

General Item Representation Learning for Cold-start Content Recommendations

Jooeun Kim
kje980714@snu.ac.kr
Seoul National Univ.
Seoul, Korea

Jinri Kim
ruth9811@snu.ac.kr
Seoul National Univ.
Seoul, Korea

Kwangeun Yeo
kwangeun.yeo@snu.ac.kr
Seoul National Univ.
Seoul, Korea

Eungi Kim
kuman5262@snu.ac.kr
Seoul National Univ.
Seoul, Korea

Kyoung-Woon On
kloud.ohn@kakaobrain.com
Kakao Brain
Sungnam, Korea

Jonghwan Mun
jason.mun@kakaobrain.com
Kakao Brain
Sungnam, Korea

Joonseok Lee
joonseok@snu.ac.kr
Seoul National Univ.
Seoul, Korea

ABSTRACT

Cold-start item recommendation is a long-standing challenge in recommendation systems. A common remedy is to use a content-based approach, but rich information from raw contents in various forms has not been fully utilized. In this paper, we propose a domain/data-agnostic item representation learning framework for cold-start recommendations, naturally equipped with multimodal alignment among various features by adopting a Transformer-based architecture. Our proposed model is end-to-end trainable completely free from classification labels, not just costly to collect but suboptimal for recommendation-purpose representation learning. From extensive experiments on real-world movie and news recommendation benchmarks, we verify that our approach better preserves fine-grained user taste than state-of-the-art baselines, universally applicable to multiple domains at large scale.

KEYWORDS

cold-start, recommendation, content-based, transformer, multimodal

Reference Format:

Jooeun Kim, Jinri Kim, Kwangeun Yeo, Eungi Kim, Kyoung-Woon On, Jonghwan Mun, and Joonseok Lee. 2024. General Item Representation Learning for Cold-start Content Recommendations.

1 INTRODUCTION

Recommendation systems are widely adopted for a variety of real-world applications, *e.g.*, online retails, video sharing platforms, and more, as the scale of items that people may choose from has been rapidly growing. Collaborative filtering (CF) [8, 20], recognizing preference patterns observed in user-item interactions, has been successfully applied to personalized recommendation systems to provide potentially preferred items in a personalized manner.

Despite its success, CF approaches suffer from several challenges, one of which is the cold-start problem. Since CF relies only on user

and item interaction, it is not capable of generating personalized recommendations for a new user without any records. Likewise, a brand-new item with no user feedback cannot be recommended to the right customers who are most likely to prefer that item.

Cold-start is actually a common problem in modern recommendation systems. On YouTube, for example, 500 hours of contents are being uploaded every minute [29]. With a CF recommendation system, fresh contents can only be recommended to some random users until sufficient interaction data is collected. Another example is Netflix, where new movies or TV series often compete for a limited main advertisement space. It is important for the supplier to select users who will most likely enjoy the new contents to maximize its revenue, where cold-start item recommendation plays a key role in selecting the right set of users for each fresh content without any user feedback. Another domain that the cold-start is important is news articles. Unlike multimedia contents that their value lasts for a long time, news contents are useful only for a short period. In other words, it is more important to recommend news articles to the right people before we collect sufficient activities on them, and thus cold-start is the key in this domain.

To tackle the cold-start problem, side information about the users or items has been utilized. Since content information becomes available at the time of release, it is possible to retrieve a set of neighboring items that are of similar content, and it may be recommended to users who like this kind of items. Traditional approaches [41, 49, 59, 79] used demographic information of the users or meta-data of the items, *e.g.*, genre or artist, to get prior knowledge of them. With recent advances in deep learning, extracting semantics from the raw content, *e.g.*, videos [4, 9, 61] or music [27, 31, 51], has become pervasive. Some recent works attempt to learn more powerful item representations for recommendation purpose, directly from the raw contents [37, 38, 67].

Here, we pose the key question: is this rich content information being properly and sufficiently utilized for cold-start recommendations? From two observations below, we believe it is still limited.

First, most existing methods are specifically designed for a particular dataset on a specific domain. This is probably because the “content information”, by nature, varies depending on the domain, service, or dataset. For instance, movie contents densely provide visual frames, while music contents are mainly sound tracks. A visual-signal-based movie recommendation model will not work

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

KDD'24, Aug 25 – 29, 2024, Barcelona, Spain

© 2024 Association for Computing Machinery.

ACM ISBN 978-1-4503-XXXX-X/18/06...\$15.00

for music due to the absence of visual signals. The opposite, a sound-track-based music recommendation model may also not work well for movies, as audio in the movies is not as dense and informative as in music. For this reason, content-based recommendation models have been developed independently for each domain and dataset, without having a general, common framework like CF.

Second, in spite of the advances in deep learning, raw multimedia contents are still not easily utilized due to their high cost of training and serving. Specifically, it is a common practice to learn multimedia representations on large, human-labeled classification data, e.g., JFT [58], HowTo100M [48] or Kinetics [9]. Setting aside the high cost of collecting a large volume of examples and labeling them, we claim that feature encoding learned from classification labels is sub-optimal for recommendations. Classifying a sample to a predefined category intrinsically forces the model to learn only the common aspects within each class, ignoring subtle differences among individual examples in the same class. We hypothesize that an encoding learned from such a classifier does not sufficiently preserve fine-grained details that are necessary and useful for a recommendation model to distinguish subtle preference of individual users on a variety of items.

In this paper, we seek a general item content representation learning framework which is domain and dataset-agnostic, and preferably, which does not rely on a human-labeled classification dataset. In order to overcome the aforementioned problems, we propose to utilize the Transformer [63] architecture, which has been the basis of most state-of-the-art models in recent language [15, 47], image [17, 43], video [4, 7] and audio [21] understanding. Transformer architecture is particularly appropriate for our use case, since it applies in most steps a common architecture to the input sequence regardless of its nature, once each input token is mapped to an embedding simply by a linear mapping. In other words, its input-type-specific part is light, making it more appropriate for a general feature extractor. This is in contrast to a CNN-based image model or an RNN-based text model, where the architecture-specific representations are kept until the very last few layers. Thanks to its nature to rely less on data modality, it also provides a natural way of multimodal fusion [2, 45, 57]. Our proposed framework is trained end-to-end solely on user activities, e.g., clicking or rating, without pre-training on classification labels on a predefined set.

To verify effectiveness and generalizability of the proposed approach, we conduct extensive cold-start recommendation experiments on multiple domains, where the multimodal content signals are particularly rich (movie) and where the cold-start recommendation is particularly important (news). From experiments, we demonstrate that the content representations learned by our general framework perform significantly better recommendations than existing methods, preserving finer subtleties about items. Note that we focus only on the cold-start recommendations, where the content signals are required and play the key role. Combining this with a CF-based model for warm-start cases will be an interesting extension, but this is beyond the scope of this paper.

Our main contributions are summarized as follows:

- We propose a *domain/dataset-agnostic* item content representation learning framework for cold-start recommendations, effectively fusing *multimodal signals*.

- Our framework is *end-to-end trainable*, without relying on human-labeled large-scale classification data to train modality-specific encoders. Trained solely on user activities, our item representations better preserve *fine-grained taste* of users.
- From extensive experiments, we demonstrate that our proposed approach achieves state-of-the-art performance on cold-start recommendation on large-scale datasets from multiple domains.

2 RELATED WORK

2.1 Cold-start Recommendations

Collaborative Filtering (CF) has been successful in personalized recommendation systems with the existence of plentiful historical data [25, 40, 52, 54, 56, 78], but the cold-start problem is its long-standing challenge, where no historical interaction record of user or item exists. To tackle this problem, MWUF [82] warms up cold items with meta-scaling and shifting networks. DropoutNet [64] randomly drops items or users to make the model better adapt to cold-start. Heater [83] tackles the problem with a randomized training mechanism and mixture-of-experts transformation. Recently, meta-learning approaches [16, 36, 46, 50, 77] are proposed to tackle cold-start recommendation.

2.2 Content-based Recommendations

Auxiliary information like content features has been integrated to CF models to alleviate the cold-start problem. CB2CF [6] connects the gap between item content and their CF representations. CWH [5] balances the quality of warm and cold items by utilizing text-based item content. CLCRec [67] maximizes the mutual dependencies between item content and collaborative signals using contrastive learning. CLCRec shares a common theme with our model in that it utilizes multimodal content features to tackle cold-start recommendation. However, it trains embeddings on image classification labels and transfers them to the recommendation task, while our framework is completely free from human labels. More recently, CVAR [79] adopts conditional variational autoencoders to warm up cold item embeddings using content metadata. Recently, graph neural networks (GNN) become increasingly prevalent in recommender systems. PMGT [76], for example, combines GNN with multimodal side information in item recommendation [42]. There are more examples, e.g., DUIF [19], MTPR [18], CC-CC [55], MMGCN [68], and Movie Genome [12]. See a survey [13] for more.

CDML [38] is another model that proves usefulness of audio-visual features in cold-start scenario. GCML [37], learns video embeddings from a relational graph. However, both models are not personalized in that they learn item-item co-watch similarity aggregated over all users, not at individual user level. On the other hand, our model explicitly uses individual user feedback to learn the item representations.

Although many content-based approaches tackle cold-start, they are restricted to a particular domain and features specific to the target dataset, often trained on classification labels. Our approach, on the other hand, is applicable to arbitrary domains and features, and is end-to-end trainable free from human-labeled data.

2.3 News Recommendations

News recommendation models particularly exploit content features to tackle the item cold-start problem [34, 65, 75], since news articles are replaced with new ones in a short period of time. For instance, NRMS [73] and NPA [71] learn article representations from news titles. TANR [72], LSTUR [3], and NAML [70] utilize news topics or article bodies in addition to titles to enrich the news representations. Most existing models exploit textual modality to represent news articles [3, 70, 72–74], but recently, visual modality is also considered [75]. Along with this trend, our framework supports arbitrary number of content features in various forms, including visual and textual. Recently, visual modality is also considered [75]. Along with this trend, our framework supports arbitrary content features, including visual and textual.

