Learning the Co-evolution Process on Live Stream Platforms with Dual Self-attention for Next-topic Recommendations

Hsu-Chao Lai
Department of Computer Science, National Yang Ming Chiao Tung University
Hsinchu, Taiwan
hsuchao.cs05@nycu.edu.tw

Philip S. Yu
Department of Computer Science, University of Illinois at Chicago
Chicago, Illinois, USA
psyu@uic.edu

Jiun-Long Huang
Department of Computer Science, National Yang Ming Chiao Tung University
Hsinchu, Taiwan
jlhuang@cs.nycu.edu.tw

ABSTRACT

Live stream platforms have gained popularity in light of emerging social media platforms. Unlike traditional on-demand video platforms, viewers and streamers on the live stream platforms are able to interact in real-time, and this makes viewer interests and live stream topics mutually affect each other on the fly, which is the unique *co-evolution* phenomenon on live stream platforms. In this paper, we make the first attempt to introduce a novel next-topic recommendation problem for the streamers, LSNR which incorporates the co-evolution phenomenon. A novel framework CENTR introducing the Co-evolutionary Sequence Embedding Structure that captures the temporal relations of viewer interests and live stream topic sequences with two stacks of self-attention layers is proposed. Instead of learning the sequences individually, a novel dual self-attention mechanism is designed to model interactions between the sequences. The dual self-attention includes two modules, LCA and LVA, to leverage viewer loyalty to improve efficiency and flexibility. Finally, to facilitate cold-start recommendations for new streamers, a collaborative diffusion mechanism is implemented to improve a meta learner. Through the experiments in real datasets, CENTR outperforms state-of-the-art recommender systems in both regular and cold-start scenarios.

CCS CONCEPTS

• **Information systems** → **Web applications**; **Recommender systems**.

KEYWORDS

Co-evolution; next-topic recommendation; live stream platform

ACM Reference Format:
Hsu-Chao Lai, Philip S. Yu, and Jiun-Long Huang. 2023. Learning the Co-evolution Process on Live Stream Platforms with Dual Self-attention for Next-topic Recommendations. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23), October 21–25, 2023, Birmingham, United Kingdom.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3583780.3614952

CIKM '23, October 21–25, 2023, Birmingham, United Kingdom

ACM ISBN 979-8-4007-0124-5/23/10…$15.00
https://doi.org/10.1145/3583780.3614952

1 INTRODUCTION

With the great improvement on data transferring and image processing, live streaming platforms such as Twitch, Facebook Gaming, YouTube Live, and Microsoft Mixer have grown to be the most popular social media. Compared to traditional on-demand video platforms (e.g., YouTube and Netflix), live streaming platforms provide more immersive user experience by supporting viewers interacting with streamers (i.e., people broadcasting contents) in real-time, which generates huge revenue. According to Digital 2022 Global Overview Report[1], about 30.4% of the global internet users have identified live video streams as the type of video content that they watch the most each week. The global live streaming market was valued at 38 billion USD in 2021 and is expected to grow to 245 billion by 2027, reported by Bloomberg in 2022.[2]

The success of live stream platforms can be credited to their interactive and immersive user experience. Despite that the researchers of recommender systems have published a line of studies regarding the on-demand video platforms, they haven't put much attention on the live streaming platforms. Here we make the first attempt to introduce the uniqueness of live steam platforms:

i) *Real-time social interaction.* Unlike watching videos solely on on-demand video platforms, viewers on live steam platforms can easily interact with streamers and other viewers via commenting, donating, or virtual gifting. On the other hand, streamers usually respond verbally and instantly to viewers, and they sometimes organize online events to engage with their loyal viewers. According to a report from Facebook in 2016, friends interact 10 times more on live streams than on pre-recorded videos.[3] Donations enable streamers to easily wade through a large number of chat messages and pay attention to those viewers who donate [12, 19]. These real-time social interactions naturally bring users of live stream platforms together to form a virtual community, and further boost the revenue due to increased viewership and donations.

ii) *Diverse topics of contents.* Thanks to the great improvement on network bandwidth and high-resolution cameras, streamers are able to produce and share contents of their interests online in high quality. Starting from gaming and eSports, live streaming platforms have developed multiple and diverse topics, such as live commerce, just chatting, music playing, and more. There were more than 300 topics on Twitch in February 2022, and the amount keeps growing

[1]https://wearesocial.com/uk/blog/2022/01/digital-2022-another-year-of-bumper-growth-2/
[2]https://www.bloomberg.com/press-releases/2022-11-23/live-streaming-market-to-grow-at-a-cagr-of-16-14-valuates-reports
[3]https://about.fb.com/news/2016/04/introducing-new-ways-to-create-share-and-discover-live-video-on-facebook/

on the fly. With the great diversity of topics, viewers and streamers of all kinds can meet people with similar interests and accommodate themselves on live streaming platforms.

iii) *Co-evolution.* As a result of the above two characteristics, we recognize a unique phenomenon in live stream platforms, referred to as the *co-evolution* phenomenon [24] between viewer interests and live streaming topics. When streamers are aware of their shared interests with viewers via real-time social interactions, they will interact and constantly engage viewers with their content topics [13]. Otherwise, they could alternate their topics [17]. By dynamically changing the live stream topics, streamers are able to increase viewership, enhance viewer engagement, and build viewer loyalty. On the other hand, while 55% and 64% viewers are respectively reported to be loyal to a certain streamer [7, 19], the personal interests of viewers could also vary by following the evolving streaming topics of their favorite streamers.

Figure 1 illustrates an example of the co-evolution. The upper sequence presents the live stream topics in four continuous sessions, consisting of two shooting games: Fortnite (blue) and Call of Duty (grey). The sequence below shows the viewer interests of those two games. A relatively large picture of a gaming topic indicates that the viewer is more interested in it. In the first session, The viewer was interested in Fortnite more, so the streamer decided to play Fortnite during the second session (affected via green arrows). However, the viewer was not enjoying the second session, so she switched to Call of Duty (affected via blue arrows). Accordingly, the streamer streamed Call of Duty instead in the next session. The live stream topics and the viewer interests keep co-evolving on live streaming platforms, and, indeed, the co-evolution must be considered when building sophisticated recommender systems. Nevertheless, the existing recommender systems [2, 11] for on-demand video platforms do not factor in the co-evolution.

In this paper, we define a novel research problem Live Stream Next-topic Recommendation (LSNR) as follows. Given the interactions of viewers (watch, chat, donate, subscribe, etc) and live streaming topics of streamers, the goal of LSNR is to recommend the next-topic to a streamer to maximize viewership while considering the co-evolution. Unlike traditional video recommendation problems [2, 11] that satisfy viewers only, maximizing viewership to recommend the next-topics may have greater potential to benefit all viewers, streamers, and the live stream platforms. The larger viewership has the potential to generate more revenue for streamers and platforms through viewer donations, subscriptions, ads, and brand sponsorship. Plus, viewers could also be satisfied while their engagements are the measurement of the objective.

LSNR raises three new research challenges: (1) Co-evolution Modeling. In live streaming industries, both viewer interests and streaming topics change over time, and their respective changes are mutually affected. Existing sequential recommenders like BERT4Rec [15] and SASRec [9] have shown great power in extracting evolutionary relations from sequential data (e.g., e-commerce). However, they formulate a single sequence for each user because the items (e.g., products) would not change anyway. In our case, implementing those sequential models to learn the co-evolution may fail.

