Online Distillation-enhanced Multi-modal Transformer for Sequential Recommendation

Wei Ji*
National University of Singapore
Singapore, Singapore
weiji0523@gmail.com

Xiangyan Liu*
National University of Singapore
Singapore, Singapore
liu.xiangyan@u.nus.edu

An Zhang†
National University of Singapore
Singapore, Singapore
anzhang@u.nus.edu

Yinwei Wei
Monash University
Melbourne, Australia
weiyinwei@hotmail.com

Yongxin Ni
National University of Singapore
Singapore, Singapore
niyongxin@u.nus.edu

Xiang Wang‡
University of Science and Technology of China
Hefei, China
xiangwang1223@gmail.com

ABSTRACT

Multi-modal recommendation systems, which integrate diverse types of information, have gained widespread attention in recent years. However, compared to traditional collaborative filtering-based multi-modal recommendation systems, research on multi-modal sequential recommendation is still in its nascent stages. Unlike traditional sequential recommendation models that solely rely on item identifier (ID) information and focus on network structure design, multi-modal recommendation models need to emphasize item representation learning and the fusion of heterogeneous data sources. This paper investigates the impact of item representation learning on downstream recommendation tasks and examines the disparities in information fusion at different stages. Empirical experiments are conducted to demonstrate the need to design a framework suitable for collaborative learning and fusion of diverse information. Based on this, we propose a new model-agnostic framework for multi-modal sequential recommendation tasks, called **Online Distillation-enhanced Multi-modal Transformer (ODMT)**, to enhance feature interaction and mutual learning among multi-source input (ID, text, and image), while avoiding conflicts among different features during training, thereby improving recommendation accuracy. To be specific, we first introduce an ID-aware Multi-modal Transformer module in the item representation learning stage to facilitate information interaction among different features. Secondly, we employ an online distillation training strategy in the prediction optimization stage to make multi-source data learn from each other and improve prediction robustness. Experimental results on a stream media recommendation dataset and three e-commerce recommendation datasets demonstrate the effectiveness of the proposed two modules, which is approximately 10% improvement in performance compared to baseline models. Our code will be released at: https://github.com/xyliugo/ODMT.

CCS CONCEPTS

• **Computing methodologies** → **Recommender systems**.

KEYWORDS

Multi-modal Recommendation, Knowledge Distillation, Sequential Recommendation

ACM Reference Format:
Wei Ji, Xiangyan Liu, An Zhang, Yinwei Wei, Yongxin Ni, and Xiang Wang. 2023. Online Distillation-enhanced Multi-modal Transformer for Sequential Recommendation. In *Proceedings of the 31st ACM International Conference on Multimedia (MM '23), October 29-November 3, 2023, Ottawa, ON, Canada.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3581783.3612091

1 INTRODUCTION

With the emergency of multimedia platforms (*e.g.*, TikTok, Youtube), multi-modal recommendation system is becoming increasingly important in both academic [32, 37, 51, 55] and industry [2, 41]. It aims to understand user preferences of multi-modal content information, based on their historical behaviors (*e.g.*, clicks, comments).

Compared to general recommendation systems, multi-modal recommendation requires not only model architecture design, but also consideration of how to effectively apply multi-modal features in downstream tasks, especially in a way compatible with the current recommendation system. Most current recommendation systems rely on collaborative filtering [10, 11, 31, 48, 49], which predicts user-item interactions by modeling users and items separately and then computing the similarity between the user and candidate item to generate a prediction score. In general multi-modal recommendation systems, multi-modal features are used to enhance the connection between user-item pairs [32, 34, 36] or as side information, which is complementary to ID features [9, 52]. In contrast, sequential recommendation systems rely more on item representation learning than collaborative filtering and model user representations according to items clicked by the user and their temporal sequences [20, 29, 38, 39, 57]. This approach places a higher demand on item representation learning, especially in the case of multi-modal recommendation systems, where the raw item information is more diverse.