2.4 Contrastive Learning

Contrastive learning is a self-supervised task, learning to discriminate which pairs of data points are similar and different from the dataset, widely used in computer vision and NLP [10, 11, 22, 24, 26, 32]. Recent works employ contrastive learning in recommender systems to optimize the user and item representations. For instance, Liu et al. [44] proposes a graph contrastive learning to alleviate the sample bias. CLRec [80] employs it to improve DCG in recommendation. SLMRec [60] incorporates contrastive learning into multimedia recommendation with a graph neural network. Our method also employs contrastive loss for rating prediction and multi-modal alignment, detailed in Sec. 5.

3 PROBLEM FORMULATION AND NOTATIONS

In this paper, we presume implicit feedback from the users, so there are only two types of ratings: *preferred* and *unknown*. Given a binary preference matrix $\mathbf{R} \in \{0, 1\}^{M \times N}$ with M users and N items, an element $\mathbf{R}_{ij} = 1$ indicates that the user i prefers the item j , while $\mathbf{R}_{ij} = 0$ means unknown. The matrix \mathbf{R} can be split into two parts: \mathbf{R}_w with warm items and \mathbf{R}_c with cold items, where all entries within \mathbf{R}_c are zeros. The cold-start recommendation task is predicting preferable items within \mathbf{R}_c ; in other words, retrieving a list of items that each user i may prefer among the cold items.

Each item is provided with a set of C content attributes. The content information for each attribute $c = 1, \dots, C$ is denoted by $\mathbf{X}^{(c)} \in \mathbb{R}^{N \times D_c}$, where D_c is the dimensionality of the content information for the attribute c . Depending on its nature (modality), D_c may be in a structured form. For an image (e.g., a raw frame), for instance, $D_c = H_c \times W_c \times 3$, where H_c and W_c are the height and width of the image. For a video, $D_c = T_c \times H_c \times W_c \times 3$, where T_c is the maximum number of frames in the video. For a textual modality (e.g., synopsis), $D_c = \{1, \dots, |V|\}^{T_c}$, where T_c is the maximum length of the text for c and V is the vocabulary set. The content information for c of a particular item j is denoted by $\mathbf{X}_j^{(c)} \in \mathbb{R}^{D_c}$.

We tackle cold-start items only, not cold-start users, since no public dataset provides meaningful user side information due to privacy, although cold-start users can be modeled in a similar way.

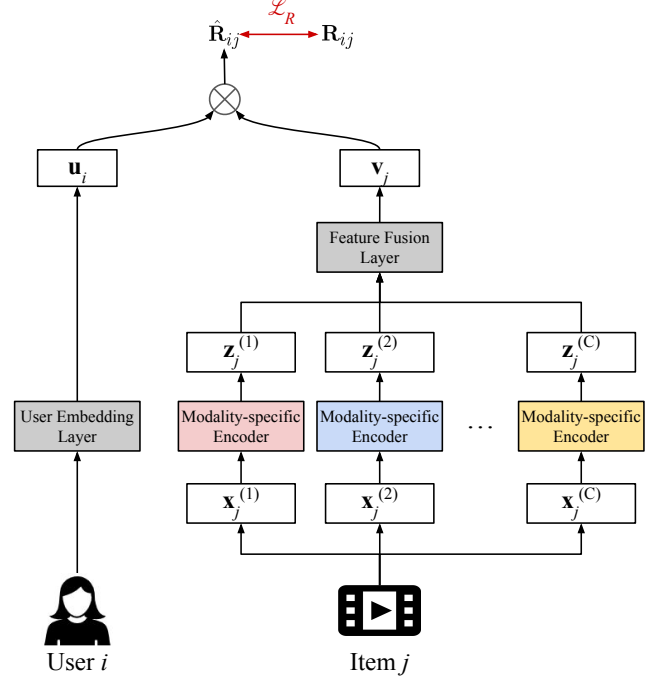


Figure 1: Overall Architecture. C content features are extracted for each item using modality-specific encoders. (A few examples are illustrated in Fig. 2.) Then, the Feature Fusion Layer aggregates them into the final item representation \mathbf{v}_j , and the rating \mathbf{R}_{ij} is predicted by taking dot product with the target user embedding \mathbf{u}_i , learned in the manner of collaborative filtering.

4 PRELIMINARY

We briefly review Transformers [63], on which our general item representation learning framework is built. Transformer is a powerful model that achieves state-of-the-art performance on sequence-to-sequence tasks [45] like machine translation as well as general representation learning for images [17] and videos [4]. Taking as input a sequence of its sub-component (e.g., words for a sentence, smaller patches for an image, and frames for a video), it applies a self-attention mechanism in an encoder-decoder structure to learn context by tracking relationships among those sub-components. We first describe the Transformer encoder in detail, followed by how it is utilized for two important modalities: text and visual. The decoder is not used in our framework.

4.1 Transformer Encoder

Recall that a side information c for an item j is denoted by $\mathbf{X}_j^{(c)}$. To be uncluttered, we omit c and j whenever clear. \mathbf{X} is split into a sequence of T sub-components, denoted by $\{\mathbf{x}_{[1]}, \dots, \mathbf{x}_{[T]}\}$, and how to split varies by modalities. Some modalities like a video or a sentence are sequential in nature. An image may be split into multiple smaller patches [17].

Given this sequence $\{\mathbf{x}_{[1]}, \dots, \mathbf{x}_{[T]}\}$ of T tokens as input, they are first embedded into vectors, $\mathbf{Z} \equiv \{\mathbf{z}_{[1]}, \dots, \mathbf{z}_{[T]}\}$, where $\mathbf{z}_{[t]} \in \mathbb{R}^d$

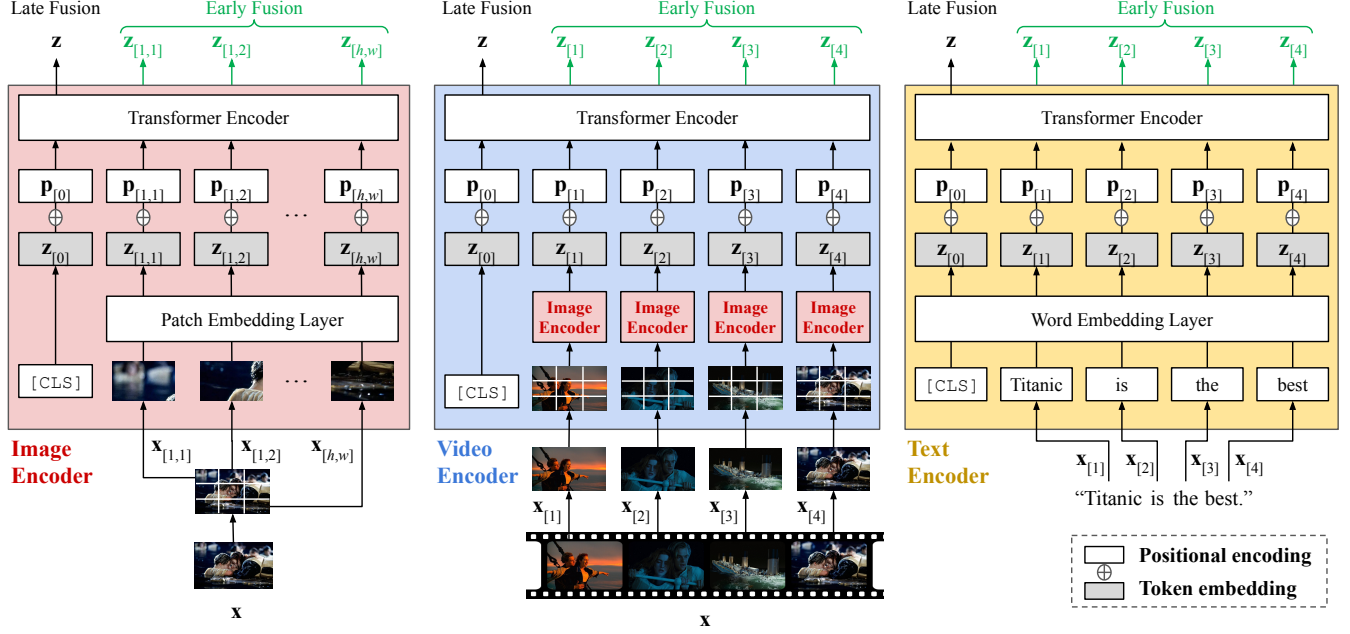


Figure 2: Examples of Modality-specific Encoders. From the left, we illustrate the image, video, and text encoders.

and d is the token embedding size. Then, Z is fed to a series of encoder blocks, where each block is composed of a self-attention layer and a feed-forward network, which enrich token representations with contextual information from other tokens in the sequence.

First, the token embeddings $Z \in \mathbb{R}^{T \times d}$ are transformed to three special representations, namely, query ($Q \in \mathbb{R}^{T \times d'}$), key ($K \in \mathbb{R}^{T \times d'}$), and value ($V \in \mathbb{R}^{T \times d'}$), by linear transformation, where d' is not necessarily same as d . Then, the self-attention is defined as $\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d'}}\right)V$. Intuitively, the attention of each token is represented as a weighted average of other token embeddings (using V) in the same sequence, where the weight is proportional to the relevance (computed using Q and K) between them. The learnable parameters are linear mappers from token embeddings to Q , K , and V . Multiple heads are often used to allow each token to represent more than one semantics depending on the context.

After the multi-head self-attention, the embeddings are fed into a position-wise feed-forward network, allowing further transformation. These steps are repeated by L blocks. The output of the last encoder block is the final embedding of each token. Optionally, we may put an additional classification token ([CLS]) to learn the aggregated representation of the entire sequence. Without having specific meaning, [CLS] aggregates tokens without being biased towards itself as other regular tokens do. The Transformer is often trained by losses arisen from a downstream task like classification, performed based on this aggregated embedding from [CLS] token.