(2) Imbalanced Viewership. It is worth noting that the viewership of a popular streamer and a new (or unpopular) streamer could be extremely imbalanced. Nevertheless, it is important to align the

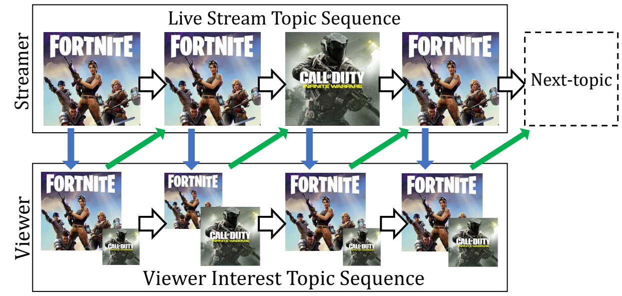


Figure 1: An illustration of the co-evolution.

amount of training samples (viewers in our case) for each instance for machine learning models to prevent from efficiency and performance degradation. Intuitively, sampling viewers from a uniform distribution is effortless and fast. Nevertheless, there may exist too many disloyal (or random) viewers watching popular streamers. Training with those random instances may degrade the performance, so an effective sampling strategy to is needed to address the imbalance viewership.

Finally, (3) Cold-start Issue. Recommender systems are unable to provide accurate recommendations for new users with insufficient interactions or information. In our case, while the live stream industries grow fast, the population of new streamers also increases fast. It is essential for a live stream platform and its recommenders to help new streamers find appropriate topics rapidly, so they are willing to stay on the platform, keep producing contents, and make profits. However, the lack of streaming experience makes it challenging to recommend next-topics. Meta learning [25] has been a state-of-the-art method in cold-start sequential recommendation problems by encoding and matching cold-start sequences to warm-start sequences. However, we argue that the existing meta learners do not fully utilize the information from live streams, leaving room for improvement by considering the behavior of the viewers of cold-start streamers.

To tackle the above challenges of LSNR, we propose a novel two-phase recommendation framework, namely *Co-evolutionary Live Stream Next-topic Recommender System (CENTR)*, based on the self-attention layers. For challenge (1), the first phase introduces a novel *Co-evolutionary Sequence Embedding Structure* to adapt these sequences of live stream topics and viewer interest topics jointly. A *dual self-attention* mechanism, including two loyalty-based modules *Loyalty-based Viewer Cohesion Aggregation (LCA)* and *Loyalty-based Viewer Interest Adaption (LVA)*, is designed to make the sequences interact mutually. For challenge (2), LCA and LVA further leverage the loyalty of viewers towards a channel to sample a subset of viewers for training, and hence provide efficiency and flexibility. For instance, the interests of those viewers with greater loyalty (e.g., greater amount of donations) reflect their beloved streamers the most. Sampling them with a higher probability may result in better recommendation results. Beyond that, LCA provides flexibility by increasing the probability of sampling disloyal viewers when a streamer would like explore new viewers.

For challenge (3), a common solution of the cold-start problems is enriching the cold-start embeddings with additional information,

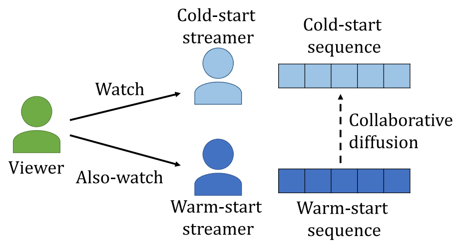


Figure 2: An illustration of the collaborative diffusion.

such as content or social relations. To choose the right information, a *collaborative diffusion mechanism*, as illustrated in Figure 2, is employed to improve the meta learners for cold-start streamers in the second phase of CENTR. Inspired by the classic collaborative filtering mechanism in recommender systems, we assume that a warm-start and a cold-start streamer, both watched by the same viewer, are similar. As a result, the embedding of the also-watched warm-start streamer (dark blue) is diffused to the embedding of the watched cold-start streamer (light blue) due to their shared viewer (green). In comparison to directly match raw sequence embeddings of the cold-start streamers to warm-start embeddings, the collaborative diffusion mechanism enables the meta learner to consider more informative cold-start embeddings for matching. In summary, the paper makes the following main contributions:

- To our best knowledge, this is the first paper to consider the co-evolution phenomenon for recommending next-topics to streamers on live stream platforms. This novel problem is named LSNR and is distinguished by three major challenges.
- To address the challenges, a new framework CENTR, including a dual self-attention structure and two loyalty-based mechanisms, is devised. A collaborative diffusion mechanism is used to improve the cold-start recommendation.
- Experimental results and case studies on real datasets manifest that CENTR outperforms state-of-the-art sequential and video recommender systems on LSNR.

2 RELATED WORK

2.1 Video and Live Stream Recommendation

Existing video recommendation systems focus on extracting useful information from contents and contexts [2, 11]. Chen et al. adopted a multi-level attention structure to analyze components (e.g., regions of frames, characters, and subtitles) of videos to learn sophisticated user preference [2]. Gao et al. captured the dynamic of user preferences by leveraging a Recurrent Neural Network [5]. Nevertheless, the above methods are designed for on-demand platforms, which fail to consider the streamer in live streaming platforms, not to mention the interaction between viewers and streamers. UVCAN [11] first used a co-attention structure to extract both the viewer interests and the video targets as features. However, it does not consider the temporal relations of topic evolution.

Most of the live streaming research focuses on enhancing watching experiences by optimizing network transmitting latency and bitrate [14, 16]. HCI communities concluded the reasoning of unique user behavior (e.g., donation) in live streaming platforms [12, 19]. The recommendation system researchers have yet to put much attention on live steam platforms [23]. Yu et al. used a social-aware attention network to recommend channels based on the content of both live videos and chats [21]. Lai et al. proposed a tensor co-factorization model to make channel recommendation and donation recommendation jointly [10]. Yu et al. employed a GNN-based model to facilitate product recommendation on live streams [22]. Zhang et al. [24] firstly identified the co-evolutionary process in live streams. They further predict the next-topic of streamers and recommend the streamer to viewers jointly. Nevertheless, the above research does not recommend the next-topics for streamers.

2.2 Sequential Recommender Systems

Sequential recommendation aims at learning the dynamics of a previous user behaviors in order to predict the next item [9, 15]. Locker [6] introduced a local-constrained model to improve the short-term predictions in sequential models. TGSRec [18] adopted a collaborative attention to simultaneously capture collaborative signals from both users and items. STOSA [3] devised the uncertainty into model training with a Wasserstein Self-Attention module to model item-item position-wise relationships in sequences. However, the above research was designed for a single sequence (e.g., products), failing to model the co-evolution of viewer interests and live stream topic sequences jointly.

