous for even minutes of video watching. As a result, consistently uploading the raw readings would consume significant energy overhead. It becomes a critical issue, especially for the battery-powered standalone VR headsets. Figure 7 shows the device's energy consumption breakdowns of one minute of video watching. Data transmission takes more than half of the energy consumption. Figure 8 further shows that the energy consumption grows linearly with the video length.

We further measure the power consumption on the VR device for *case 1*: continuously acquiring behavioral signals vs *case 2*: acquiring signals only (without uploading). As indicated in Table 1, data uploading incurs 52.1% terminal power consumption of *Bere*. The battery life (excluding video play and idle power consumption) significantly drops from 34 hours to 16 hours when continuously uploading raw data.

Motivated by our observation, we propose to diminish the uploaded data size to reduce the corresponding energy consumption on the terminal. Specifically, we develop an encoding scheme that compresses the raw signals into vector embeddings. They are then uploaded to the cloud and serve as inputs for the GCN model. Given a behavioral signal, the

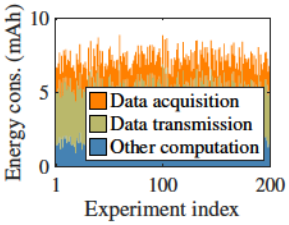


Figure 7: Empirical energy consumption breakdown on the VR headset.

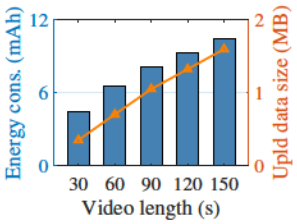


Figure 8: Energy consumption and data size with respect to video length.

Table 1: Power consumption comparison.

Case	Power consumption	Battery life
1. Acq. + uploading	119 μA	5.2 h
2. Acquisition only	57 μA	10.8 h

Algorithm 1: Energy-efficient Adaptive Encoding.

Input: Behavioral signal $\boldsymbol{\phi}$; embedding length k; minimum segment length t; entropy threshold ϵ_θ; pre-trained model *LSTM*

Output: Signal embedding $\boldsymbol{e}$

$\boldsymbol{e} \leftarrow Zeros(k)$; $c \leftarrow 0$; $i \leftarrow 0$; $s \leftarrow \boldsymbol{\phi}(0:t)$ // Initialize
while $i < |\boldsymbol{\phi}|$ **do**
  while $ShannonEntropy(s) < \epsilon_\theta$ **do**
    $s.append(\boldsymbol{\phi}(i))$; $i \leftarrow i+1$; // Add next point
  $\boldsymbol{e} \leftarrow \boldsymbol{e} + LSTM(s,k)$; // Add segment embedding
  $s \leftarrow \boldsymbol{\phi}(i:i+t)$; $i \leftarrow i+t$; $c \leftarrow c+1$;
$\boldsymbol{e} \leftarrow \boldsymbol{e}/c$; // Average for the signal embedding

encoder divides it into segments depending on the entropy and encodes each segment into an embedding with an adaptive compression ratio; the signal embedding is the average of all segment embeddings. In this way, segments with higher entropy are preserved with more information with a lower compression ratio; segments with lower entropy contain less information and are thus more aggressively compressed.

Algorithm 1 outlines our encoder design. After a video session, the encoder takes as input the time-series behavioral signal $\boldsymbol{\phi}$ and outputs a vector embedding $\boldsymbol{e}$. We first divide $\boldsymbol{\phi}$ into multiple segments s: we apply a sliding window that continuously adds data points to a segment until its Shannon entropy reaches a threshold value. Then, each segment is passed into an LSTM model to generate a fixed-length embedding. Finally, all segment embeddings are averaged as the signal embedding $\boldsymbol{e}$. The intuition is that the more informative segments should be preserved to benefit recommendation in later stages; in contrast, aggressive compression can be applied to segments with lower information. According to our testing, given a raw time-series behavioral signal of 1 MB, its corresponding embedding only takes less than 1 KB.

Table 2: A list of public datasets adopted in *Bere*.

Public dataset	# users	# videos	# interactions
MovieLens-100K	1,000	1,700	100,000
MovieLens-1M	6,040	3,884	1,000,209
TikTok-1/50	1,434	29,662	95,426
TikTok-1/5	16,538	366,017	1,047,358

8 EVALUATION

8.1 Evaluation Setup

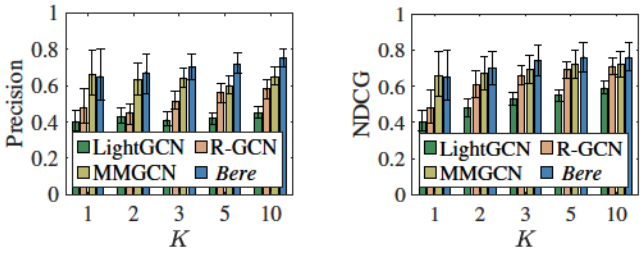
We develop a VR app on a Focus 3 VR Headset running an Android OS and collect our hybrid dataset involving 60 participants, 400 videos, and 3K interactions. The VR app is used to play videos, enable controller interaction, and acquire behavioral signals through the headset's embedded eye tracker and IMU sensor at a sampling rate of 120 Hz and 200 Hz, respectively. Videos are accessed via API from online resources such as YouTube and TikTok. They cover a wide range of topics and categories. The cloud server is in charge of graph construction, model training, and recommendation.