Our study focuses on item representation learning in multi-modal sequential recommendation systems and explores the performance of single-modal features (text or image) as individual input and as input combined with ID features in downstream recommendation tasks. We also investigate different fusion strategies

*Equal Contribution
†Corresponding Author
‡Xiang Wang is also affiliated with Institute of Artificial Intelligence, Institute of Dataspace, Hefei Comprehensive National Science Center

when combining ID and multi-modal information. Our exploration experiments reveal that compared to other network structures, Transformers provide better semantic transformation and representation learning for single-modal features, leading to more accurate predictions in recommendation tasks. However, the advantage of strong representation brought by Transformers weakens when using multi-source data (ID, text, and image) as input. Further analysis reveals that this is due to ID features being easier to optimize and producing lower training loss in recommendation prediction tasks, whereas multi-modal features provide valuable prior information on item similarity, making recommendation systems easy to retrieve items of interest for users. Therefore, when ID and modal information are combined, the improvement in evaluation metrics may not perfectly align with the direction of the loss reduction.

To address the challenges in multi-modal sequential recommendation, we propose a new model-agnostic framework called Online Distillation-enhanced Multi-modal Transformer (ODMT) equipped with two novel modules. Firstly, we introduce an ID-aware Multi-modal Transformer module in the item representation learning stage to facilitate information interaction among different features. Secondly, we apply an online distillation training strategy in the prediction optimization stage to obtain more robust predictions without compromising the loss optimization of the multi-modal features. Overall, our contributions can be summarized as follows:

- To relieve the incompatibility issue between multi-modal features and existing sequential recommendation models, we introduce the ODMT framework, which comprises an ID-aware Multi-modal Transformer module for item representation.
- To obtain robust predictions from multi-source input, we propose an online distillation training strategy in the prediction optimization stage, which marks the first instance of applying online distillation to a multi-modal recommendation task.
- Comprehensive experiments on **four** diverse multi-modal recommendation datasets and **three** popular backbones for sequential recommendation validate the effectiveness and transferability of proposed method, which is about 10% performance improvement compared with other baseline models.

2 PRELIMINARIES

This section aims to thoroughly investigate the effects of Item Representation Learning (IRL) and Information Fusion (IF) modules on downstream recommendation networks.

We provide empirical evidence through experiments, which highlight two key findings: 1) Transformers are effective in transforming multi-modal information from general semantic to specific recommendation semantic; 2) simple fusion strategies can cause a discrepancy between the direction of loss optimization and the direction of metric improvement during the training process, thus affecting the significance of multi-modal features in the recommendation model.

2.1 Brief Concepts

The IRL module is responsible for generating the final item embeddings by converting raw input data into distributed representations. Input data can be categorized as item ID or item other modalities (*e.g.*, image and text). The embedding table of items is a