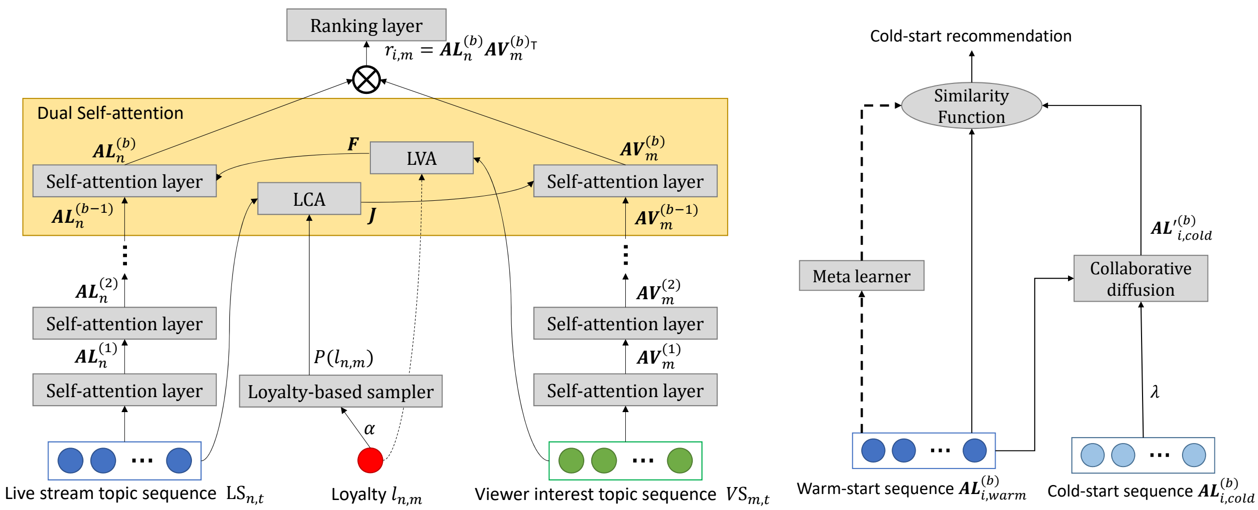
3 RECOMMENDING NEXT-TOPICS FOR LIVE STREAMS

3.1 Problem Definition and Preliminaries

The goal of LSNR is to recommend the best next-topic that maximizes viewer engagement, such as the total amount of donations or the viewership, for a streamer by considering the co-evolutionary influence between the viewer interest topics and the live streaming topics. However, it is challenging to define the best next-topics (as ground-truth) in practice because streamers cannot perform multiple topics simultaneously for comparisons. In lieu of defining the best next-topics artificially, this paper focuses on a proxy objective, which aims to accurately *rank viewer engagement for any set of candidate next-topics*. With this ranking objective, live stream platforms can recommend the candidate next-topic with the greatest total amount of ranked viewer engagement to a streamer. For example, the recommended next-topic will be the one predicted to have the highest sum of donations from viewers.

In the following, U and V denote the universal sets with respect to viewers and streamers. S is a set of live stream topics. Let $\mathbf{v}_{n,t}$ be the latent live streaming topic vector of streamer n at the t-th live streaming session. $LS_{n,t} = \langle \mathbf{v}_{n,t-L+1}, ..., \mathbf{v}_{n,t} \rangle$ is the sequence composed of continuous live streaming topics from the $t - L + 1$-th to the t-th sessions of streamer n, where L is the length of the look-back window. Similarly, let $\mathbf{u}_{m,t}$ be the latent viewer interest topic vector of viewer m at the t-th live streaming session, and $VS_{m,t} = \langle \mathbf{u}_{m.t-L+1}, ..., \mathbf{u}_{m,t} \rangle$ is the sequence of viewer interest topics.[4]

[4]The one-hot encoding for the stream topics and the viewer interest topics based on the labeled topics (e.g., LoL and CoD in Section 4.3) are used for initialization. The live stream topics are what the streamers played during the sessions. The viewer interest topics are what the viewers watched during the sessions



(a) Phase 1: the co-evolutionary sequence embedding structure.

(b) Phase 2: the collaborative diffusion mechanism.

Figure 3: An illustration of the proposed framework CENTR.

Figure 3 illustrates the two-phase CENTR and the overview is as follows. To fully utilize the benefits of the state-of-the-art self-attention layers for the co-evolutionary sequences, CENTR proposes the Co-evolutionary Sequence Embedding Structure in phase one, as shown in Figure 3(a). The live stream topic sequence and the viewer interest topic sequence are respectively fed into two stacks of self-attention layers [9] (the left and the right vertical sequences). The yellow area highlights the innovative *dual self-attention* that enables the mutual effect between two sequences with two novel loyalty modules, *Loyalty-based Viewer Cohesion Aggregation (LCA))* and *Loyalty-based Viewer Interest Adaption (LVA)*. Existing sequential recommenders [9, 15] are designed to process one type of sequence only (e.g., the live stream topic sequence), so they fail to incorporate the important co-evolution like CENTR. Figure 3(b) further shows the design for cold-start recommendation in the second phase. The warm-start sequences are first used to train a similarity function by a meta learner by following the dashed route. For the cold-start sequences, a *collaborative diffusion mechanism* is employed to strengthen the embeddings (the solid route), so the scarcity of their data is alleviated.

We define the the b-th self-attention layer $\mathbf{AL}^{(b)}$ for the live stream topic sequence $LS_{n,t}$ in the setting of single sequence by following [9]. Here we give a brief introduction and please refer to [9] for more details. For the first layer $\mathbf{AL}^{(1)}$:

$$\mathbf{AL}^{(1)} = \mathrm{SA}(\hat{\mathbf{LS}}_{n,t} \cdot \mathbf{W}^{Q}, \hat{\mathbf{LS}}_{n,t} \cdot \mathbf{W}^{K}, \hat{\mathbf{LS}}_{n,t} \cdot \mathbf{W}^{V}), \tag{1}$$

where $\mathbf{AL}$ denotes the self-attentive live stream topic sequences. SA is the abbreviation of Self-Attention and is formulated as the scaled dot-product attention. $\mathbf{W}^{Q}$, $\mathbf{W}^{K}$, and $\mathbf{W}^{V}$ are the projection matrices with respect to queries, keys, and values. $\hat{\mathbf{LS}}_{n,t}$ is the input live stream topic embeddings added up with positional embeddings to gain awareness of the position of each topic in a sequence. For the following layers ($b > 1$), the formulation of $\mathbf{AL}^{(b)}$ is:

$$\mathbf{AL'}^{(b)} = \mathbf{AL}^{(b-1)} + DP(\mathrm{SA}(LN(\mathbf{AL}^{(b-1)}))) \tag{2}$$

$$\mathbf{AL}^{(b)} = \mathbf{AL'}^{(b)} + DP(\mathrm{FFN}(LN(\mathbf{AL'}^{(b)}))), \tag{3}$$

where DP, LN, and FFN denote the Dropout, the Layer Normalization, and the Feed-Forward Network respectively. $\mathbf{AL'}^{(b)}$ is an intermediate result to get $\mathbf{AL}^{(b)}$. Specifically, the input from the $(b-1)$-th layer is normalized first. The outputs of the self-attention layer and the feed-forward network dropout redundant neurons.

3.2 Dual Self-attention in Co-evolutionary Sequence Embedding Structure

To fully utilize the co-evolution between the streaming topics and the viewer interests, CENTR first duplicates Eq. (1) to model the viewer interest topic sequences.

$$\mathbf{AV} = \mathrm{SA}(\hat{\mathbf{VS}}_{m,t} \cdot \mathbf{W}^{Q}, \hat{\mathbf{VS}}_{m,t} \cdot \mathbf{W}^{K}, \hat{\mathbf{VS}}_{m,t} \cdot \mathbf{W}^{V}), \tag{4}$$

where $\mathbf{AV}$ denotes the self-attentive viewer interest topic sequences. $\hat{\mathbf{VS}}_{m,t}$ injects the positional information into $VS_{m,t}$. Here CENTR also applies Layer Normalization and Dropout to $\mathbf{AV}$ as in Eq. (2) and (3) to make stacking multiple self-attention layers stable. The two sequence embeddings $\mathbf{AL}^{(b)}$ and $\mathbf{AV}^{(b)}$ regarding live stream topics and viewer interest topics from the output of the b-th self-attention layers are shown on the left and the right in Figure 3(a).