4.2 Transformers for the Text Modality.

Bidirectional Encoder Representations from Transformers (BERT) [15] is a language model that learns representations from unlabeled text by self-supervised learning, based on the Transformer encoder. The

main training objectives are to predict masked tokens in sentences (Masked language modeling; MLM) and to predict whether two input sentences are consecutive (Next Sentence Prediction; NSP). With MLM, the randomly masked tokens are classified based on context (remaining tokens). For NSP, the embedding corresponding to the [CLS] token is fed to a classifier determining if the two input sentences are consecutive. For both, a classification loss (e.g., cross entropy) is used to train the model. BERT is powerful in precisely learning semantics of words when trained on large-scale corpus, achieving state-of-the-art performance on various NLP tasks.

4.3 Transformers for the Visual Modality.

The Vision Transformer (ViT) [17] is a Transformer-based object recognition or image classification model. ViT employs a Transformer over fixed-size (e.g., 16×16) patches split from the input image. Each image patch is linearly transformed to a patch embedding, added with learnable positional encoding and fed into the Transformer encoder. Optionally, multiple blocks of Transformers may be stacked. At the end of the last block, a learnable classification [CLS] token is appended to aggregate the learned representation of the entire image. It is fed into an MLP head performing the downstream task, e.g., image classification. Video Vision Transformer (ViViT) [4] and TimeSFormer [7] extend this idea to the sequence of frames, equipped with several options to reduce computational overhead.

5 THE PROPOSED METHOD

For a user i and an item j , the goal of our model estimates the preference score R_{ij} . As illustrated in Fig. 1, the user representation $u_i \in \mathbb{R}^D$ is simply learned with an embedding layer, similarly to

the traditional collaborative filtering models. In order to treat cold-start items, however, item representations are learned from their content information. Given C content information $\{\mathbf{x}_j^{(1)}, \dots, \mathbf{x}_j^{(C)}\}$ for each item j , where $\mathbf{x}_j^{(c)} \in \mathbb{R}^{D_c}$ for $c = 1, \dots, C$ and each $\mathbf{x}_j^{(c)}$ is potentially in different forms from various modalities, our model feeds each of them into a modality-specific encoder to embed them into a common embedding space. This embedding is annotated by $\mathbf{z}_j^{(c)} \in \mathbb{R}^d$ for $c = 1, \dots, C$. If $C > 1$, all C content embeddings are fused into a single item representation $\mathbf{v}_j \in \mathbb{R}^D$ by the Feature Fusion Layer (See Sec. 5.2). The final preference \mathbf{R}_{ij} is estimated by the dot-product of user and item embeddings; that is, $\hat{\mathbf{R}}_{ij} = \mathbf{u}_i^\top \mathbf{v}_j$.

The overall architecture might look standard in recommendation literature; however, we adopt Transformer-based architectures universally for all Modality-specific Encoders, enabling flexible contextualization and fusion across different features. More details on how to represent each modality will be described in Sec. 5.1.

5.1 Modality-specific Encoders

We elaborate our Transformer-based modality-specific encoders, illustrating visual and text representation modules representatively. We emphasize, however, any feature type can be applied similarly.

5.1.1 Image Encoder. The left-most box in Fig. 2 illustrates our Image Encoder. To be uncluttered, we omit the feature index superscript (c) and the item index subscript j inside each modality-specific encoder. Given an image $\mathbf{x} \in \mathbb{R}^{H \times W \times 3}$, where H and W are its height and width, respectively, it is divided into $P \times P$ smaller image patches, forming a set $\{\mathbf{x}_{[1,1]}, \mathbf{x}_{[1,2]}, \dots, \mathbf{x}_{[h,w]}\}$, where $h = H/P$, $w = W/P$, and $\mathbf{x}_{[a,b]} \in \mathbb{R}^{P \times P}$ for $a = 1, \dots, h$ and $b = 1, \dots, w$. Adopting ViT [17], our Image Encoder first linearly maps the input patches $\{\mathbf{x}_{[1,1]}, \mathbf{x}_{[1,2]}, \dots, \mathbf{x}_{[h,w]}\}$ to an embedding space with the Patch Embedding Layer, where the resulting embeddings are denoted by $\{\mathbf{z}_{[1,1]}, \mathbf{z}_{[1,2]}, \dots, \mathbf{z}_{[h,w]}\}$, where $\mathbf{z}_{[a,b]} \in \mathbb{R}^d$ for $a = 1, \dots, h$ and $b = 1, \dots, w$. Then, following common practice, learnable positional encodings $\{\mathbf{p}_{[a,b]} \in \mathbb{R}^d\}$ are added to the patch embeddings, depending on the location of each patch within the image. They are fed into L_c Transformer Encoder blocks, where $c = 1, \dots, C$ is the content attribute index, contextualizing them by repeated multi-head self-attention and multi-layer perceptrons. During this process, each patch embedding is updated to capture diverse semantics (e.g., objects and their relations) in the image. The output is the transformed sequence embeddings $\{\mathbf{z}_{[1,1]}, \mathbf{z}_{[1,2]}, \dots, \mathbf{z}_{[h,w]}\}$ from the last Transformer block. Optionally, an additional [CLS] token is appended to the sequence. The output embedding corresponding to this [CLS], denoted by $\mathbf{z} \in \mathbb{R}^d$, encodes semantics of the entire image \mathbf{x} . Either the entire sequence or this aggregated \mathbf{z} is used depending on the feature fusion methods (Sec. 5.2).

5.1.2 Video Encoder. The second encoder in Fig. 2 illustrates the Video Encoder. Instead of a single image, it takes as input a video $\mathbf{x} \in \mathbb{R}^{T \times H \times W \times 3}$ with T frames. Among T video frames, we first randomly sample a clip of F consecutive frames, denoted by $\{\mathbf{x}_{[1]}, \mathbf{x}_{[2]}, \dots, \mathbf{x}_{[F]}\}$. Then, we adopt a two-stage architecture where we first compute the frame-level embeddings $\{\mathbf{z}_{[1]}, \mathbf{z}_{[2]}, \dots, \mathbf{z}_{[F]}\}$, where $\mathbf{z}_{[f]} \in \mathbb{R}^d$ for each frame $f = 1, \dots, F$, using the Image Encoder with [CLS] (Sec. 5.1.1). Then, a learnable temporal positional encoding

$\{\mathbf{p}_{[1]}, \dots, \mathbf{p}_{[F]} \in \mathbb{R}^d\}$ is added. The sequence is fed into an additional Transformer Encoder. While the Image Encoder captures the spatial semantics of each frame, this second-level Transformer Encoder is in charge of capturing temporal semantics in the clip. Similarly to the image case, a classification token ([CLS]) may be appended to the sequence to aggregate the entire clip representation $\mathbf{z} \in \mathbb{R}^d$ out of it, or the entire output sequence $\{\mathbf{z}_{[1]}, \mathbf{z}_{[2]}, \dots, \mathbf{z}_{[F]}\}$ is kept depending on the fusion method.

We choose this two-stage architecture in order to effectively capture the spatio-temporal semantics of the video, including both details at frame level and overall information flow through the temporal axis. The architecture is similar to the model 2 of the Video Vision Transformer (ViViT) [4], reported as the most efficient and cost-effective. In order to learn complex underlying spatio-temporal dynamics from videos, choosing a computationally efficient architecture is critically important.

5.1.3 Text Encoder. Similarly to the visual modalities, we use a Transformer-based Text Encoder similar to BERT [15], illustrated in the right-most box in Fig. 2.

Given a sequence $\{\mathbf{x}_{[1]}, \dots, \mathbf{x}_{[T]}\}$ of T words (or sub-words, depending on the particular tokenizer used), where $\mathbf{x}_{[t]} \in \{1, \dots, |V|\}$ for $t = 1, \dots, T$ and V is the vocabulary set, the Word Embedding Layer encodes them into a sequence of word embeddings, $\{\mathbf{z}_{[1]}, \dots, \mathbf{z}_{[T]}\}$ with $\mathbf{z}_{[t]} \in \mathbb{R}^d$ for each $t = 1, \dots, T$. Then, they are added with the positional encoding $\{\mathbf{p}_{[1]}, \dots, \mathbf{p}_{[T]} \in \mathbb{R}^d\}$. Unlike the learnable positional encodings used for visual modalities, we follow the fixed positional encodings following BERT [15], as it is more suitable for text. The position-aware word embeddings pass through a Transformer Encoder which contextualizes the word embeddings throughout the entire text and produces another sequence of transformed word representations.

5.2 Feature Fusion

Once C content information is represented in a common embedding space, $\mathbf{z}_j^{(1)}, \dots, \mathbf{z}_j^{(C)} \in \mathbb{R}^d$ for each item j , we fuse them into a single item embedding $\mathbf{v}_j \in \mathbb{R}^D$ through the Feature Fusion Layer. This fusion can be implemented in a variety of ways, described below.

5.2.1 Late Fusion. Different content signals are fused at the last step, where each modality-specific encoder provides an aggregated single embedding $\mathbf{z}^{(c)} \in \mathbb{R}^d$ for $c = 1, \dots, C$, corresponding to each content signal. For this, an additional [CLS] token is appended to the input sequence to each modality-specific encoder, illustrated in Fig. 2. Without being biased to any specific token, the output embedding $\mathbf{z}^{(c)} \in \mathbb{R}^d$ corresponding to this [CLS] token is used to represent each content signal c . Using the concatenation approach among various fusion methods, we map $\{\mathbf{z}^{(1)}, \dots, \mathbf{z}^{(C)}\}$ to the final item representation, $\mathbf{v} \in \mathbb{R}^D$, with a few fully-connected layers in the Feature Fusion Layer; that is, $\mathbb{R}^{C \times d} \rightarrow \mathbb{R}^D$. Thanks to the flexibility of MLPs, D is not necessarily same as d .

Note that each feature $\mathbf{z}_j^{(c)}$ may not be aligned yet even though they are in the same embedding space. To further align multimodal information for the same item j , we may additionally apply Multimodal Alignment Loss, detailed in Sec. 5.3.2.