Before data collection, each participant is required to fill out the screening questionnaire and read and sign the consent form. Then, the participant is instructed by a researcher through the calibration phase and basic operations. During data collection, their behavioral signals are captured in real-time. The participant watches 50 videos from a randomly sampled video set. Participants can freely interact with the video by hitting the *like* or the share button, fast-forwarding, rewinding the video play, or skipping the current video as they like. After a video play, the participant is asked to rate the preference score from 1 (lowest) to 5 (highest) based on how much they enjoy watching this video. After watching the entire video set, each participant is compensated with $10. After the main data collection phase, participants were invited to a user study. All data collection phases are carried out in a university lab with normal lighting and environmental conditions. The entire experiment takes around 1 hour for each participant. The study meets all ethical requirements and holds active IRB approval at the researchers' university.

After acquiring the hybrid dataset, we randomly divide it (user-wise) into a training set (80%) for model fine-tuning and a testing set (20%) for evaluation. To pre-train the base model, we employ and study the commonly adopted public datasets as listed in Table 2 [17, 18, 52].

To demonstrate the superior performance of *Bere*, we adopt the following state-of-the-art recommendation models as baselines for a comprehensive comparison.

- LightGCN [22], a GCN model with a simplified and concise design, tailored for recommendation tasks.



(a) Precision (b) NDCG

Figure 9: Overall performance comparison.

- R-GCN [47], a GCN framework focusing on relational modeling and graph construction, which has been effectively used for recommendation.
- MMGCN [57], a recommendation framework that considers multiple modalities or types of information when generating embeddings of users and items.

We choose two commonly adopted metrics to evaluate *Bere*: *precision* and *the normalized discounted cumulative gain (NDCG)*. Precision is defined as the proportion of recommended videos that are actually liked by the target user. It serves as an important measure for precise recommendation in the real-world. NDCG measures how well recommended videos are ranked compared to an ideal ranking, which evaluates the overall quality of recommender systems.

8.2 Overall Performance

We showcase the overall performance of *Bere* and compare the with state-of-the-art baseline models in Figure 9. Generally, *Bere* achieves higher recommendation precision and NDCG than all baselines. This indicates that *Bere* can deliver recommendations to target users more precisely with higher ranking quality to attract users in the real-world. For top-1 recommendation (K=1), *Bere* maintains similar precision with MMGCN; as K grows to 10, *Bere* outperforms all state-of-the-arts significantly by 0.11-0.31 (16.3-68.0%) in precision and 0.04-0.17 (5.6-28.8%) in NDCG. This is promising as K=10 is a commonly adopted real-world setting. In summary, *Bere* achieves superior performance compared with all state-of-the-art solutions for video recommendation in VR.

We observe that the number of recommended items K in the top-K recommendation plays a crucial role in the recommendation performance. To investigate its impact on *Bere*, we change the value of K within {1, 2, 3, 5, 10} and exhibit the corresponding performance@K. Within 5 recommended items, increasing K would slightly increase the recommendation accuracy. This may be caused by the enlarged diversity and item space coverage, which may better cater to the user's varied preferences and needs, reducing uncertainty with more recommended items. On the other hand, as K continuously increases, the recommendation precision and NDCG remain stable. This is potentially because recommending more items may introduce the lower-ranking items in the recommendation list, which may not be generated as accurately as the top-ranking ones, thus affecting the overall accuracy. Following common settings [21, 28], we adopt K as 10 for the rest of our evaluation, as a too small K value would limit coverage and navigation freedom, whereas a too large K value would decrease precision and efficiency.

Table 3: Precision@10 w.r.t. base models and datasets.

Base model	Movie-Lens-100K	Movie-Lens-1M	Tik-Tok-1/50	Tik-Tok-1/5
Vanilla GCN	0.344	0.393	0.570	0.620
LightGCN	0.393	0.437	0.627	0.699
R-GCN	0.408	0.455	0.699	0.751
MMGCN	0.436	0.489	0.742	**0.755**

Note that our training and testing sets are split with no user overlaps; therefore, the above results indicate that *Bere* produces recommendation for unseen users with good performance. A possible explanation is that *Bere* is pre-trained on a large-scale dataset covering a wide spectrum of heterogeneous users, and discovers the hidden link between their traditional and behavioral interactions through CMCCDA.