A common choice of recommenders [1, 10, 21] is to rank the inner product of $\mathbf{AL}^{(b)}$ and $\mathbf{AV}^{(b)}$, as the viewer engagement, for modeling their interactions. Nevertheless, the mutual and temporal co-evolution is not carefully designed, and hence the learning process may not converge well. In order to make the two sequences interact with each other, the dual self-attention is introduced. In the dual self-attention, LCA samples viewers and aggregates their interest cohesion to affect the evolution of streamer topics. On the other hand, LVA propagates live stream topics to viewer interests, weighted by viewer loyalty. With the dual self-attention, CENTR captures the co-evolution and derives sophisticated embeddings.

3.2.1 Loyalty-based Viewer Cohesion Aggregation (LCA). While the viewership has become one of the most important sources of income for the streamers [13], the cohesion of the viewer interests motivates streamers to change their next-topics [17]. Note that understanding the cohesion from viewer interactions could be overwhelming for the streamers because they can be excessive and fast. For example, an analysis on Twitch in 2019 manifests that viewers can generate 6,257 chats per minute in average in a popular channel [4]. Therefore, it is important to identify whose interests are informative in learning the co-evolutionary process. We then propose a novel mechanism, Loyalty-based Viewer Cohesion Aggregation (LCA), to sample viewers based on their loyalty (as shown in the middle of Figure 3(a)).

To integrate the viewer interest cohesion into the self-attention blocks, LCA aggregates viewer interest topic sequences into a matrix $\mathbf{J}$ as follows:

$$\mathbf{J} = \begin{bmatrix} \mathbf{u}_{1,t-L+1}^{\top} & \mathbf{u}_{2,t-L+1}^{\top} & \cdots & \mathbf{u}_{c,t-L+1}^{\top} \\ \mathbf{u}_{1,t-L+2}^{\top} & \mathbf{u}_{2,t-L+2}^{\top} & \cdots & \mathbf{u}_{c,t-L+2}^{\top} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{u}_{1,t}^{\top} & \mathbf{u}_{2,t}^{\top} & \cdots & \mathbf{u}_{c,t}^{\top} \end{bmatrix}.$$

Each row of $\mathbf{J}$ consists of the concatenation of the viewer interest topics of c sampled viewers. Note that the timestamp is aligned to the row-wise setting of $\hat{\mathbf{LS}}_{n,t}$, i.e., from $t-L+1$ to t. Therefore, the fine-grained viewer interests in $\mathbf{J}$ matches the live stream topic in $\hat{\mathbf{LS}}_{n,t}$ at the same row, and is used for extension based on Eq. (2):

$$\mathbf{AL'}^{(b)} = \mathbf{AL}^{(b-1)} + DP(\mathrm{SA}(LN(\mathbf{AL}^{(b-1)}))) + \mathbf{J}\cdot\mathbf{W}^{J}, \quad (5)$$

where $\mathbf{W}^{J}$ projects the high-dimensional fine-grained viewer interests to low-dimensional cohesion matrix. The embeddings $\mathbf{AL}^{(b)}$ now moves towards the cohesion of viewers in the latent space via the last term in Eq. (5).

Imbalanced viewership between famous and non-famous streamers may cause troubles while sampling the viewers in $\mathbf{J}$. A flock of random viewers may exist in popular channels and sampling their interests may cause the embeddings to randomly move in the latent space, which could be extremely harmful for the recommendation. In contrast, the feedback of loyal viewers is more worth considering, and keeping them engaging to the channel is likely to maintain a great viewership. Accordingly, we propose to leverage *loyalty* for sampling viewers with the following exponential distribution:

$$P(l_{n,m}) = Exp(l_{n,m}) = \frac{1}{\alpha\cdot l_{n,avg}}e^{-\frac{l_{n,m}}{\alpha\cdot l_{n,avg}}}, \quad (6)$$

where $l_{n,m}$ is the loyalty of viewer m to streamer n and $l_{n,avg}$ is the average loyalty in the channel of streamer n. $\alpha \in (0,1]$ is the weighting parameter, which provides flexibility for sampling.

It is worth noting that several engagements generated by viewer m to streamer n (e.g., donations) could be a hint to derive $l_{n,m}$, supported by research [17] manifesting that the value of gifts received is a key metric of whether one's performance has been successful, and a basis for deciding when to alter content. CENTR hence calculates $l_{n,m}$ using a regression model based on the following features: 1) the cumulative amount of donations, 2) the cumulative amount of chat messages, 3) the cumulative amount of donations/chats in the recent L slots, 4) the ranks of the cumulative donations/chats among all viewers in the channel, and 5) the cosine similarity between the viewer and the streamer $cos(\mathbf{AL}^{(b)}, \mathbf{AV}^{(b)})$.

Compare to using a uniform distribution, adopting Eq. (6) allows viewers with greater loyalty to be sampled more easily when α is close to 1. LCA thus potentially improves the prediction performance. However, sometimes streamers are encouraged to explore new viewers, in order to expand the viewership. In this case, lowering α can increase the probability of sampling disloyal viewers and their interests could be considered when forming the cohesion matrix $\mathbf{J}$ in Eq. (5). Hence, LCA can be adjusted flexibly for different purposes of streamers and live stream platforms.

3.2.2 Loyalty-based Viewer Interest Adaption (LVA). In addition to the effect of viewer cohesion on streamers, the live stream topics may gradually change the preferences of viewers, especially for loyal viewers. In other words, when a viewer sticks to watching a specific channel, she is likely to adapt herself to the changes of the live stream topics, and hence her preferences are affected.

While a viewer usually watches one channel at a time, it is not necessary to sample multiple streamers like $\mathbf{J}$. The information of those watched channels is aggregated as follows:

$$\mathbf{F} = \begin{bmatrix} \mathbf{v}_{1,t-L+1}^{\top} \\ \mathbf{v}_{2,t-L+2}^{\top} \\ \vdots \\ \mathbf{v}_{s,t}^{\top} \end{bmatrix}, \mathbf{l}_m = \begin{bmatrix} l_{1,m} \\ l_{2,m} \\ \vdots \\ l_{s,m} \end{bmatrix},$$

where each row of $\mathbf{F}$ is the live stream topic embeddings of the watched streamers at the timestamp, which is aligned to the timestamp as $\hat{\mathbf{LS}}_{n,t}$. $\mathbf{l}_m$ is the loyalty vector, composed of the loyalty of the viewer m to those watched channels. Similar to Eq. (5), we extend the b-th self-attention block for viewer interest topics embeddings:

$$\mathbf{AV'}^{(b)} = \mathbf{AV}^{(b-1)} + DP(\mathrm{SA}(LN(\mathbf{AV}^{(b-1)}))) + \mathbf{l}_m \odot \mathbf{F}\cdot\mathbf{W}^{F}, \quad (7)$$

where $\odot$ is the element-wise product and $\mathbf{W}^{F}$ is a learnable weighting matrix. The watched live stream topic embeddings are now weighted by the loyalty vector l_m. The more loyal the viewer is to a streamer, the live stream topic embedding of that streamer has a stronger effect in shifting the viewer interest topics in latent space.