5.2.2 Early Fusion. In contrast to the Late Fusion where we use only a single output embedding from each modality-specific encoder, with Early Fusion, information across different content signals are fused before the token embeddings are aggregated. That is, we fuse all output tokens $\{z_{[t]}^{(c)} \in \mathbb{R}^d\}$ from the modality-specific encoders for $t = 1, \dots, T_c$ and $c = 1, \dots, C$. An additional Transformer Encoder is adopted for this, taking all tokens $\{z_{[t]}^{(c)} \in \mathbb{R}^d\}$ to capture dependencies between them by cross-modal attention [45]. At the end, all contextualized token embeddings may be averaged, or an additional [CLS] token is appended, to aggregate the semantics of the target item, denoted by $\mathbf{v}_j \in \mathbb{R}^D$. Note that $D = d$ here due to restriction of the Transformer architecture that token embedding size should be always the same. To have $D \neq d$, a fully-connected layer may be added in the end.

With this Early Fusion, additional multimodal alignment may not be necessary, since tokens from all different side information are aligned within the Transformer structure described above.

5.2.3 Mixture of Late and Early Fusions. Between the two extreme cases of Late (Sec. 5.2.1) and Early (Sec. 5.2.2) Fusions, there are various possibility of mixing those two. For instance, only a subset of features are fused early and fed into the fusion Transformer.

5.3 Training Objectives

The entire model is trained *end-to-end* using the two losses: Rating Ranking Loss and Multimodal Alignment Loss.

5.3.1 Rating Ranking Loss. We train the model to predict higher scores for preferred items and lower scores for the others. That is, the model is trained to maximize $\{\hat{\mathbf{R}}_{ij} : \mathbf{R}_{ij} = 1\}$ and minimize $\{\hat{\mathbf{R}}_{ij} : \mathbf{R}_{ij} = 0\}$. We use contrastive loss, which has been widely adopted for representation learning [10, 24, 35, 66].

Specifically, for each user, the item paired in the same example (*i.e.*, this user actually likes the item) is used as positive, while all other items belonging to different pairs in the minibatch are considered as negatives. With contrastive loss, the encoder is trained to maximize the dot product between the user and item embeddings in the same pair, while minimizing that of the different pairs in the mini-batch. The Rating Ranking Loss \mathcal{L}_R for a pair of a user i and an item j is defined as

$$\mathcal{L}_R = -\log \frac{\exp(\mathbf{u}_i^\top \mathbf{v}_j)}{\sum_{j' \in \mathcal{B}} \exp(\mathbf{u}_i^\top \mathbf{v}_{j'})} - \log \frac{\exp(\mathbf{u}_i^\top \mathbf{v}_j)}{\sum_{j' \in \mathcal{B}} \exp(\mathbf{u}_{i'}^\top \mathbf{v}_j)}, \quad (1)$$

where \mathcal{B} is the set of user-item pairs in the minibatch.

Here, one might argue that the items in the minibatch other than the paired one with the user might not be actually negative. Unlike classification models like SimCLR [10], the user might actually like additional items other than the currently paired one. We thus optionally filter out these false negatives from the denominator of Eq. (1) for more precise training. We report empirical performance with or without false negative filtering in Sec. 6.4.

5.3.2 Multimodal Alignment Loss. With the Late Fusion of multiple ($C > 1$) content features, an additional Multimodal Alignment Loss can be beneficial, as various content features may not be aligned yet

in the common embedding space. Specifically, we apply contrastive loss to all item embeddings within the minibatch, maximizing the similarity between content embeddings for the same item, while minimizing it between all other combinations. Multimodal Alignment Loss \mathcal{L}_M is defined by

$$\mathcal{L}_M = -\log \frac{1}{Z} \sum_{c=1}^C \sum_{c' > c} e^{z_j^{(c)\top} z_{j'}^{(c')}}, \quad (2)$$

with $Z = \sum_{j' \in \mathcal{B}} \sum_{c=1}^C \sum_{c'=1}^C e^{z_j^{(c)\top} z_{j'}^{(c)'}}$,

where \mathcal{B} is the set of items in the minibatch. When \mathcal{L}_M is used, we linearly combine it with \mathcal{L}_R ; that is,

$$\mathcal{L} = \mathcal{L}_R + \lambda \mathcal{L}_M, \quad (3)$$

where λ controls relative importance of the two losses.

5.4 Inference

For videos, recall that we randomly sample a segment with F frames at training. At inference, we sample $S > 1$ segments and predict preference scores with each of them. Then, we aggregate those scores by taking the max:

$$\hat{\mathbf{R}}_{ij} = \max_{s=1, \dots, S} \mathbf{u}_i^\top \mathbf{v}_{j_s}, \quad (4)$$

where \mathbf{v}_{j_s} is the video embedding based on the segment s for the item j . In this way, we cover wider range of the video and compute the score based on a segment that the user most likely prefers.

For the news domain, the content representation embedding is directly generated by feeding the entire image and text without any cropping, so we simply estimate by

$$\hat{\mathbf{R}}_{ij} = \mathbf{u}_i^\top \mathbf{v}_j. \quad (5)$$

6 EXPERIMENTS

We conduct extensive experiments to verify the effectiveness of our framework on multiple recommendation domains and datasets.

6.1 Experimental Settings

Datasets. We choose the movie and news domain for our experiments. The movie domain is chosen as it contains the richest content signals, *e.g.*, visual scenes, textual summary or script, metadata like genre, director, or main actors. We choose the news domain due to its cold-start nature; that is, recent news articles are mostly valuable to recommend. As listed in Table 1, we use two widely-used standard benchmarks on the movie domain, MovieLens 25M [23] and Yahoo Movies [53]. For the news domain, we use Chosun News 2022, containing all the articles and user activities between January and December 2022 on chosun.com, one of the most representative newspapers in Korea. Both MovieLens and Yahoo Movies provide explicit ratings from 1 (least preferred) to 5 (most preferred), so we convert them to implicit ones with 3.5 as the threshold, following [38, 81]. Chosun News dataset considers a click from a user on a news article as a positive feedback, and negative otherwise.

We exclude items with any missing content information from all datasets. Also, we filter out users with less than 20 ratings from MovieLens, following [39, 69]. We do not filter out ratings from

Table 1: Overview of Our Datasets

Dataset	Users	Items	Ratings	Density	Domain
MovieLens 25M	162,541	62,423	25,000,095	0.246%	Movie
Yahoo Movies	7,642	11,915	211,231	0.232%	Movie
Chosun News 2022	389,188	288,540	42,457,502	0.038%	News

Yahoo Movies and Chosun News. After filtering, we randomly split the items into training, cold validation, and cold test with the ratio of 85:7.5:7.5 for MovieLens and 70:15:15 for Yahoo Movies. For Chosun News, we use activities from the first 6 months for training, next 3 months for validation, and the rest for testing. The cold validation set is used to tune hyper-parameters, and the cold test set is used to evaluate the final performance. The two movie datasets contain only 891 overlapping movies, $\sim 2.77\%$ out of 32,156 movies in total (regarding the transfer learning experiment in Sec. 6.5).

Content Features. As all three datasets provide limited content signals, we collect additional visual and text data.

For visual content of the movie datasets, we use movie trailers provided by MovieLens [1] and MovieNet [28], since the full videos are publicly unavailable for most movies due to copyright. From each video, frames of size 224×224 are sampled at 2 fps. We drop the first and last 10% of the sampled frames, since they are often age rating screen or ending credits. The average length of the trailers is 137 seconds for MovieLens and 140 seconds for Yahoo Movies, so we get around 220 frames per video on average for both datasets. For visual content of the news dataset, we use up to 3 images collected from the web queried by the title of each article.

For text content of the movie datasets, we use movie synopsis collected from *imdb.com* for MovieLens. Yahoo Movies self-contains synopsis. These synopses are 2–3 sentences that summarize the movie overview. The sentences are first tokenized at word level with the maximum length of 512, using uncased BERT_{BASE} tokenizer [15] with $|V| = 30,522$. The average number of text tokens is 54.7 and 83.0 for MovieLens and Yahoo Movies, respectively. For Chosun News, we use the title and the body text of the article, following the same preprocessing above. The average number of tokens in this dataset is 257.3. We use the KoBERT_{Base-V1} tokenizer [30] with $|V| = 8,002$ for Korean language in Chosun News.

Evaluation Metrics. We measure recommendation performance by ranking all unseen items for each user in a held-out test set and comparing the top K items from the ranked list with the items that the user actually gave positive feedback to. Following CLCRec [67], we treat all users with varied number of ratings equally by averaging the score for each user. We use three widely-used metrics for ranking tasks: {Precision, Recall, NDCG}@ K with $K = \{1, 5, 10, 20\}$.

Competing Models. We compare with 4 recent cold-start item recommendation models using content information: CLCRec [67], DropoutNet [64], CVAR [79], and PMGT [42]. For fair comparison, we use the same set of multimedia features for all models, not the categorical side information used in baseline papers (e.g., [79]); that is, ViT [17] embeddings pretrained on ImageNet [14] and BERT [15] embeddings for visual and text features, respectively.

Model Hyperparameters. We experiment with $D = \{32, 64, 128, 256, 512, 1024, 2048\}$ for the user (\mathbf{u}_i) and item (\mathbf{v}_j) embedding size. The token embedding size d is set to 192, following [4]. For visual features, we spatially split each frame to $P \times P$ patches with $P = 16$. We stack $L = 4$ Transformer blocks for the Image and Video Encoders, and $L = 12$ for the Text Encoder. Positional encodings \mathbf{p} for visual encoders are learned from data [17]. Visual encoders are trained from scratch to avoid use of classification labels, while the text encoder starts from the pretrained BERT [15]. For the Feature Fusion Layer, we try Late Fusion with a single or two-layer MLP and Early Fusion with a single Transformer layer (Sec. 6.4). We perform grid search for λ within the range $[0, 1]$. We randomly sample $S = 10$ segments for video inference.