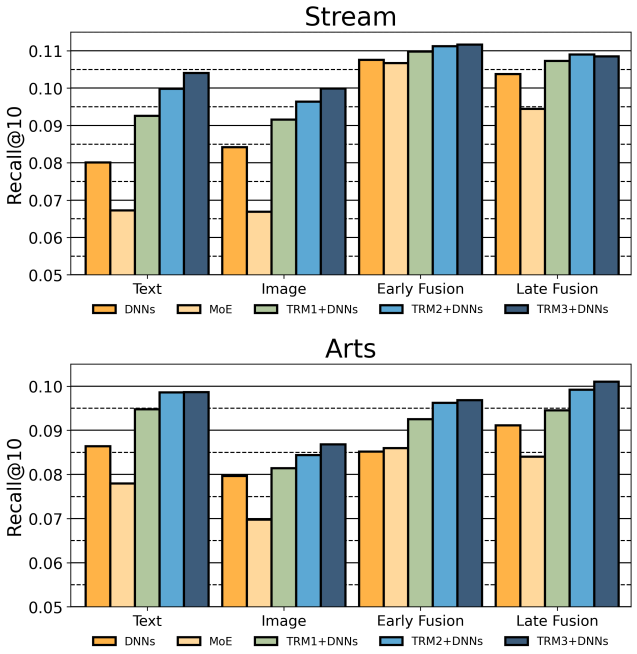


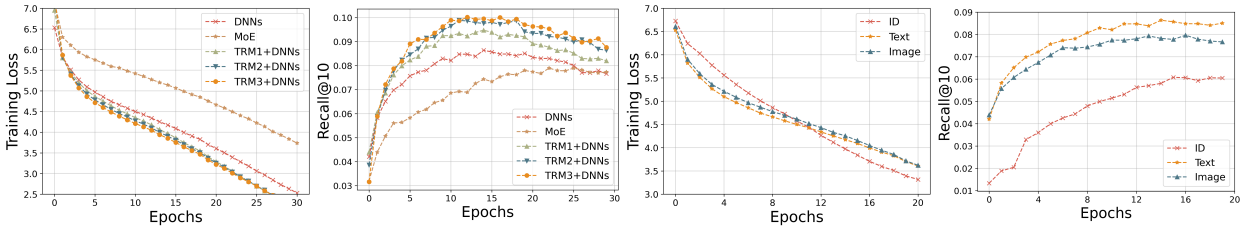
Figure 1: Comparison of Recall@10 for different FST modules based on single-modal and multi-source input, across Arts and Stream datasets. "x" in TRMx+DNNs represents the number of Transformer layers. For x = 1, it represents 1 Transformer layer for text and 1 Transformer layer for image.

crucial component in sequential recommendation models [12, 20]. Each item has a unique embedding representation that corresponds to its index. For multi-modal data, BERT [6] and ViT [7] are utilized to extract textual and visual features from raw data. The extracted multi-modal features are then fed into a Feature Semantic Transformation (FST) module to convert modal information into a semantic space suitable for recommendations. Our FST module candidates include DNNs, MoE Adaptor (MoE), and Transformers+DNNs (TRM+DNNs), which have been widely used in previous researches [15, 24, 30, 46].

In multi-modal sequential recommendation models, the IF module can be categorized into Early Fusion and Late Fusion [14, 26, 28, 52]. Early Fusion involves embedding all item information into a single feature representation, which is then inputted into the model, while Late Fusion is based on the prediction results or the prediction scores of each feature. In this paper, we consider three types of input information, namely ID, text, and image, that need to be fused. In Early Fusion, we obtain fused item embeddings by averaging the three features. In contrast, in Late Fusion, we get a user's general preferences by averaging the three logits, where different logits correspond to different user preferences. Intermediate Fusion is not discussed in this section, as it is a model-dependent fusion method [56].

2.2 Empirical Explorations

To ensure fair comparison in our experiments, we control all variables other than the FST module and IF module, including random



(a) Training Loss v.s. FST (b) Testing Recall v.s. FST (c) Training Loss v.s. Input (d) Testing Recall v.s. Input

Figure 2: Training curves of FST modules with text input, evaluating (a) training loss and (b) testing Recall@10, and training curves of different single input with DNNs as FST module, evaluating (c) training loss and (d) testing Recall@10 on the Arts dataset.

seed, pre-trained encoders, hyper-parameters (*e.g.*, learning rate, embedding size, hidden size, and dropout ratio), and experimental codes, all experiments are conducted in a unified framework. The backbone model for our sequential model is the representative one, SASRec [20], which uses self-attention mechanisms for sequence modeling. Figure 1 shows experimental results on the Stream and Arts (Amazon) datasets.

General visual and textual features extracted by pre-trained models (*e.g.*, BERT and ViT) are not necessarily suitable for recommendation tasks. Therefore, FST module is needed to transform the modality features into recommendation semantics. Figure 1 indicates that Transformers are capable of performing semantic transformation more effectively than DNNs and MoE when inputted with single-modal data. This finding highlights the potential of Transformers in learning powerful representations for recommendation systems.