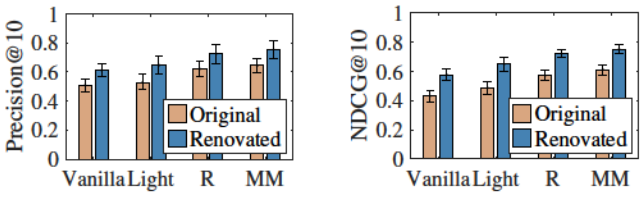
8.3 Impact of Base Models and Datasets

The model pre-trained on a public dataset plays an important role in *Bere*'s performance. We compare the recommendation precision of *Bere* within four base models, namely vanilla GCN, LightGCN, R-GCN, and MMGCN. The graph learning models in these works are all members within the GCN family and therefore share the common basic kernel structure. This allows us to implement our renovated learning technique proposed in §5.2 in their message passing functions.

We pre-train these base models on each public dataset. Table 3 demonstrates the result of their top-10 precision. We observe that using MMGCN pre-trained on TikTok-1/50 renders the highest precision score (0.755), followed by R-GCN on TikTok-1/50 (0.751); the lowest precision (0.344) is obtained by adopting vanilla GCN on MovieLens-100K. A potential reason for MMGCN's superior performance is the attentiveness to the user-video interaction, which better suits our scenario. TikTok datasets perform better than MovieLens due to more similar video genres and durations to ours. For optimal performance, we select MMGCN as the base model and pre-train it on TikTok-1/50 as the basis of *Bere*.

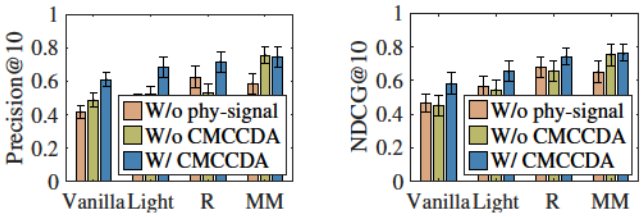
8.4 Ablation Study

Renovated GCN. We investigate the effectiveness of the renovated GCN introduced in §5.2, one of the core techniques in *Bere*, compared with the original GCN in the base models. We



(a) Precision@10 (b) NDCG@10

Figure 10: Ablation study of renovated GCN.



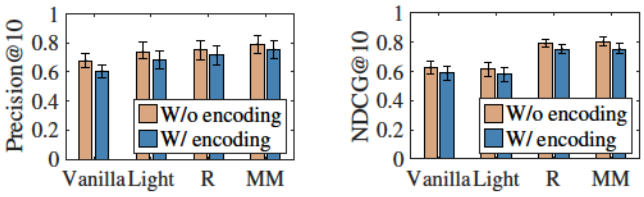
(a) Precision@10 (b) NDCG@10

Figure 11: Ablation study of CMCCDA.

analyze the performance of *Bere* and that of each base model using its original graph learning strategy. As illustrated in Figure 10, compared to each base model, *Bere* significant improves the recommendation precision by 0.10-0.12. Similarly, a 0.14-0.16 improvement in NDCG from the base models also suggests the effectiveness of the proposed graph learning in *Bere*. The major reason is its improved ability to excavate and preserve essential information from behavioral signals.

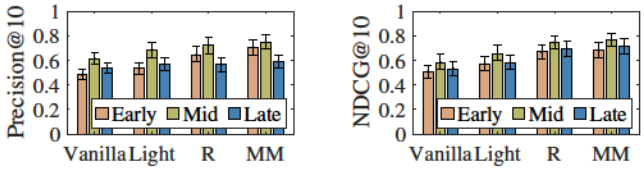
CMCCDA. We study the efficacy of CMCCDA proposed in §6. Three strategies are compared: a) *Bere* with CMCCDA, b) *Bere* without CMCCDA, equivalent to using the pre-trained model without domain adaptation, and c) *Bere* with only traditional interaction data. Figure 11 demonstrates the result. With a slight variance across datasets, we can clearly observe that our strategy, a) *Bere* with CMCCDA, achieves the best result. Surprisingly, b) *Bere* without CMCCDA renders even worse performance than c) *Bere* without behavioral signals. This may be due to the large domain distance, making the behavioral signal data an overwhelming noise that does not improve but even deteriorates the pre-trained model's performance. This proves that CMCCDA is an indispensable technique in *Bere* by "teaching" the model the knowledge of the VR context and behavioral signals.