In summary, we propose Co-evolutionary Sequence Embedding Structure and the innovative dual self-attention for modeling the co-evolving process on live stream platforms. Specifically, LCA and LVA entangle two sets of self-attention layers by viewer cohesion and viewer adaption, respectively. Furthermore, LCA and LVA uniquely leverage the viewer loyalty to facilitate flexibility and efficiency for practical usage.

3.3 Ranking Layer

After stacking b self-attention blocks on both the live stream topic sequence and the viewer interest sequence, we aim at recommending the next-topic for streamers based on $\mathbf{AL}^{(b)}$ and $\mathbf{AV}^{(b)}$. However, it is hard to define the ground truth of the best live stream topic in practice so a classification objective (e.g., cross entropy) is inapplicable. Regarding this, we propose a proxy objective that aims at *ranking* a set of candidate next-topics, and the candidate with the highest ranking scores will be recommended.

To derive the ranking scores, CENTR concatenates a Matrix Factorization layer as the ranking layer after the b-th self attention block, as illustrated at the top in Figure 3(a). The ranking score is formulated as follows:

$$\hat{r}_{i,m,n} = \mathbf{AL}_n^{(b)} \mathbf{AV}_m^{(b)\intercal}, \tag{8}$$

where $\hat{r}_{i,m,n}$ denotes the ranking score of live stream topic i being the next-topic of streamer n and viewer m. A high $\hat{r}_{i,m,n}$ indicates that viewer m is more likely to engage (e.g., donate) with topic i.

CENTR implements a pairwise loss function as the ranking objective, which is commonly used in recommendation problems [1]:

$$\min \mathcal{L} = \sum_{n \in V} \sum_{(i,j,m) \in O_n} (\hat{r}_{i,m,n} - \hat{r}_{j,m,n} - 1)^2,$$

where each instance in O_n is a triplet (i, j, m) meaning that viewer m interacts with a next-topic i but not j in the channel of streamer n. In other words, viewer m is interested in i rather than j. Through this objective, if $\hat{r}_{j,m,n}$ is incorrectly assigned with a greater value than $\hat{r}_{i,m,n}$, the value of the loss function increases, penalizing the current learning parameters. The cumulative loss is minimized by learning model parameters to optimize ranking performance.

To use CENTR to recommend the next-topics in the testing phase, given a candidate next-topic set S_v for streamer v, each $s \in S_v$ is first embedded into the latent space with a regression model. Its embedding is then concatenated with the latest $L-1$ live stream topics and CENTR derives the sequence embedding. The ranking score $r_{s,m}$ of s for each viewer $m \in U$ now can be calculated based on Eq. (8). Finally, the candidate next-topic in S_v with the greatest total ranking score is recommended since the greatest number of viewers are predicted to engage with it.

3.4 Collaborative Diffusion for Cold-start Streamers

While the number of streamers has rapidly increased from 3.3M to 9M since 2018[4], it is essential for the live stream platforms to accommodate new streamers by recommending proper live stream topics. However, the lack of streaming sequences of those new streamers makes the next-topic recommendation challenging. Inspired by meta-learning [25], CENTR trains a matching network based on warm-start sequences of those mature streamers (as shown in Figure 3(b)). The network then matches a cold-start sequence to a similar warm-start sequence and finally recommends the next-topic of the warm-start sequence for the cold-start sequence. Recall the illustration in Figure 2. CENTR novelly proposes to collaboratively diffuse the embeddings of warm-start sequences to those cold-start sequences of the newcomers, and this mechanism is expected to remedy the instability of the cold-start sequences.

To train the matching network, a warm-start sequence $\mathbf{AL}_{warm}^{(b)}$ encoded in the phase one by CENTR is formatted into a query pair $(\mathbf{AL}_{warm}^{(b)}, \mathbf{v}_{n,t+1})$, which shows the outcome next-topic at timestamp $t+1$ is $\mathbf{v}_{n,t+1}$. MECOS [25] is then employed to train an LSTM model as the matching network, which aims to extract sequential features for learning similarity metrics, based on the collection of the warm-start query pairs.

For a cold-start sequence $\mathbf{AL}_{cold}^{(b)}$, its query pair is formatted as $(\mathbf{AL}_{cold}^{(b)}, x)$, in which x denotes the next-topic waiting for recommendation. An intuition is to query MECOS with $\mathbf{AL}_{cold}^{(b)}$ and recommend the next-topic $\mathbf{v}'_{n,t+1}$ (i.e., $x = \mathbf{v}'_{n,t+1}$) of the most similar warm-start sequence $\mathbf{AL}'^{(b)}_{warm}$. Nonetheless, we argue that $\mathbf{AL}_{cold}^{(b)}$ could be too unstable to generate reliable and reasonable recommendation results in practice.

In view of this, based on the assumption that a viewer may watch similar channels, we novelly leverage the *collaborative diffusion mechanism* to strengthen the meaning of the cold-start sequences. Specifically, this mechanism aims at diffusing the embeddings of the warm-start sequences of the streamers that the viewers are also watching to the cold-start sequence. The formulation is designed as follows:

$$\mathbf{AL}'^{(b)}_{p,cold} = \mathbf{AL}^{(b)}_{p,cold} + \lambda \cdot \mathbf{AL}^{(b)}_{q,warm}, \tag{9}$$

where λ is a weighting parameter. The warm-start streamer q is sampled based on the joint probability $P(l_{p,m}) \cdot P(l_{q,m})$. Therefore, if a mediate viewer m is loyal to the cold-start streamer p and a warm-start streamer q, the sequence embedding of q is more likely to be considered for refining $\mathbf{AL}^{(b)}_{p,cold}$. Note that the viewer interest topic embedding of viewer m has already been incorporated in $\mathbf{AL}^{(b)}_{p,cold}$ by Eq. (5). Therefore, $\mathbf{AL}'^{(b)}_{p,cold}$ is more informative and ready to be fed into MECOS for sophisticated similarity calculation.

4 EXPERIMENTAL RESULTS

4.1 Experiment Setup and Evaluation Plan

4.1.1 Evaluation Plans and Metrics. We design the experiment to answer the following research questions:

RQ1: Does CENTR outperform existing sequential/video recommender systems in LSNR?
RQ2: What is the influence of each novel component in CENTR?
RQ3: How is the performance of CENTR in cold-start scenario?
RQ4: How is the convergence of CENTR?
RQ5: What is the effect of loyalty in real cases?
RQ6: How to set hyperparameters, the length of sequences and the threshold separating warm-start and cold-start streamers?

For evaluating the ranking performance, top-K precision (Pre@K), recall (Rec@K), and F1 scores (F1@K) are presented. We also show the value of normalized loss along epochs in training sets to observe the convergence. Finally, a case study of the effect of the loyalty in LCA is presented.

4.1.2 Datasets and Pre-processing. 1) *Douyu* collects 242K channels, 7M viewers, and 64.9M donations (in terms of virtual gifts) from the most popular live stream platform Douyu in China. 2) *Twitch* includes 600K viewers, 140 channels, and 86K donations. 3) *Bilibili* is a danmu (i.e., time-synchronized comments) platform. While watching pre-recorded videos on Bilibili, viewers are able to leave their danmus at a certain timestamp as demanded. Whenever the other viewers watch the videos, those danmus pop up on the screen at those timestamps. Viewers therefore interact with each other in a semi-live streaming way. On the other hand, content providers (i.e., streamers) are able to learn from the danmus and improve their content topics accordingly.

Table 1: Top-K recommendation performance comparisons on real live stream datasets ($K = 1, 5$).