Training Hyperparameters. We use Adam optimizer [33] with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$. We linearly warm up the learning rate during the first 3 epochs, and train up to 200 epochs. After 70% of training, we decay the learning rate to 20% of the initial one, which is found by grid search among $\{10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\}$. We use batch size $B = 48$. For the movie datasets, a single sub-clip of length $F = 32$ is randomly sampled within each trailer, allowing the model to see various parts of the video uniformly throughout the whole training process. All reported results are averaged over five experiments with random initialization.

6.2 Comparison to the Baselines

Table 2 reports the performance on cold-start recommendation evaluated by {NDCG, Prec, Recall}@ K with $K = \{1, 5, 10, 20\}$. Our model outperforms all baselines under almost all metrics. Another notable observation is the relationship between the models’ performance and the value of K . On MovieLens, the average number of positive items in the test set is 23.1. Our approach tends to be stronger with smaller K , so it will be more suitable for cases like watch next, where only the top one item will be auto-played. Baseline models like CVAR, on the other hand, tend to be stronger with larger K , so they will be more suitable for homepage recommendations, where multiple items are presented at the same time. On Yahoo Movies, the average number of positive items is 3.5, much lower than 20. Thus, all methods tend to show higher scores with larger K .

6.3 General (Warm) Recommendations

Although we focus on cold-start recommendation problem in this paper, we also evaluate on the general recommendation task, where some test items are not necessarily cold-start. We conduct this experiment on Yahoo Movies, where the training set is the same but the test set consists of both warm and cold items. We compare our model to DropoutNet and CVAR in Table 3. As seen in the table, our method consistently outperforms the two strongest baselines not just on the cold-start, but on the general recommendation task.

Table 2: Comparison with the Baselines on All Datasets (%)

Dataset	Method	NDCG (\uparrow)				Precision (\uparrow)				Recall (\uparrow)			
		@1	@5	@10	@20	@1	@5	@10	@20	@1	@5	@10	@20
MovieLens	DropoutNet [64]	7.33	4.99	4.72	5.29	7.33	4.36	4.27	4.79	7.33	4.38	4.34	5.54
	CLCRec [67]	9.09	6.59	6.77	8.62	9.09	6.02	6.31	7.19	9.09	6.06	6.71	10.27
	CVAR [79]	8.89	9.11	9.09	9.49	8.89	9.12	9.10	9.56	8.89	9.13	9.12	9.71
	PMGT [42]	14.13	7.64	7.22	8.54	14.13	6.20	6.24	7.54	14.13	6.22	6.43	8.98
	Ours	14.05	11.41	10.16	10.40	14.05	10.77	9.13	8.14	14.05	10.80	9.39	11.33
Yahoo Movies	DropoutNet [64]	1.10	1.68	2.40	3.29	1.10	1.12	1.10	1.01	1.10	2.16	4.05	6.40
	CLCRec [67]	0.75	6.50	6.47	6.65	0.75	3.87	2.24	1.24	0.75	7.95	8.34	8.97
	CVAR [79]	1.07	1.74	2.31	3.09	1.07	1.34	1.17	1.05	1.07	2.67	4.13	6.49
	PMGT [42]	2.04	4.55	5.27	6.47	2.04	3.38	2.33	1.88	2.04	6.10	8.48	12.31
	Ours	5.39	8.53	12.42	12.48	5.39	5.87	5.74	6.27	5.39	8.86	15.14	16.24
Chosun News	DropoutNet [64]	2.43	1.92	2.18	2.17	2.43	2.07	2.07	2.14	2.43	2.08	2.11	2.22
	CLCRec [67]	1.52	1.36	1.29	1.23	1.52	1.40	1.28	1.19	1.52	1.40	1.30	1.23
	CVAR [79]	3.34	3.11	2.75	2.63	3.34	2.92	2.61	2.34	3.34	2.93	2.66	2.41
	PMGT [42]	2.80	2.50	2.19	2.20	2.80	2.30	1.93	2.00	2.80	2.30	1.93	2.10
	Ours	4.13	4.06	3.52	3.68	4.13	4.13	3.49	3.47	4.13	4.14	3.50	3.53

Table 3: Evaluation on General (Warm + Cold) Recommendation Task (YM)

Modality	NDCG (\uparrow)				Precision (\uparrow)				Recall (\uparrow)			
	@1	@5	@10	@20	@1	@5	@10	@20	@1	@5	@10	@20
DropoutNet [64]	2.04	1.91	2.41	2.59	2.04	1.53	1.52	1.09	2.04	1.99	3.25	3.79
CVAR [79]	2.31	1.98	2.35	2.62	2.31	1.66	1.47	1.15	2.31	2.14	3.18	3.89
Ours	6.71	5.94	6.33	6.94	6.71	3.75	2.63	1.70	6.71	6.56	7.73	9.59

Table 4: Modality Ablation Study (CN)

Modality	NDCG (\uparrow)				Precision (\uparrow)				Recall (\uparrow)			
	@1	@5	@10	@20	@1	@5	@10	@20	@1	@5	@10	@20
Visual Only	0.61	1.06	1.02	1.01	0.61	1.22	1.07	0.93	0.61	1.22	1.07	1.27
Text Only	3.51	2.69	2.87	2.83	3.51	2.76	2.97	2.86	3.51	2.76	2.97	2.88
Visual + Text	4.13	4.06	3.52	3.68	4.13	4.13	3.49	3.47	4.13	4.14	3.50	3.53

Table 5: MLP Architecture (CN)

Modality	NDCG (\uparrow)				Precision (\uparrow)				Recall (\uparrow)			
	@1	@5	@10	@20	@1	@5	@10	@20	@1	@5	@10	@20
No FC layer	3.72	3.41	3.40	3.38	3.72	3.53	3.44	3.26	3.72	3.53	3.44	3.56
1 FC layer	4.32	3.72	3.52	3.66	4.32	3.84	3.50	3.59	4.32	3.84	3.52	3.62
2 FC layers	4.02	3.89	3.77	3.68	4.02	3.84	3.69	3.58	4.02	3.84	3.69	3.63

6.4 Ablation Study

Modality Ablation. To explore the effectiveness of multimodal features and their alignment, we compare the performance of our full model against the same model with either visual or text content only. As seen in Table 4, using multimodal features and alignment loss improve the performance over single-modality baselines.

Model Architecture Ablation. We compare the Late and Early Fusions (Sec. 5.2). Table 6 reports that the Late Fusion outperforms on YahooMovies, while the Early Fusion slightly performs better on Chosun News. We conjecture that this difference comes from the nature of each dataset. Specifically, when two modalities provide more distinct information about the item, referring to tokens from each other is more beneficial during self-attention. The news domain

Table 6: Feature Fusion (YM, CN)

Data	Fusion	NDCG@10	Prec@10	Recall@10
YM	Late	12.42	5.74	15.14
	Early	10.00	4.19	14.18
CN	Late	3.40	3.43	3.44
	Early	3.52	3.49	3.50

Table 7: Embedding Size (YM)

D	NDCG@10	Prec@10	Recall@10
32	8.87	4.10	9.72
64	10.21	4.55	11.48
128	11.37	5.59	14.17
256	11.84	5.61	14.16
512	12.02	5.79	14.46
1024	12.42	5.74	15.14
2048	12.17	5.66	14.56

Table 8: False Negative Filter (ML)

Filtering	NDCG@10	Prec@10	Recall@10
Yes	9.85	9.14	9.21
No	10.16	9.13	9.39

*Datasets: ML = MovieLens, YM = Yahoo Movies, CN = Chosun News

seems more like this, where text tends to provide a chronological outline and interpretation of an incident in abstract, while images provide a snapshot of the event with visual details. Comparing against baselines in Table 2, however, our method still outperforms all baselines regardless of the fusion method.

We additionally compare the performance with various number of MLP layers after the fusion. From the model design perspective, having at least one FC layer is beneficial, since it allows us to arbitrarily set the output embedding dimensionality. Before fusion, we have C modalities, and simply concatenating the features from each modality results in a $D' = \sum_{c=1}^C d_c$ dimensional vector, where d_c is the feature dimensionality of modality c . Without any FC layer on top of this, this D' becomes the output vector size. With an FC layer, we can map D' to an arbitrary size D . Having additional FC layers, at least up to 2, is indeed beneficial, as shown in Table 5. Stacking more layers shows marginal performance gain, indicating that the complexity of content representations is learned well enough at the lower-level encoders, so the MLP layers can be concise.

Embedding Size Exploration. Table 7 summarizes the performance of our model with different embedding sizes (D) on Yahoo Movies. As expected, larger embedding size leads to better performance in general. It peaks around $D = 1024$ and saturates with diminishing returns. We also observe that $D = 128$ is a good trade-off between the cost and performance, aligned with a previous observation [38]. We use $D = 128$ for other ablation studies for efficient exploration.

False Negatives Filtering. To quantify the effect of false negatives discussed in Sec. 5.3.1, we compare our models with and without false negatives filtering in \mathcal{L}_R on MovieLens. Table 8 shows that this filtering has minimal impact. We conjecture that false negatives are less likely to be included in a small minibatch as the scale of the dataset gets larger. Considering additional computational overhead, we conduct all other experiments without it.

Table 9: Experimental Result on Content Representations

Pretraining	Target	N@10	P@10	R@10
From scratch	MovieLens	9.50	8.80	9.09
ViT (ImageNet)		5.32	5.31	5.50
From scratch	Yahoo Movies	7.25	3.63	9.24
ViT (ImageNet)		1.59	0.79	2.66
Ours (MovieLens)		5.22	2.63	6.90

*Metrics: N = NDCG, P = Precision, R = Recall

6.5 Content Representation Evaluation

To verify if our content embeddings properly capture users' watch behavior in general, we conduct two studies of transfer learning.