Energy-efficient adaptive encoding. Lastly, we evaluate the energy-efficient adaptive encoding proposed in §7. We compare performance between *Bere* with encoding vs. that without encoding, i.e., uploading raw signals and extracting embeddings later on the cloud. Here we focus on its performance drop, an inevitable side-effect of encoding due to certain data loss. We evaluate this drop from the optimal baseline, i.e., upload raw signals without energy limitations. As illustrated in Figure 12, *Bere* with adaptive encoding achieves a comparable performance with the optimal



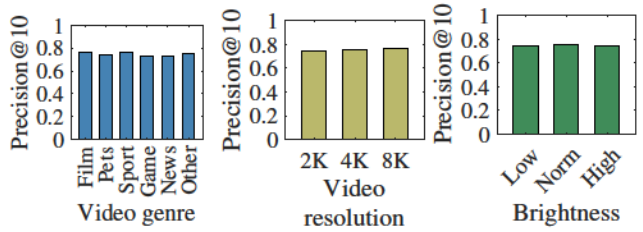
(a) Precision@10 (b) NDCG@10

Figure 12: Ablation study of energy-efficient adaptive encoding in recommendation performance.



(a) Precision@10 (b) NDCG@10

Figure 13: Comparison of three design choices.



(a) Video genre (b) Resolution (c) Brightness

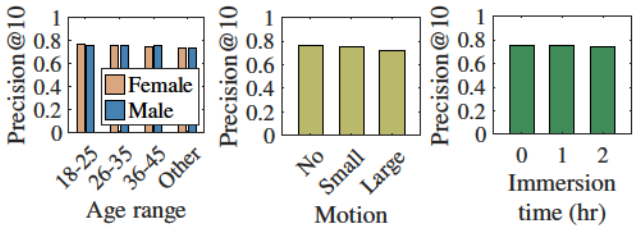
Figure 14: Impact of video-related factors.

baseline with only a marginal drop in precision (0.04-0.07) and NDCG (0.04-0.05). This result indicates that our adaptive encoding strategy preserves recommendation performance.

8.5 Multi-Modality Design Choices

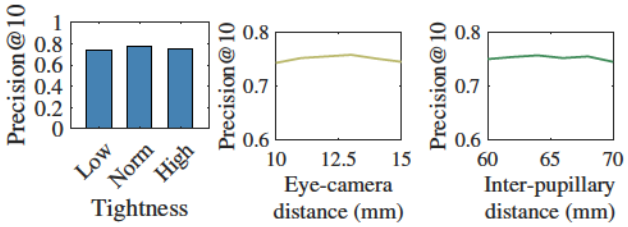
As introduced in §5.2, multi-modality data are fused at the feature level in the middle stage of the learning framework. Here we explore the possibility of two other design choices: early-stage (signal-level) fusion and late-stage (decision-level) fusion. The former integrates signals from different modalities before passing them into one single GCN model; here we adopt concatenation for its simplicity and the maximal data originality. The latter, on the other hand, involves parallel pipelines for different modalities and generates the final preference result based on each preference predictor's output.

Figure 13 exhibits the performance of *Bere* with these different multi-modality design choices. Among them, middle-stage (feature-level) fusion, adopted in *Bere*, results in the highest precision and NDCG, potentially due to its advantage in effectively exploiting the synergies between different modalities and its balanced informativeness and efficiency.



(a) User demo (b) User motion (c) Immersion time

Figure 15: Impact of user-related factors.



(a) Head strap (b) ECD (c) IPD

Figure 16: Impact of device wearing styles.

8.6 Robustness Against Impact Factors

It is important for *Bere* to make recommendation videos with a robust, unbiased performance across various impact factors related to videos, users, and their wearing styles of the device. First, we categorize the tested videos into 6 genres (see Figure 14a). We further randomly select 10 videos and prepare three different resolutions versions (2K, 4K, and 8K) for each source, rendered at three different brightness levels to users. As shown in Figure 14, *Bere* exhibits similar performance across video genres with minimal stand deviation $\sigma = 0.014$. Notably, *film* yields the highest precision at 0.761, while *news* sees the lowest at 0.728. This disparity is possibly attributed to users' higher visual engagement with film content compared to the relatively lower visual attention watching news. Meanwhile, *Bere* remains robust against video resolution and brightness ($\sigma < 0.01$). The highest performance is achieved at 8K resolution with normal brightness.

Next, we study the impact of user demographics, motion, and immersion time. We categorize participants based on gender and age range and display the performance in Figure 15a. The recommendation precision of *Bere* is consistently over 0.732 across various demographics with minimal fluctuations. A *Kruskal–Wallis test* suggests no significant difference among each demographic group ($p > 0.05$). This indicates that *Bere* works well for all users with no discrimination. Then, we randomly select 10 users to watch videos while performing different levels of body motions (no, small, and large), and another 10 users to watch videos after 0, 1, and 2 hours of immersion in the VR environment. As demonstrated in Figure 15b, large motion reduces the recommendation precision by 0.05, potentially due to the increased tracking errors

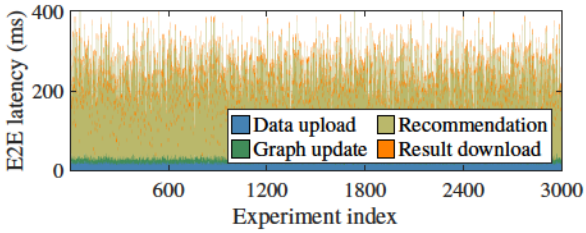


Figure 17: End-to-end latency breakdown.

of the sensors. Figure 15c indicates that *Bere*'s performance is slightly impacted with 2 hours (or more) of immersion time, which may cause user fatigue. We thus suggest users avoid excessive body motions and take rests between long video watching sessions in the real-world use.