		Baseline Methods					CENTR and Variations				
Dataset	Metric	BERT4Rec	SASRec	UVCAN	SHAN	SelCA	CENTR	UNI	α-0.5	NoLVA	NoDual
Douyu	Pre@1	0.361	0.360	0.250	0.346	0.360	**0.459**	0.373	0.401	0.391	0.370
	Rec@1	0.253	0.281	0.218	0.228	0.240	**0.331**	0.260	0.319	0.322	0.270
	Pre@5	0.432	0.431	0.371	0.401	0.411	**0.609**	0.522	0.567	0.565	0.506
	Rec@5	0.301	0.322	0.280	0.300	0.270	**0.422**	0.383	0.408	0.399	0.372
Twitch	Pre@1	0.310	0.299	0.219	0.301	0.263	**0.387**	0.322	0.332	0.327	0.310
	Rec@1	0.259	0.256	0.203	0.252	0.199	**0.283**	0.245	0.260	0.273	0.261
	Pre@5	0.436	0.410	0.303	0.418	0.355	**0.572**	0.503	0.559	0.499	0.473
	Rec@5	0.300	0.289	0.257	0.277	0.211	**0.379**	0.351	0.359	0.351	0.344
Bilibili	Pre@1	0.268	0.269	0.212	0.255	0.263	**0.293**	0.269	0.275	0.270	0.269
	Rec@1	0.212	0.228	0.153	0.227	0.225	**0.233**	0.211	0.230	0.218	0.219
	Pre@5	0.322	0.323	0.281	0.317	0.324	**0.389**	0.355	0.366	0.365	0.340
	Rec@5	0.290	0.287	0.207	0.266	0.265	**0.359**	0.307	0.320	0.307	0.298

We follow the same preprocessing procedures from [9]. The donations in *Douyu* and *Twitch* and the danmus in *Bilibili* are treated as implicit feedback. The timestamps are used to form sequences. The viewers with fewer than 5 actions and the streamers with fewer than 5 live stream sessions are excluded. The streamers with 5 to 15 sessions are classified as cold-start streamers, and those with more than 15 sessions are warm-start streamers. The above default thresholds are determined by experimental results in Section 4.4. For each warm-start streamer, their sequences are split into three parts: 1) the latest sequence for testing, 2) the second latest sequence for validation, and 3) the remaining sequence for training. The sequences of cold-start streamers are only used in testing for cold-start evaluation proposes.

4.1.3 Comparing Methods. To evaluate the next-topic recommendation for streamers, we compare CENTR with state-of-the-art sequential models and video recommender systems:

- **BERT4Rec [15]:** The first bidirectional self-attentive sequential recommender system.
- **SASRec [9]:** A transformer-based recommender system learning sequential relevance in sparse datasets.
- **UVCAN [11]:** A co-attention video reommender system that extracts both viewer interests and video targets as features.
- **SHAN [20]:** A session-based hierarchical attentive model that mixes the long-term viewer preferences and the short-term dynamic.
- **SelCA [8]:** A transformer-based sequential recommender system with topic modeling-based category embedding to exploit global information.
- **CENTR:** The proposed method conducting LCA ($\alpha = 1$) and LVA to capture the co-evolution between viewer interests and live stream topics. A collaborative diffusion mechanism improves the meta-learning block for cold-start scenario.

For ablation studies, we compare with multiple variations of CENTR:

- **UNI:** CENTR sampling viewers randomly from uniform distribution without considering the loyalty.
- **α-0.5:** CENTR using LCA with $\alpha = 0.5$, which increases the probability of sampling disloyal viewers.
- **NoLVA:** CENTR excluding LVA, meaning that it ignores the influence from live stream topics to viewer interests.
- **NoDual:** CENTR training the two sequences individually (i.e., without using LCA and LVA).

For cold-start evaluation, in addition to the above methods, we compare with the following methods:

- **MECOS [25]:** A basic meta learning-based model that employs an LSTM model to learn the similarity between sequences.
- **NoMETA:** CENTR without using the meta-learning block predicts actions of cold-start sequences with Eq. (8).

4.1.4 Implementation Details. CENTR is built on TensorFlow with the Adam optimizer. For fair comparisons, all the transformer-based models stack two self-attention layers and adopt the positional embeddings [9]. The hyperparameters are determined based on five-fold cross-validation. The training process terminates until the performance does not improve for 20 epochs in validation. The batch size, the learning rate, and the dropout rate are set as 128, 0.001, and 0.5 respectively. The dimension of the embeddings is set to 40. The sequences length is set to 5 for all datasets. The number of sampled viewers c in LCA is 30 and the negative sampling ratio for ranking is 5. The default weighting parameter λ of the collaborative diffusion is set to 0.1. Finally, to eliminate bias, each result presented in the following is the average result from at least 50 repetitions.

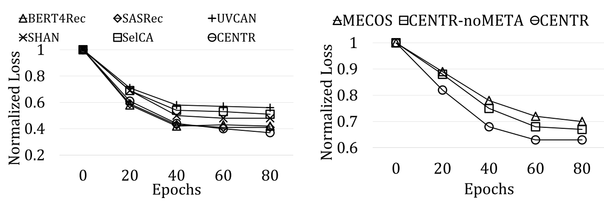
4.2 Recommendation Results

4.2.1 Performance Tests. To answer RQ1, Table 1 presents the performance results in terms of Pre@1, Rec@1, Pre@5, and Rec@5 in *Douyu*, *Twitch*, and *Bilibili* datasets. The differences of the recommendation results between CENTR and the others are statistically significant (p-value $\leq 0.008 < 0.05$). The improvements of CENTR manifest that CENTR outperforms the state-of-the-art approaches by at least 27.1% among those metrics in *Douyu*. It is because CENTR introduces the *Co-evolutionary Sequence Embedding Structure* and the *dual self-attention* that learn the interplay of viewer interests topics and live stream topics from time to time, and precisely model the *co-evolution* in live stream platforms. In contrast, partially considering the evolution of viewer interests like BERT4Rec, SASRec, SHAN, and SelCA leads to worse recommendation results.

Although UVCAN considers both the viewer interests and the video topics, it decouples the temporal relations between any pair

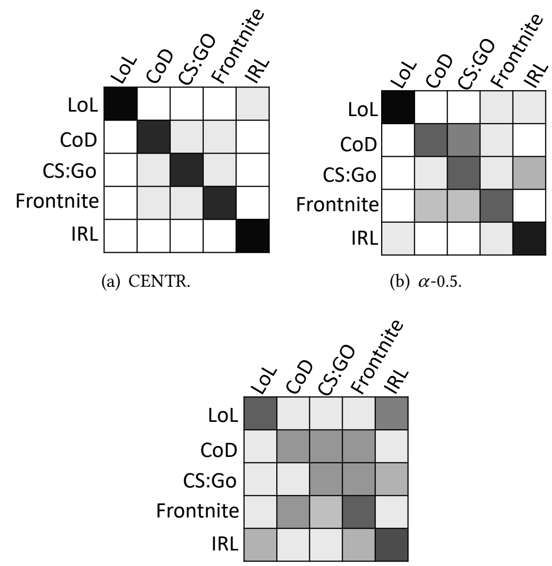
Table 2: Top-K recommendation performance comparisons on cold-start Douyu dataset ($K = 1, 5$).