First, we compare our full model against the same model where the feature extractor is replaced with ViT [17] trained on ImageNet, average-pooled over the temporal axis. Comparing the first four rows in Table 9 reveals the difference of training directly on the user activities vs. classification data. We observe that the performance of our model trained from scratch outperforms the same model using ViT pre-trained embeddings on both MovieLens and Yahoo Movies. Our hypothesis that classification labels are not the best signal to train on for recommendation purpose is quantitatively confirmed from this result. From this, we confirm the importance of direct training on recommendation signals, instead of relying on labels for a proxy classification task.

Next, we evaluate transferability of our learned content representation from one dataset to another, to see if the learned content model is general enough to be competent on different set of users. The last row in Table 9 shows reasonable performance of cold-start recommendation on Yahoo Movies, using content embeddings trained on MovieLens. Considering the low overlapping movies ($\sim 2.77\%$) between these two datasets, our model turns out to truly map the raw content signals to users' taste, successfully transferring user behaviors from one dataset to another.

6.6 Qualitative Analysis

We visualize the learned video embeddings in 2D for qualitative understanding. Fig. 3 presents the t-SNE plot [62] of the video embeddings learned by our model using visual and text content on MovieLens. We observe that similar movies are positioned nearby each other in the embedding space. For instance, Fig. 3 illustrates 4 clusters with highly relevant movies in different colors: heroes (red), romantic comedies from mid-2000s (orange), science fictions (green), and western movies from mid-1900s (blue). The full list of colored dots is listed in Table 10.

For a deeper understanding of the improved performance of our model, we look into actual predicted scores for a couple of users in Table 11, comparing with the strongest baseline on MovieLens, CLCRec [67]. User 5649 is known to like the *Lion King* (1994) in the training set. Although this user likes animations, the test set indicates she prefers *Aladdin* (1992) and *Tarzan* (1999) (all from Disney), but not *Sinbad* (2003) from DreamWorks. Our model captures the user’s taste precisely, estimating higher scores, 0.848 and 0.762, for the two preferred items, while significantly lower one (-0.799) for *Sinbad*. The baseline, on the other hand, predicts similar scores for all three animations, even slightly higher (0.270) for *Sinbad*. This example illustrates that the proposed method trained directly on the user activities is better capable of capturing fine-grained tastes of users than previous works trained on classification labels.

Another example is user 1837. This user likes *Star Wars Episode IV, V, and VI*, but for some reason not the *Episode I*. Given the user likes the *Episode IV* only in the training set, our method retrieves *Episodes V and VI* (100% correct), while the baseline model recommends *Episode I and VI* (50% correct). Again, this example indicates our approach better captures subtle difference among multiple episodes of the same series, *Star Wars*, than existing methods.

In addition, Table 12 presents the pairwise cosine similarities between the clusters of select movies using the example users and items illustrated in Fig. 4, comparing with the same baseline. We calculate the cosine similarities between clusters based on the mean of the embeddings belonging to each cluster.

The first example is user 39430, who watched many fantasy movies like the *Lord of the Rings*, the *Hobbit*, and the *Harry Potter* series. This user liked the two *Lord of the Rings* movies in 2002 and 2003, but disliked an older one released in 1978. According to the third row of Table 12, our model successfully captures this difference (low similarity of -0.072), while the baseline model fails to (high similarity of 0.921). Fig. 4 indicates that the non-preferred movies are located far from the preferred ones in the embedding space, with the overall cosine similarity -0.0794. The baseline model, on the other hand, positions most of these fantasy movies closely to each other, with the cosine similarity 0.980. Interestingly, our model even distinguishes the disliked movies into two different clusters, clearly characterized by the release years (Fig. 4, left-most).

Another example is user 1837, also used in the Table 11, who likes *Star Wars Episodes IV, V, and VI*, but not the *Episode I*. Our model embeds the non-preferred series far from the preferred ones with the cosine similarity of -0.646, while CLCRec puts them closer with that of 0.811. (One might ask why the embedding space in Fig. 4 does not reflect this difference. This is because of the dimension reduction to 2D for visualization. The reported cosine similarities

are computed with the original embeddings before dimension reduction.) Comparing the preferred clusters, one from the training and the other from the validation set, we observe that our model locates them closer (0.627) than CLCRec (-0.532).

These examples illustrate that our approach captures the fine-grained difference among the movies of the same genre or even the same series, which is underrepresented with the baseline. We believe this difference comes from the fact that the existing models train the content feature on classification labels, forced to unlearn subtle differences between items belonging to the same class, while our model is completely free from the classification labels, fully utilizing its capacity to deeply understand the item contents.

7 SUMMARY

In this work, we propose a general item content representation learning framework to tackle the item cold-start recommendation problem. Our proposed framework is agnostic to a specific domain or dataset, applicable to various real-world services with minimal modifications. Taking advantage of the Transformer architecture, the proposed framework fuses signals from multimodal features in a natural way. Our framework does not rely on any human-labeled large-scale classification datasets to train modality-specific encoders. Relying solely on user activities, our model learns to represent items preserving fine-grained details of user tastes. From extensive experiments, we demonstrate the superior performance of our proposed framework both quantitatively and qualitatively, on movie and news domains with multiple datasets.

ETHICAL CONSIDERATIONS

The proposed approach in this paper is about how to better use raw content signals for cold-start item recommendations. We believe the proposed method itself does not impose any immediate positive or negative impact on fairness, privacy, or other ethical concerns, as long as the recommendation systems are trained on fairly collected training data.

We do expect, however, that this line of research possesses a potential to eventually promote user privacy. As the recommendation systems rely more heavily on content signals that are publicly available rather than individual user activity logs, less private data may need to be collected to achieve the same quality of recommendations. Although this work does not claim this line of contributions, it will be an interesting future work to explore and measure how content-based recommendation systems can save the privacy burden from modern recommendation systems.

REFERENCES

- [1] Sami Abu-El-Hajja, Joonseok Lee, Max Harper, and Joseph Konstan. 2018. MovieLens 20M YouTube Trailers Dataset.
- [2] Hassan Akbari, Liangzhe Yuan, Rui Qian, Wei-Hong Chuang, Shih-Fu Chang, Yin Cui, and Boqing Gong. 2021. VATT: Transformers for multimodal self-supervised learning from raw video, audio and text.
- [3] Mingxiao An, Fangzhao Wu, Chuhan Wu, Kun Zhang, Zheng Liu, and Xing Xie. 2019. Neural news recommendation with long-and short-term user representations. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*.
- [4] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. 2021. ViViT: A video vision transformer. In *Proc. of the IEEE/CVF Conference on International Conference on Computer Vision (ICCV)*.
- [5] Oren Barkan, Roy Hirsch, Ori Katz, Avi Caciularu, Jonathan Weill, and Noam Koenigstein. 2021. Cold item integration in deep hybrid recommenders via

Table 10: Full list of the movie titles in the colored clusters in Fig. 3

Red Cluster	Orange Cluster
Batman: The Dark Knight Returns, Part 1 (2012)	Love Actually (2003)
Batman: Year One (2011)	Break-Up, The (2006)
Superman Unbound (2013)	Notebook, The (2004)
Captain America: The First Avenger (2011)	How to Lose a Guy in 10 Days (2003)
Captain America: The Winter Soldier (2014)	12 Dates of Christmas (2011)
Iron Man 2 (2010)	Princess Diaries 2: Royal Engagement, The (2004)
Iron Man 3 (2013)	P.S. I Love You (2007)
Thor: The Dark World (2013)	Elizabethtown (2005)
Guardians of the Galaxy (2014)	Bridget Jones: The Edge of Reason (2004)
Fantastic Four (2005)	Catch and Release (2006)
Green Cluster	Blue Cluster
I, Robot (2004)	Duel in the Sun (1946)
Star Trek VI: The Undiscovered Country (1991)	Ride Lonesome (1959)
Star Wars: Episode I - The Phantom Menace (1999)	Montana (1950)
Star Wars: Episode II - Attack of the Clones (2002)	Man of the West (1958)
Back to the Future Part II (1989)	Man Who Never Was, The (1956)
Back to the Future Part III (1990)	Unforgiven, The (1960)
Matrix, The (1999)	Bonnie and Clyde (1967)
Matrix Revolutions, The (2003)	She Wore a Yellow Ribbon (1949)
Battlefield Earth (2000)	Rio Bravo (1959)
Pitch Black (2000)	Hombre (1967)

Table 11: Examples of Fine-grained Taste Estimation. Values in the range [-1, 1] represent preference.

User	Split	Movie Title	GT	Ours	Baseline
5649	Train	Lion King (1994)	Like	-	-
		Grease (1978)	Like	-	-
	Test	Aladdin (1992)	Like	0.848	0.006
		Tarzan (1999)	Like	0.762	0.102
		Sixth Sense (1999)	Like	0.424	-0.056
		Sinbad (2003)	Dislike	-0.799	0.270
1837	Train	Star Wars IV - A New Hope (1977)	Like	-	-
		Pulp Fiction (1994)	Like	-	-
		Forrest Gump (1994)	Like	-	-
	Test	Star Wars VI - Return of the Jedi (1983)	Like	0.489	0.242
		Star Wars V - Empire Strikes Back (1980)	Like	0.014	-0.294
		Star Wars I - Phantom Menace (1999)	Dislike	-0.276	0.462

Table 12: Cosine Similarity Examples of Item Embeddings

User	Comparison	Ours	Baseline
39430	Like (All ●) vs. Dislike (All ●)	-0.079	0.980
	Like (Lord of the Rings ●) vs. Dislike (Harry Potter & Hobbit ●)	-0.021	0.987
	Like (Lord of the Rings ●) vs. Dislike (Lord of the Ring ●)	-0.072	0.921
1837	Like (All ●) vs. Dislike (All ●)	-0.646	0.811
	Like (Training set ●) vs. Like (Validation set ○)	0.627	-0.532
	Like (Training set ●) vs. Dislike (All ●)	-0.717	-0.569