Lastly, we assess the impact of users' wearing styles of the VR device, i.e., head strap tightness, eye-camera distance (ECD), and inter-pupillary distance (IPD), which may affect the tracking accuracy of the sensors. To this end, we recruited another 8 participants to conduct the experiments while wearing the VR devices with different styles. As displayed in Figure 16, both the head strap tightness and ECD may slightly impact the recommendation precision, and the best performance is achieved with normal tightness and 13 mm of ECD. On the other hand, IPD is more randomly scattered due to its idiosyncratic nature. To maximize user comfort and system performance, we suggest users to carefully adjust their wearing styles and calibrate before use.

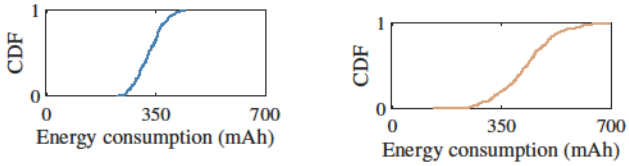
8.7 System Overhead

End-to-end latency. The end-to-end latency of *Bere* is defined as the time interval between a user finishes watching a video and when she receives a new recommendation. As shown in Figure 17, the end-to-end latency for *Bere* ranges from 135 ms to 414 ms with an average of 225 ms, which is practically acceptable for real-world adoption.

Energy consumption. We evaluate the energy overhead of *Bere* at VR terminals. The CDF is plotted in Figure 18a. The energy overhead of adopting *Bere* ranges from 234 mAh to 438 mAh with an average of 330 mAh per hour, translating to 14.9% battery drop on the device. We also test the energy consumption of uploading the raw signals without encoding, as shown in Figure 18b. The energy consumption ranges from 192 mAh to 696 mAh with an average of 428 mAh. Thus, 30% energy overhead is saved by applying our adaptive encoding scheme, extending battery life by 1.5 hours.

8.8 User Study

12 participants completed the user study by responding to a survey, which consists of 8 questions in a five-point Likert



(a) Encoding + uploading **(b) Raw uploading**

Figure 18: CDFs of energy consumption of encoding and uploading embeddings vs uploading raw signals.

Table 4: Questionnaire.

	Question
Q1	I like the content *Bere* recommends for me.
Q2	I think *Bere* does well in recommendation ranking.
Q3	I can imagine myself using *Bere* frequently.
Q4	I think *Bere* is easy to use.
Q5	I find that functions in *Bere* are well integrated.
Q6	I think there is not too much inconsistency in *Bere*.
Q7	*Bere* can be generalized to a wide range of devices.
Q8	I think *Bere* can work for a wide range of users.
O1	What is your overall experience with *Bere*?
O2	Do you have any suggestions to improve *Bere*?

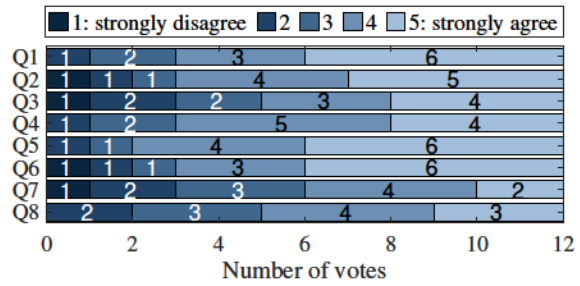


Figure 19: Subjective feedback to the questionnaire.

scale regarding *Bere*'s performance, usability, and deployability, and 2 open-ended questions for overall experience and suggestions for improvement. Table 4 lists all questions.

The distribution of participants' subjective feedback is displayed in Figure 19. For all questions, at least 50% participants provided a score of 4 (agree) or 5 (strongly agree). The average score of each question ranges from 3.33 (Q7) to 4.25 (Q5). This indicates that *Bere* is well received among users. For open-ended questions, most participants provided positive answers to O1, e.g., "*It learns my preferences so fast and makes accurate recommendations!*" Some concerns were raised regarding generalization, e.g., "*I'm not confident if other VR models can provide a similar level of tracking accuracy and comparable recommendation performance.*" and privacy, e.g., "*I'm a bit concerned if my eye data are uploaded to the cloud, and some privacy countermeasures can be added in the future.*" These will be part of our future work.