	Baseline Methods						CENTR and Variations				
Metric	BERT4Rec	SASRec	UVCAN	SHAN	SelCA	MECOS	CENTR	UNI	NoLVA	NoDual	NoMETA
Pre@1	0.182	0.180	0.166	0.176	0.173	0.155	**0.211**	0.183	0.188	0.183	0.180
Rec@1	0.134	0.152	0.116	0.117	0.120	0.137	**0.170**	0.128	0.152	0.140	0.144
Pre@5	0.209	0.208	0.183	0.185	0.180	0.178	**0.257**	0.220	0.231	0.209	0.207
Rec@5	0.188	0.191	0.157	0.167	0.156	0.187	**0.221**	0.180	0.181	0.165	0.170



(a) Warm-start scenario.
(b) Cold-start scenario.

Figure 4: Convergence comparisons in Douyu dataset.



(a) CENTR.
(b) α-0.5.
(c) UNI.

Figure 5: Live stream topic transition under different sampling strategies. X and Y axes represent the next-topic and the lastest topic respectively.

of videos even if they are made by the same producer. Moreover, this setting also eliminates the causal relations from the viewer interests to the next-topic determination. Both of them ultimately leads UVCAN to the worst performance among all the compared methods (7.1%-31.6% lower), indicating the crucial role of the temporal relations in determining live stream topics. On the other hand, the improvement of CENTR in *Bilibili* dataset is relatively small (9.3%-22.3%). It is because danmus can be added after the next video is uploaded so they are not considered while producing the next video. Therefore, the post-added danmus (or the viewers generating them) could mislead the learning process of recommenders.

4.3 Case Study

4.3.1 Ablation Tests. For RQ2, Table 1 also presents the recommendation performance of the ablation tests of CENTR. The results show that excluding the influence from viewers to streamers (UNI) degrades the performance more than excluding the opposite way of influence (NoLVA), pointing out that live stream is a viewer-driven industry. For the convenience of comparisons, UNI can be deemed to adopt a extremely small value of α (e.g., 0.001). As α increases, the performance of all metrics also raises. This gives credit to the proposed *LCA* that successfully makes CENTR concentrate on those loyal viewers more when α grows, and those loyal viewers may contribute more on deciding the next-topics for streamers. In contrast, the improvement of UNI compared to other transformer-based methods is relatively insignificant (-0.1%-10.2%), showing that sampling randomly is not helping.

On the other hand, the improvement of CENTR compared to UNI in *Douyu* (24.4%) dataset is more significant than in *Twitch* (20.1%) and *Bilibili* (8.9%) datasets in terms of Pre@1. It is because *Douyu*, the largest live stream platform in China, has the most random viewers which is at least 13 times of the other two datasets. In this case, it is easier to sample disloyal viewers when α decreases. In other words, putting more emphasis on those loyal viewers in a huge live stream platform could be important in making accurate next-topic recommendations. Last but not least, NoLVA excluding the influence from live stream topics to viewer interests degrades the performance by 5.7% to 17.4% from CENTR. Overall, the results of the ablation analysis manifests that factoring in both directions of influence with LCA and LVA are effective and necessary.

4.3.2 Cold-start Performance. Table 2 compares the ranking performance of each method in the cold start scenario in *Douyu* dataset to answer RQ3. BERT4Rec, SASRec, UVCAN, SHAN, SelCA, and NoMETA do not employ the meta-learning blocks, so they make predictions for the cold-start streamers based on the ranking scores directly. Among all the methods, CENTR outperforms the others by at least 15.9% of all metrics. Note that implementing the meta-learning block is not the only reason of the improvement in cold-start scenario, because the transformer-based models BERT4Rec and SASRec outperform MECOS as well. Beyond that, CENTR integrates both the benefit of co-evolutionary process and the meta learning technique to achieve the best performance. Furthermore, CENTR leverages the *collaborative diffusion mechanism* in Eq. (9) to respectively surpass the performance of NoMETA and MECOS by at least 17.2% and 18.2% of all metrics. The collaborative diffusion mechanism effectively supplements the embeddings of cold-start streamers by adding side information from the warm-start streamers who share the same viewers.

4.3.3 Convergence. For RQ4, Figure 4(a) compares the convergence of each recommender system in the warm-start scenarios in *Douyu* dataset. CENTR converges at a better position after 60 epochs, while the others converge at around 40-50 epochs but in worse positions. This is because CENTR has a more complex architecture that includes two different and mutually influencing sequences, and hence it suffers from the trade-off between computational effort and accuracy. Figure 4(b) shows the convergence of each method using meta-learning blocks only, for fair comparisons. CENTR converges at a better position than MECOS, manifesting that the co-evolutionary modeling is also helpful in the cold-start scenario. Moreover, introducing the collaborative diffusion mechanism does not generate too much computational overhead but brings great benefits in convergence. The total training time of CENTR with respect to warm-start and cold-start cases are 6 hours and 1.3 hours. The runtime of recommending the next-topic is no more than 27.1 seconds. Given that most of the live streaming sessions on Twitch usually last 2-4 hours and most of the streamers go live no more than once per day, CENTR is applicable for real-world usage.

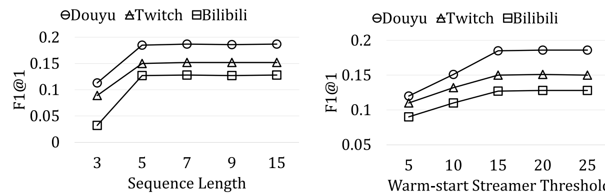
For RQ5, Figure 5 presents the live stream topic transition prediction of CENTR, α-0.5, and UNI. The observed topics are conditional to the five most popular ones on $Twitch$. League of Legend (LoL) is a strategic game; Call of Duty (CoD), Frontnite, and CS:GO are different shooting games; In-Real-Life (IRL) is a non-gaming topic that streamers share their real lives with viewers. A darker grid indicates that, given the latest topic (Y-axis), more viewers are predicted to like the next-topic (X-axis).

In Figure 5(a), CENTR samples more loyal viewers when $\alpha = 1$. While those loyal viewers often watch the same topics of their favorite streamers, so CENTR tends to stick to the original topics for recommendations. When α decreases (as shown in Figures 5(b) and 5(c)), grey grids increase since more disloyal viewers are sampled. Their diverse interests of topics are considered during learning, and hence make α-0.5 and UNI recommend different topics other than the original one. In this way, CENTR is able to provide flexibility for streamers. Specifically, when a streamer is aiming at exploring new topics or enlarging her viewership, CENTR can offer diverse recommendation results by lowering α.

For UNI in Figure 5(c), the recommended next-topics of CoD, Frontnite, and CS:GO are still in this shooting game category in a relatively high probability. It is because these three topics share many overlapping viewers. Therefore, a sampled disloyal viewer of a shooting game (e.g., CoD) could be a loyal viewer of another shooting game (e.g., Frontnite or CS:GO). In contrast, the viewer groups of IRL is isolated from the other gaming topics since its non-gaming topic is unique. Although UNI samples many disloyal viewers with low α, the other gaming topics are rarely recommended.

4.4 Sensitivity Tests on Hyperparameters

For RQ6, Figure 6 presents the ranking performance in terms of F1@1 of CENTR in the three datasets. F1@1 is measured because it is a balance metric of both precision and recall. For the sequence length in Figure 6(a), when the sequence is shorter than five, the



(a) Sequence length. (b) Warm-start threshold.