Figure 3: t-SNE Visualization of Learned Video Embeddings

- tunable stochastic gates. In *2021 IEEE International Conference on Data Mining (ICDM)*, IEEE, 994–999.
- [6] Oren Barkan, Noam Koenigstein, Eylon Yogev, and Ori Katz. 2019. CB2CF: a neural multiview content-to-collaborative filtering model for completely cold item recommendations. In *Proceedings of the 13th ACM Conference on Recommender Systems*. 228–236.
 - [7] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. 2021. Is space-time attention all you need for video understanding?. In *Proc. of the International Conference on Machine Learning (ICML)*.
 - [8] John S Breese, David Heckerman, and Carl Kadie. 1998. Empirical analysis of predictive algorithms for collaborative filtering. In *Proc. of the Conference on Uncertainty in Artificial Intelligence (UAI)*.
 - [9] Joao Carreira and Andrew Zisserman. 2017. Quo vadis, action recognition? a new model and the kinetics dataset. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
 - [10] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *Proc. of the International Conference on Machine Learning (ICML)*.
 - [11] Zewen Chi, Li Dong, Furu Wei, Nan Yang, Saksham Singhal, Wenhui Wang, Xia Song, Xian-Ling Mao, Heyan Huang, and Ming Zhou. 2021. InfoXLM: An information-theoretic framework for cross-lingual language model pre-training. (2021).
 - [12] Yashar Deldjoo, Maurizio Ferrari Dacrema, Mihai Gabriel Constantin, Hamid Eghbal-Zadeh, Stefano Cereda, Markus Schedl, Bogdan Ionescu, and Paolo Cremonesi. 2019. Movie genome: alleviating new item cold start in movie recommendation. *User Modeling and User-Adapted Interaction* 29, 2 (2019), 291–343.
 - [13] Yashar Deldjoo, Markus Schedl, Paolo Cremonesi, and Gabriella Pasi. 2020. Recommender systems leveraging multimedia content. *ACM Computing Surveys (CSUR)* 53, 5 (2020), 1–38.
 - [14] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
 - [15] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proc. of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.
 - [16] Manqing Dong, Feng Yuan, Lina Yao, Xiwei Xu, and Liming Zhu. 2020. Mamo: Memory-augmented meta-optimization for cold-start recommendation. In *Proc. of the ACM SIGKDD International conference on knowledge discovery & data mining*.
 - [17] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *Proc. of the International Conference on Learning Representations (ICLR)*.
 - [18] Xiaoyu Du, Xiang Wang, Xiangnan He, Zechao Li, Jinhui Tang, and Tat-Seng Chua. 2020. How to learn item representation for cold-start multimedia recommendation?. In *Proc. of the ACM International Conference on Multimedia*.
 - [19] Xue Geng, Hanwang Zhang, Jingwen Bian, and Tat-Seng Chua. 2015. Learning image and user features for recommendation in social networks. In *Proc. of the IEEE/CVF Conference on International Conference on Computer Vision (ICCV)*.
 - [20] David Goldberg, David Nichols, Brian M Oki, and Douglas Terry. 1992. Using collaborative filtering to weave an information tapestry. *Commun. ACM* 35, 12 (1992), 61–70.
 - [21] Yuan Gong, Yu-An Chung, and James Glass. 2021. AST: Audio spectrogram transformer. *arXiv:2104.01778* (2021).
 - [22] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. 2020. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in Neural Information Processing Systems (NIPS)* 33 (2020).
 - [23] F Maxwell Harper and Joseph A Konstan. 2015. The MovieLens datasets: History and context. *ACM Transactions on interactive intelligent systems (TIIS)* 5, 4 (2015), 1–19.
 - [24] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *CVPR*.
 - [25] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proc. of the ACM International Conference on World Wide Web (WWW)*.
 - [26] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. 2019. Learning deep representations by mutual information estimation and maximization. In *Proc. of the International Conference on Learning Representations (ICLR)*.
 - [27] Qingqing Huang, Aren Jansen, Joonseok Lee, Ravi Ganti, Judith Yue Li, and Daniel PW Ellis. 2022. MuLan: A joint embedding of music audio and natural language. In *Proc. of the International Society for Music Information Retrieval Conference (ISMIR)*.
 - [28] Qingqu Huang, Yu Xiong, Anyi Rao, Jiaze Wang, and Dahua Lin. 2020. MovieNet: A holistic dataset for movie understanding. In *Proc. of the European Conference on Computer Vision (ECCV)*.
 - [29] Seong Jae Hwang, Joonseok Lee, Balakrishnan Varadarajan, Ariel Gordon, Zheng Xu, and Apostol Natsev. 2019. Large-scale training framework for video annotation. In *Proc. of the ACM SIGKDD International conference on knowledge discovery & data mining*.
 - [30] Heewon Jeon, Donggeon Lee, and Jangwon Park. 2019. Korean bert pre-trained cased (kobert). URL <https://github.com/SKTBrain/KoBERT> (2019).
 - [31] Junyan Jiang, Gus G Xia, Dave B Carlton, Chris N Anderson, and Ryan H Miyakawa. 2020. Transformer VAE: A hierarchical model for structure-aware and interpretable music representation learning. In *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

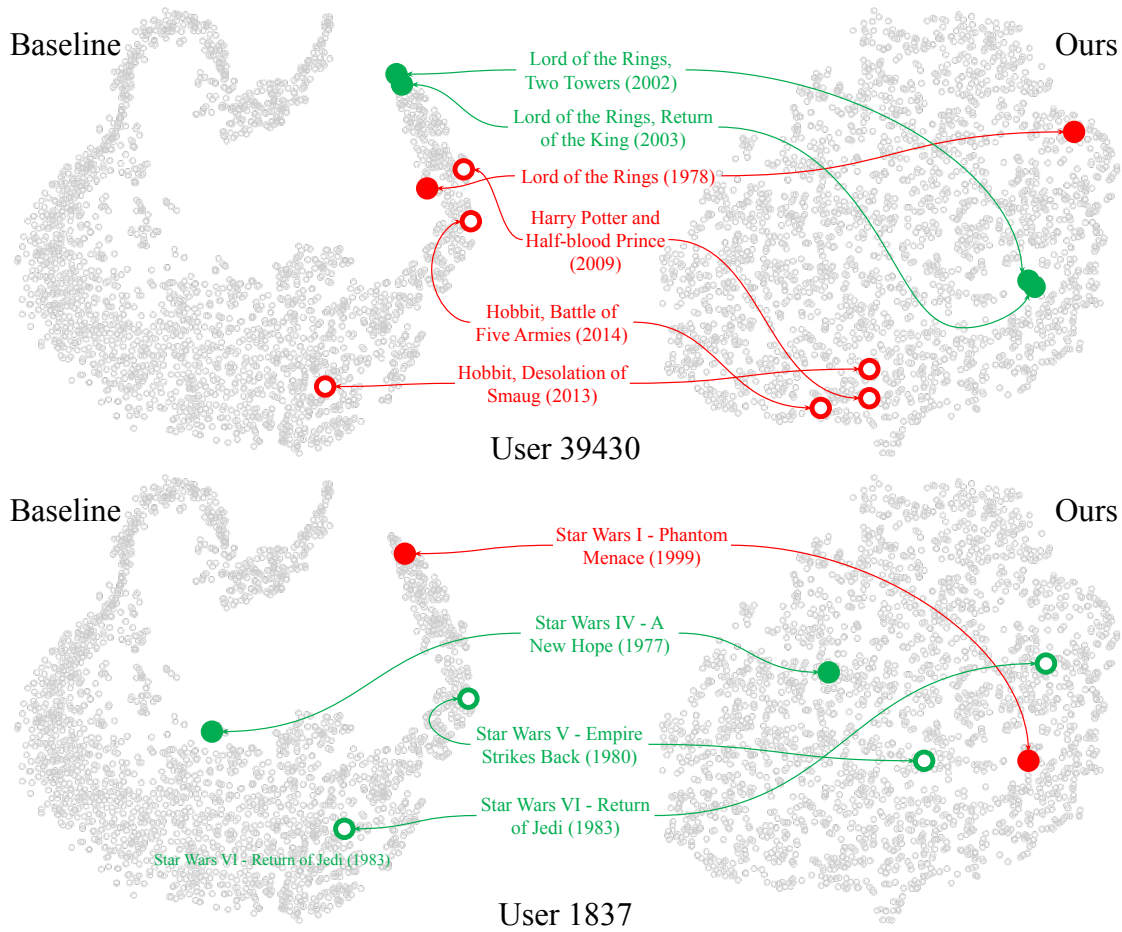


Figure 4: Illustration of item embeddings used in the pairwise similarity analysis in Table 12.