9 DISCUSSION AND FUTURE WORK

Other human-induced signals. This work aims to investigate the feasibility of incorporating human signals into video recommendation in VR. We focus on gaze and head movement. Other data such as pupillometry and facial expression can be acquired by commercial VR devices and adopted in *Bere* too. For example, the correlation between pupil size and user perception has been established [25, 63]. We plan to extend *Bere* by incorporating other signals to further enhance its performance in the future.

Privacy considerations. Uploading behavioral signals to the cloud server may expose user privacy. Fortunately, in *Bere*, only signal embeddings are uploaded, rather than raw measures, which mitigates the privacy concern. Yet, prior work has pointed out that this still poses potential privacy threats [2], e.g., reconstruction attacks that may reveal the original data. Existing privacy-preserving techniques such as homomorphic encryption, differential privacy, and federated learning can provide potential solutions. We plan to integrate these approaches into our design in the future.

Other graph learning models. *Bere* applies GCN as the basis for graph learning. There are several other graph learning models, such as Graph Autoencoder [32], GraphSAGE [20], and graph attention networks (GAT) [55], which have also demonstrated their advanced performance in a range of graph-based tasks. As part of our future work, we plan to modify *Bere* by incorporating other graph learning models and compare their recommendation performances.

10 CONCLUSION

In this paper, we introduce *Bere*, a behavioral-signal-enhanced video recommender system for VR. To integrate behavioral signals into the recommendation framework, we renovate the GCN learning paradigm to extract essential information from these signals. To address the data scarcity problem during model training, we propose a novel domain adaptation strategy CMCCDA to bridge the discrepancy between the source and target domains. We further develop an energy-efficient adaptive encoding algorithm to improve energy efficiency on the VR device. We demonstrate through a comprehensive evaluation that *Bere* outperforms state-of-the-art solutions by up to 68.0% in recommendation precision and up to 28.8% in the ranking quality.

ACKNOWLEDGEMENT

The work of Ming Li is partially supported by NSF under grants CNS-1943509 and CNS-2343618. The work of Wenqiang Jin is supported by National Natural Science Foundation of China (62202150), Hunan Provincial Key Research and Development Program (2024AQ2041), and YueLuShan Center Industrial Innovation (2024YCII0110).

REFERENCES

[1] Gediminas Adomavicius and Jingjing Zhang. Impact of data characteristics on recommender systems performance. *ACM Transactions on Management Information Systems (TMIS)*, 3(1):1–17, 2012.

[2] Mohammad Al-Rubaie and J Morris Chang. Privacy-preserving machine learning: Threats and solutions. *IEEE Security & Privacy*, 17(2):49–58, 2019.

[3] Xuan Bao, Songchun Fan, Alexander Varshavsky, Kevin Li, and Romit Roy Choudhury. Your reactions suggest you liked the movie: Automatic content rating via reaction sensing. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, pages 197–206, 2013.

[4] Daniel Billsus and Michael J Pazzani. User modeling for adaptive news access. *User modeling and user-adapted interaction*, 10:147–180, 2000.

[5] Robin Burke. Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction*, 12:331–370, 2002.

[6] Christos Calcanis, Vic Callaghan, Michael Gardner, and Matthew Walker. Towards end-user physiological profiling for video recommendation engines. 2008.

[7] Erion Çano and Maurizio Morisio. Hybrid recommender systems: A systematic literature review. *Intelligent data analysis*, 21(6):1487–1524, 2017.

[8] Christoforos Christoforou, Spyros Christou-Champi, Fofi Constantinidou, and Maria Theodorou. From the eyes and the heart: a novel eye-gaze metric that predicts video preferences of a large audience. *Frontiers in psychology*, 6:118967, 2015.

[9] Christoforos Christoforou, Timothy C Papadopoulos, Fofi Constantinidou, and Maria Theodorou. Your brain on the movies: a computational approach for predicting box-office performance from viewer's brain responses to movie trailers. *Frontiers in neuroinformatics*, 11:72, 2017.

[10] Paul Covington, Jay Adams, and Emre Sargin. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*, pages 191–198, 2016.

[11] James Davidson, Benjamin Liebald, Junning Liu, Palash Nandy, Taylor Van Vleet, Ullas Gargi, Sujoy Gupta, Yu He, Mike Lambert, Blake Livingston, et al. The youtube video recommendation system. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 293–296, 2010.

[12] Toon De Pessemier, Ine Coppens, and Luc Martens. Evaluating facial recognition services as interaction technique for recommender systems. *Multimedia Tools and Applications*, 79(31):23547–23570, 2020.

[13] Yaling Deng, Ye Wang, Liming Xu, Xiangli Meng, and Lingxiao Wang. Do you like it or not? identifying preference using an electroencephalogram during the viewing of short videos. *PsyCh Journal*, 12(3):421–429, 2023.