Figure 6: F1@1 scores of different hyperparameters.

performance drops because they are insufficient to learn the sequential patterns of co-evolution. When the length grows larger than five, the performance does not improve significantly. Therefore, we found that setting the sequence length to five is a good choice. The length in this problem is shorter than conventional sequential recommender systems (e.g., 15 or more [9, 15]) manifesting that long-term effects are less significant in live streams. The reason could be that streamers can cultivate viewer reactions easily on live streams and accordingly adjust their live stream topics quickly.

For the threshold discriminating warm-start and cold-start streamers in Figure 6(b), when the threshold is lower than 15, streamers with a small number of live sessions are classified as warm-start streamers. However, their training instances in terms of sequences are fewer and hence deteriorate the performance of CENTR. In contrast, when the threshold is more than 15, the performance improves insignificantly. This result manifests that increasing the threshold to more than 15 does not help since the streamers having more than 15 sessions are excluded from training CENTR. Consequently, we have experimentally found that using 15 as the threshold in our datasets is a good choice.

5 CONCLUSION

To our best knowledge, there exists no prior research that studies the next-topic recommendation problem in live stream platforms, not to mention incorporating the unique co-evolutionary phenomenon between viewers and streamers. We formulate a new research problem, namely LSNR, and propose a novel two-phase framework CENTR. The first phase introduces the Co-evolutionary Sequence Embedding Structure and the dual self-attention to learn the co-evolutionary phenomenon of live stream topics and viewer interest topics. The mutual effect are modeled by two novel loyalty-based modules, LCA and LVA, which enhance both flexibility and the efficiency by using viewer loyalty. To facilitate cold-start recommendations for new streamers in the second phase, a collaborative diffusion mechanism is implemented to improve the meta learner. Through experiments on real datasets, CENTR outperforms state-of-the-art methods in both regular and cold-start scenarios. An empirical case study manifests that CENTR is flexible in recommendations for multiple purposes by leveraging viewer loyalty.

ACKNOWLEDGMENT

This work is supported in part by NSF under grants III-1763325, III-1909323, III-2106758, and SaTC-1930941.

REFERENCES

[1] Da Cao, Xiangnan He, Lianhai Miao, Yahui An, Chao Yang, and Richang Hong. 2018. Attentive group recommendation. In *SIGIR*.

[2] Jingyuan Chen, Hanwang Zhang, Xiangnan He, Liqiang Nie, Wei Liu, and Tat-Seng Chua. 2017. Attentive collaborative filtering: multimedia recommendation with item- and component-level attention. In *ACM SIGIR*.

[3] Ziwei Fan, Zhiwei Liu, Yu Wang, Alice Wang, Zahra Nazari, Lei Zheng, Hao Peng, and Philip S. Yu. 2022. Sequential Recommendation via Stochastic Self-Attention. In *WWW*.

[4] Claudia Flores-Saviaga, Jessica Hammer, Juan Pablo Flores, Joseph Seering, Stuart Reeves, and Saiph Savage. 2019. Audience and Streamer Participation at Scale on Twitch. In *ACM HT*.

[5] Junyu Gao, Tianzhu Zhang, and Changsheng Xu. 2017. A Unified Personalized Video Recommendation via Dynamic Recurrent Neural Networks. In *ACM MM*.

[6] Zhankui He, Handong Zhao, Zhe Lin, Zhaowen Wang, Ajinkya Kale, and Julian J. McAuley. 2021. Locker: Locally Constrained Self-Attentive Sequential Recommendation. In *ACM CIKM*.

[7] Adele Lu Jia, Yuanxing Rao, Hongru Li, Ran Tian, and Siqi Shen. 2020. Revealing Donation Dynamics in Social Live Video Streaming. In *Companion of WWW*.

[8] Kyeongpil Kang, Junwoo Park, Wooyoung Kim, Hojung Choe, and Jaegul Choo. 2019. Recommender System Using Sequential and Global Preference via Attention Mechanism and Topic Modeling. In *ACM CIKM*.

[9] Wang-Cheng Kang and Julian J. McAuley. 2018. Self-Attentive Sequential Recommendation. In *IEEE ICDM*.

[10] Hsu-Chao Lai, Jui-Yi Tsai, Hong-Han Shuai, Jiun-Long Huang, Wang-Chien Lee, and De-Nian Yang. 2020. Live Multi-Streaming and Donation Recommendations via Coupled Donation-Response Tensor Factorization. In *ACM CIKM*.

[11] Shang Liu, Zhenzhong Chen, Hongyi Liu, and Xinghai Hu. 2019. User-Video Co-Attention Network for Personalized Micro-video Recommendation. In *WWW*.

[12] Zhicong Lu, Haijun Xia, Seongkook Heo, and Daniel Wigdor. 2018. You watch, you give, and you engage: a study of live streaming practices in china. In *ACM CHI*.

[13] Keri Mallari, Spencer Williams, and Gary Hsieh. 2021. Understanding Analytics Needs of Video Game Streamers. In *ACM CHI*.

[14] Ihsan Mert Ozcelik and Cem Ersoy. 2021. Low-Latency Live Streaming Over HTTP in Bandwidth-Limited Networks. *IEEE Commun. Lett.* (2021).

[15] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer. In *ACM CIKM*.

[16] Chen Wang, Jianfeng Guan, Tongtong Feng, Neng Zhang, and Tengfei Cao. 2019. Bitlat: bitrate-adaptivity and latency-awareness algorithm for live video streaming. In *ACM MM*.

[17] Dennis Wang, Yi-Chieh Lee, and Wai-Tat Fu. 2019. I Love the Feeling of Being on Stage, but I Become Greedy: Exploring the Impact of Monetary Incentives on Live Streamers' Social Interactions and Streaming Content. *Proc. ACM Hum. Comput. Interact.* (2019).

[18] Yu Wang, Hengrui Zhang, Zhiwei Liu, Liangwei Yang, and Philip S. Yu. 2022. ContrastVAE: Contrastive Variational AutoEncoder for Sequential Recommendation. In *ACM CIKM*.

[19] Donghee Yvette Wohn, Guo Freeman, and Caitlin McLaughlin. 2018. Explaining viewers' emotional, instrumental, and financial support provision for live streamers. In *ACM CHI*.

[20] Haochao Ying, Fuzhen Zhuang, Fuzheng Zhang, Yanchi Liu, Guandong Xu, Xing Xie, Hui Xiong, and Jian Wu. 2018. Sequential Recommender System based on Hierarchical Attention Networks. In *IJCAI*.

[21] Dung-Ru Yu, Chiao-Chuan Chu, Hsu-Chao Lai, and Jiun-Long Huang. 2020. Social Attentive Network for Live Stream Recommendation. In *Companion of WWW*.

[22] Sanshi Yu, Zhuoxuan Jiang, Dongdong Chen, Shanshan Feng, Dongsheng Li, Qi Liu, and Jinfeng Yi. 2021. Leveraging Tripartite Interaction Information from Live Stream E-Commerce for Improving Product Recommendation. In *ACM KDD*.

[23] Shuai Zhang, Hongyan Liu, Jun He, Sanpu Han, and Xiaoyong Du. 2021. Deep sequential model for anchor recommendation on live streaming platforms. *Big Data Mining and Analytics* (2021).

[24] Shuai Zhang, Hongyan Liu, Lang Mei, Jun He, and Xiaoyong Du. 2022. Predicting viewer's watching behavior and live streaming content change for anchor recommendation. *Appl. Intell.* (2022).

[25] Yujia Zheng, Siyi Liu, Zekun Li, and Shu Wu. 2021. Cold-start Sequential Recommendation via Meta Learner. In *AAAI*.