- [32] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschiot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. *Advances in Neural Information Processing Systems (NIPS)* 33 (2020).
- [33] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
- [34] Michal Kompan and Mária Bielíková. 2010. Content-based news recommendation. In *International conference on electronic commerce and web technologies*. Springer.
- [35] Phuc H Le-Khac, Graham Healy, and Alan F Smeaton. 2020. Contrastive representation learning: A framework and review. *IEEE Access* 8 (2020), 193907–193934.
- [36] Hoyeop Lee, Jinbae Im, Seongwon Jang, Hyunsouk Cho, and Sehee Chung. 2019. Melu: Meta-learned user preference estimator for cold-start recommendation. In *Proc. of the ACM SIGKDD International conference on knowledge discovery & data mining*.
- [37] Hyodong Lee, Joonseok Lee, Joe Yue-Hei Ng, and Paul Natsev. 2020. Large Scale Video Representation Learning via Relational Graph Clustering. In *Proc. of the ACM SIGKDD International conference on knowledge discovery & data mining*.
- [38] Joonseok Lee, Sami Abu-El-Haija, Balakrishnan Varadarajan, and Apostol Natsev. 2018. Collaborative deep metric learning for video understanding. In *Proc. of the ACM SIGKDD International conference on knowledge discovery & data mining*.
- [39] Joonseok Lee, Samy Bengio, Seungyeon Kim, Guy Lebanon, and Yoram Singer. 2014. Local collaborative ranking. In *Proc. of the ACM International Conference on World Wide Web (WWW)*.
- [40] Joonseok Lee, Seungyeon Kim, Guy Lebanon, and Yoram Singer. 2013. Local Low-Rank Matrix Approximation. In *Proc. of the International Conference on Machine Learning (ICML)*.
- [41] Tianqiao Liu, Zhiwei Wang, Jiliang Tang, Songfan Yang, Gale Yan Huang, and Zitao Liu. 2019. Recommender systems with heterogeneous side information. In *Proc. of the ACM International Conference on World Wide Web (WWW)*.
- [42] Yong Liu, Susen Yang, Chenyi Lei, Guoxin Wang, Haihong Tang, Juyong Zhang, Aixin Sun, and Chunyan Miao. 2021. Pre-training graph transformer with multi-modal side information for recommendation. In *Proc. of the ACM International Conference on Multimedia (MM)*.
- [43] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proc. of the IEEE/CVF Conference on International Conference on Computer Vision (ICCV)*.
- [44] Zhuang Liu, Yunpu Ma, Yuanxin Ouyang, and Zhang Xiong. 2021. Contrastive learning for recommender system. *arXiv:2101.01317* (2021).
- [45] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. ViLBERT: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in Neural Information Processing Systems (NIPS)* 32 (2019).
- [46] Yuanfu Lu, Yuan Fang, and Chuan Shi. 2020. Meta-learning on heterogeneous information networks for cold-start recommendation. In *Proc. of the ACM SIGKDD International conference on knowledge discovery & data mining*.
- [47] Kindian Ma, Peng Zhang, Shuai Zhang, Nan Duan, Yuexian Hou, Ming Zhou, and Dawei Song. 2019. A tensorized transformer for language modeling. In *Advances in Neural Information Processing Systems (NIPS)*.
- [48] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. 2019. HowTo100M: Learning a text-video embedding by watching hundred million narrated video clips. In *Proc. of the IEEE/CVF Conference on International Conference on Computer Vision (ICCV)*.
- [49] Xia Ning and George Karypis. 2012. Sparse linear methods with side information for top-n recommendations. In *Proc. of the ACM Conference on Recommender Systems (RecSys)*.
- [50] Feiyang Pan, Shuokai Li, Xiang Ao, Pingzhong Tang, and Qing He. 2019. Warm up cold-start advertisements: Improving ctr predictions via learning to learn id embeddings. In *Proc. of the International ACM Conference on Research and*

- Development in Information Retrieval (SIGIR)*.
- [51] Jiyoung Park, Jongpil Lee, Jangyeon Park, Jung-Woo Ha, and Juhan Nam. 2017. Representation learning of music using artist labels. *arXiv:1710.06648* (2017).
 - [52] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proc. of the Conference on Uncertainty in Artificial Intelligence (UAI)*.
 - [53] Nachiketa Sahoo, R Krishnan, G Duncan, and J Callan. 2008. On multi-component rating and collaborative filtering for recommender systems: The case of yahoo! movies. *Information Systems Research* (2008).
 - [54] J Ben Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen. 2007. Collaborative filtering recommender systems. In *The adaptive web*. Springer.
 - [55] Shaoyun Shi, Min Zhang, Xinxing Yu, Yongfeng Zhang, Bin Hao, Yiqun Liu, and Shaoping Ma. 2019. Adaptive feature sampling for recommendation with missing content feature values. In *Proc. of the ACM International Conference on Information and Knowledge Management (CIKM)*.
 - [56] Xiaoyuan Su and Taghi M Khoshgoftaar. 2009. A survey of collaborative filtering techniques. *Advances in artificial intelligence* (2009).
 - [57] Chen Sun, Fabien Baradel, Kevin Murphy, and Cordelia Schmid. 2019. Learning video representations using contrastive bidirectional transformer. *arXiv:1906.05743* (2019).
 - [58] Chen Sun, Abhinav Srivastava, Saurabh Singh, and Abhinav Gupta. 2017. Revisiting unreasonable effectiveness of data in deep learning era. In *Proc. of the IEEE/CVF Conference on International Conference on Computer Vision (ICCV)*.
 - [59] Zhu Sun, Qing Guo, Jie Yang, Hui Fang, Guibing Guo, Jie Zhang, and Robin Burke. 2019. Research commentary on recommendations with side information: A survey and research directions. *Electronic Commerce Research and Applications* 37 (2019), 100879.
 - [60] Zhulin Tao, Xiaohao Liu, Yewei Xia, Xiang Wang, Lifang Yang, Xianglin Huang, and Tat-Seng Chua. 2022. Self-supervised learning for multimedia recommendation. *IEEE Transactions on Multimedia* (2022).
 - [61] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. 2018. A closer look at spatiotemporal convolutions for action recognition. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
 - [62] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research (JMLR)* 9, 11 (2008).
 - [63] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. In *Advances in Neural Information Processing Systems (NIPS)*.
 - [64] Maksims Volkovs, Guangwei Yu, and Tomi Poutanen. 2017. DropoutNet: Addressing cold start in recommender systems. *Advances in neural information processing systems* 30 (2017).
 - [65] Hongwei Wang, Fuzheng Zhang, Xing Xie, and Minyi Guo. 2018. DKN: Deep knowledge-aware network for news recommendation. In *Proc. of the ACM International Conference on World Wide Web (WWW)*.
 - [66] Tongzhou Wang and Phillip Isola. 2020. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *Proc. of the International Conference on Machine Learning (ICML)*.
 - [67] Yinwei Wei, Xiang Wang, Qi Li, Liqiang Nie, Yan Li, Xuanping Li, and Tat-Seng Chua. 2021. Contrastive learning for cold-start recommendation. In *Proc. of the ACM International Conference on Multimedia*.
 - [68] Yinwei Wei, Xiang Wang, Liqiang Nie, Xiangnan He, Richang Hong, and Tat-Seng Chua. 2019. MMGCN: Multi-modal graph convolution network for personalized recommendation of micro-video. In *Proc. of the ACM International Conference on Multimedia (MM)*.
 - [69] Markus Weimer, Alexandros Karatzoglou, Quoc Le, and Alex Smola. 2007. CoFi Rank-maximum margin matrix factorization for collaborative ranking. In *Advances in Neural Information Processing Systems (NIPS)*.
 - [70] Chuhan Wu, Fangzhao Wu, Mingxiao An, Jianqiang Huang, Yongfeng Huang, and Xing Xie. 2019. Neural news recommendation with attentive multi-view learning. (2019).
 - [71] Chuhan Wu, Fangzhao Wu, Mingxiao An, Jianqiang Huang, Yongfeng Huang, and Xing Xie. 2019. NPA: neural news recommendation with personalized attention. In *Proc. of the ACM SIGKDD International conference on knowledge discovery & data mining*.
 - [72] Chuhan Wu, Fangzhao Wu, Mingxiao An, Yongfeng Huang, and Xing Xie. 2019. Neural news recommendation with topic-aware news representation. In *Proc. of the Annual meeting of the association for computational linguistics (ACL)*.
 - [73] Chuhan Wu, Fangzhao Wu, Suyu Ge, Tao Qi, Yongfeng Huang, and Xing Xie. 2019. Neural news recommendation with multi-head self-attention. In *Proc. of the conference on empirical methods in natural language processing and the international joint conference on natural language processing (EMNLP-IJCNLP)*.
 - [74] Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. 2021. Empowering news recommendation with pre-trained language models. In *Proc. of the International ACM Conference on Research and Development in Information Retrieval (SIGIR)*.
 - [75] Chuhan Wu, Fangzhao Wu, Tao Qi, Chao Zhang, Yongfeng Huang, and Tong Xu. 2022. MM-Rec: Visiolinguistic Model Empowered Multimodal News Recommendation. In *Proc. of the International ACM Conference on Research and Development in Information Retrieval (SIGIR)*.
 - [76] Shiwen Wu, Fei Sun, Wentao Zhang, Xu Xie, and Bin Cui. 2022. Graph neural networks in recommender systems: a survey. *Comput. Surveys* 55, 5 (2022), 1–37.
 - [77] Runsheng Yu, Yu Gong, Xu He, Yu Zhu, Qingwen Liu, Wenwu Ou, and Bo An. 2021. Personalized adaptive meta learning for cold-start user preference prediction. In *Proc. of the AAAI Conference on Artificial Intelligence*.
 - [78] Ruisheng Zhang, Qi-dong Liu, Jia-Xuan Wei, et al. 2014. Collaborative filtering for recommender systems. In *Proc. of the IEEE International Conference on Advanced Cloud and Big Data*.
 - [79] Xu Zhao, Yi Ren, Ying Du, Shenzheng Zhang, and Nian Wang. 2022. Improving Item Cold-start Recommendation via Model-agnostic Conditional Variational Autoencoder. *arXiv:2205.13795* (2022).
 - [80] Chang Zhou, Jianxin Ma, Jianwei Zhang, Jingren Zhou, and Hongxia Yang. 2021. Contrastive learning for debiased candidate generation in large-scale recommender systems. In *Proc. of the ACM SIGKDD International conference on knowledge discovery & data mining*.
 - [81] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep interest network for click-through rate prediction. In *Proc. of the ACM SIGKDD International conference on knowledge discovery & data mining*.
 - [82] Yongchun Zhu, Ruobing Xie, Fuzhen Zhuang, Kaikai Ge, Ying Sun, Xu Zhang, Leyu Lin, and Juan Cao. 2021. Learning to warm up cold item embeddings for cold-start recommendation with meta scaling and shifting networks. In *Proc. of the International ACM Conference on Research and Development in Information Retrieval (SIGIR)*.
 - [83] Ziwei Zhu, Shahin Sefati, Parsa Saadatpanah, and James Caverlee. 2020. Recommendation for new users and new items via randomized training and mixture-of-experts transformation. In *Proc. of the International ACM Conference on Research and Development in Information Retrieval (SIGIR)*.