[14] Yancarlos Diaz, Cecilia O Alm, Ifeoma Nwogu, and Reynold Bailey. Towards an affective video recommendation system. In *2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pages 137–142. IEEE, 2018.

[15] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. Graph neural networks for social recommendation. In *The world wide web conference*, pages 417–426, 2019.

[16] Cristos Goodrow. On youtube's recommendation system, 2021.

[17] GroupLens. Movielens 100k dataset. https://grouplens.org/datasets/movielens/100k/, 2023.

[18] GroupLens. Movielens 1m dataset. https://grouplens.org/datasets/movielens/1m/, 2023.

[19] Alan LV Guedes, Roberto G de A Azevedo, Pascal Frossard, Sérgio Colcher, and Simone Diniz Junqueira Barbosa. Subjective evaluation of 360-degree sensory experiences. In *2019 IEEE 21st International Workshop on Multimedia Signal Processing (MMSP)*, pages 1–6. IEEE, 2019.

[20] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

[21] Jinkun Han, Wei Li, Zhipeng Cai, and Yingshu Li. Multi-aggregator time-warping heterogeneous graph neural network for personalized micro-video recommendation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 676–685, 2022.

[22] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 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*, pages 639–648, 2020.

[23] Melanie Heck, Janick Edinger, Jonathan Bünemann, and Christian Becker. Exploring gaze-based prediction strategies for preference detection in dynamic interface elements. In *Proceedings of the 2021 Conference on Human Information Interaction and Retrieval*, pages 129–139, 2021.

[24] Jonathan L Herlocker, Joseph A Konstan, Loren G Terveen, and John T Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22(1):5–53, 2004.

[25] Eckhard H Hess. The role of pupil size in communication. *Scientific American*, 233(5):110–119, 1975.

[26] Yanxiang Huang, Bin Cui, Jie Jiang, Kunqian Hong, Wenyu Zhang, and Yiran Xie. Real-time video recommendation exploration. In *Proceedings of the 2016 international conference on management of data*, pages 35–46, 2016.

[27] Folasade Olubusola Isinkaye, Yetunde O Folajimi, and Bolande Adefowoke Ojokoh. Recommendation systems: Principles, methods and evaluation. *Egyptian informatics journal*, 16(3):261–273, 2015.

[28] Hao Jiang, Wenjie Wang, Yinwei Wei, Zan Gao, Yinglong Wang, and Liqiang Nie. What aspect do you like: Multi-scale time-aware user interest modeling for micro-video recommendation. In *Proceedings of the 28th ACM International conference on Multimedia*, pages 3487–3495, 2020.

[29] Hanseul Jun, Mark Roman Miller, Fernanda Herrera, Byron Reeves, and Jeremy N Bailenson. Stimulus sampling with 360-videos: Examining head movements, arousal, presence, simulator sickness, and preference on a large sample of participants and videos. *IEEE Transactions on Affective Computing*, 13(3):1416–1425, 2020.

[30] Hak Gu Kim, Heoun-Taek Lim, Sangmin Lee, and Yong Man Ro. Vrsa net: Vr sickness assessment considering exceptional motion for 360 vr video. *IEEE transactions on image processing*, 28(4):1646–1660, 2018.

[31] Si Jung Kim, Teemu H Laine, and Hae Jung Suk. Presence effects in virtual reality based on user characteristics: Attention, enjoyment, and memory. *Electronics*, 10(9):1051, 2021.

[32] Thomas N Kipf and Max Welling. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*, 2016.

[33] Yehuda Koren, Steffen Rendle, and Robert Bell. Advances in collaborative filtering. *Recommender systems handbook*, pages 91–142, 2021.

[34] Min-Seok Lee, Seok Ho Baek, Yoo-Jeong Shim, and Myeong-Jin Lee. Analysis of object-centric visual preference in 360-degree videos. *IEEE Access*, 9:98026–98038, 2021.

[35] Fan Liu, Zhiyong Cheng, Lei Zhu, Zan Gao, and Liqiang Nie. Interest-aware message-passing gcn for recommendation. In *Proceedings of the Web Conference 2021*, pages 1296–1305, 2021.

[36] Abhishek Mahata, Nandini Saini, Sneha Saharawat, and Ritu Tiwari. Intelligent movie recommender system using machine learning. In *Intelligent Human Computer Interaction: 8th International Conference, IHCI 2016, Pilani, India, December 12-13, 2016, Proceedings 8*, pages

94–110. Springer, 2017.

[37] Tao Mei, Bo Yang, Xian-Sheng Hua, and Shipeng Li. Contextual video recommendation by multimodal relevance and user feedback. *ACM Transactions on Information Systems (TOIS)*, 29(2):1–24, 2011.

[38] Jinyoung Moon, Youngrae Kim, Hyungjik Lee, Changseok Bae, and Wan Chul Yoon. Extraction of user preference for video stimuli using eeg-based user responses. *ETRI Journal*, 35(6):1105–1114, 2013.

[39] Raymond J Mooney and Loriene Roy. Content-based book recommending using learning for text categorization. In *Proceedings of the fifth ACM conference on Digital libraries*, pages 195–204, 2000.

[40] Deuk Hee Park, Hyea Kyeong Kim, Il Young Choi, and Jae Kyeong Kim. A literature review and classification of recommender systems research. *Expert systems with applications*, 39(11):10059–10072, 2012.

[41] Michael J Pazzani and Daniel Billsus. Content-based recommendation systems. In *The adaptive web: methods and strategies of web personalization*, pages 325–341. Springer, 2007.

[42] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618*, 2012.

[43] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. Grouplens: An open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, pages 175–186, 1994.

[44] Francesco Ricci, Lior Rokach, and Bracha Shapira. Recommender systems: Techniques, applications, and challenges. *Recommender Systems Handbook*, pages 1–35, 2021.

[45] J Ben Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen. Collaborative filtering recommender systems. In *The adaptive web: methods and strategies of web personalization*, pages 291–324. Springer, 2007.

[46] J Ben Schafer, Joseph Konstan, and John Riedl. Recommender systems in e-commerce. In *Proceedings of the 1st ACM conference on Electronic commerce*, pages 158–166, 1999.

[47] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *The Semantic Web: 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, Proceedings 15*, pages 593–607. Springer, 2018.

[48] ABM Fahim Shahriar, Mahedee Zaman Moon, Hasan Mahmud, and Kamrul Hasan. Online product recommendation system by using eye gaze data. In *Proceedings of the International Conference on Computing Advancements*, pages 1–7, 2020.

[49] Chuan Shi, Binbin Hu, Wayne Xin Zhao, and S Yu Philip. Heterogeneous information network embedding for recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 31(2):357–370, 2018.

[50] Tsz Yan So, Man Yi Erica Li, and Hakwan Lau. Between-subject correlation of heart rate variability predicts movie preferences. *PloS one*, 16(2):e0247625, 2021.

[51] David H Stern, Ralf Herbrich, and Thore Graepel. Matchbox: large scale online bayesian recommendations. In *Proceedings of the 18th international conference on World wide web*, pages 111–120, 2009.

[52] TikTok. Tiktok dataset. https://www.biendata.xyz/competition/icmechallenge2019, 2019.

[53] TikTok. How tiktok recommends videos #foryou, 2020.

[54] Huyen TT Tran, Nam Pham Ngoc, Cuong T Pham, Yong Ju Jung, and Truong Cong Thang. A subjective study on qoe of 360 video for vr communication. In *2017 IEEE 19th international workshop on multimedia signal processing (MMSP)*, pages 1–6. IEEE, 2017.

[55] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, Yoshua Bengio, et al. Graph attention networks. *stat*, 1050(20):10–48550, 2017.

[56] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. Neural graph collaborative filtering. In *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*, pages 165–174, 2019.

[57] Yinwei Wei, Xiang Wang, Liqiang Nie, Xiangnan He, Richang Hong, and Tat-Seng Chua. Mmgcn: Multi-modal graph convolution network for personalized recommendation of micro-video. In *Proceedings of the 27th ACM international conference on multimedia*, pages 1437–1445, 2019.

[58] Songhua Xu, Hao Jiang, and Francis CM Lau. Personalized online document, image and video recommendation via commodity eye-tracking. In *Proceedings of the 2008 ACM conference on Recommender systems*, pages 83–90, 2008.

[59] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 974–983, 2018.

[60] Shingchern D You and Chun-Wei Liu. Classification of user preference for music videos based on eeg recordings. In *2020 IEEE 2nd Global Conference on Life Sciences and Technologies (LifeTech)*, pages 1–2. IEEE, 2020.

[61] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. Deep learning based recommender system: A survey and new perspectives. *ACM computing surveys (CSUR)*, 52(1):1–38, 2019.

[62] Qian Zhao, Shuo Chang, F Maxwell Harper, and Joseph A Konstan. Gaze prediction for recommender systems. In *Proceedings of the 10th ACM Conference on Recommender Systems*, pages 131–138, 2016.

[63] Huadi Zhu, Tianhao Li, Chaowei Wang, Wenqiang Jin, Srinivasan Murali, Mingyan Xiao, Dongqing Ye, and Ming Li. Eyeqoe: a novel qoe assessment model for 360-degree videos using ocular behaviors. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 6(1):1–26, 2022.

[64] Wenjie Zou, Fuzheng Yang, Wei Zhang, Yi Li, and Haoping Yu. A framework for assessing spatial presence of omnidirectional video on virtual reality device. *IEEE Access*, 6:44676–44684, 2018.