Previous studies in multi-modal sequential recommendation models [15, 26, 28, 52] have typically used simple FST modules, such as DNNs and MoE, with Early Fusion or Late Fusion methods. In these cases, fusing multiple information sources has generally yielded favorable outcomes. When Transformers are employed as the FST module, we notice a diminishing advantage of utilizing multi-source input instead of single-modal input as the number of Transformer layers increases. Remarkably, even on the Arts dataset, the effectiveness of the single-modal input surpasses that of the multi-source fusion input. These results suggest that while Transformers possess strong representation learning capabilities, they may not be able to fully showcase their potential in the scenarios of multi-source input.

To provide more insights into the impact of FST modules and IF modules, we fix the input with plain text and the FST module with DNNs, respectively. Figure 2a and 2b illustrate the consistency between training loss and testing Recall, revealing that models with lower training loss corresponded to higher Recall scores. This observation suggests that single-modal input enables the FST module to better learn item representations and effectively reduce the training loss, leading to improved downstream recommendation performance. On the other hand, Figure 2c and 2d demonstrate the inconsistency between training loss and testing Recall when different types of information are used as input. This inconsistency may arise from overfitting or the mismatch between the objective function and the evaluation metric. For recommendation models, optimizing the ID features can be viewed as an unconstrained optimization problem, resulting in lower training loss. Conversely, optimizing modality features is subject to constraints due to the prior information that items' content has similarities. Therefore, even if the modality-based models do not achieve significantly lower training loss, they can achieve better performance, particularly when using Transformers as the FST module.

Based on our findings, we can conclude that the FST module plays a crucial role in extracting informative representations of items. Integrating multi-modal information may not lead to the optimal improvement in recommendation metrics, as there exists a misalignment between the direction of metric improvement and loss reduction. This misalignment creates a dilemma in learning both ID and modality features simultaneously for recommendation systems. The dilemma becomes more challenging as the modality representation capacity increases, which can ultimately lead to compromised recommendation performance. And in some cases, the performance is even worse compared to single-modal models.

3 METHOD

In the previous section, we discussed the significance of Transformers in acquiring multi-modal representations and also pointed out the shortcomings of current methods when fusing multi-source information. To further improve the representation ability of Transformers in recommendation scenarios and enable collaborative learning from different modalities without conflicts during training, we propose two modules based on the Late Fusion framework: 1) **ID-aware Multi-modal Transformer**. We incorporate the ID features with modality features and perform fine-grained feature interactions within a single multi-modal Transformer. 2) **Online Distillation**. We use an online distillation framework to compute the recommendation classification loss for each input, leveraging the strong representation capacity of Transformers. This ensures that each sub-network captures distinct user preferences by optimizing corresponding loss. Besides, we introduce a distillation loss that facilitates on-the-fly mutual learning [8, 53] among the student networks. Figure 3 shows the overall framework of ODMT with the above two modules.

3.1 Notations

We define the set of users as $\mathcal{U} = \{u\}$ and the set of items as $\mathcal{A} = \{a\}$. For the each item a_j, we record its image, text and ID as a_j^v, a_j^t, and a_j^{id}, respectively. The i-th user interaction sequence S_i, is defined in chronological order as $S_i^m = \{a_{s_1}^m(i), a_{s_2}^m(i), ..., \}$,

where $a^m_{s_k}(i)$ represents the k-th interaction item, and m represents the types of the item, i.e., image, text, and ID.

3.2 Item Representation Learning

Feature Extractor. Given an item with different types (a^v, a^t, and a^{id}), we first use fixed visual and textual feature extractors (ViT and BERT) to obtain the corresponding fine-grained patch-level and token-level features, then we obtain the corresponding ID features from a learnable embedding table, the feature extraction process is summarized as follows:

$$E^v = \text{ViT}(a^v),\ E^t = \text{BERT}(a^t),\ E^{id} = \text{EmbeddingTable}(a^{id}) \quad (1)$$

where $E^v = [E^v_1; ...; E^v_{n_v}; E^v_{cls}] \in \mathbf{R}^{d_v \times (n_v+1)}$, $E^t = [E^t_{cls}; E^t_1; ...; E^t_{n_t}] \in \mathbf{R}^{d_t \times (n_t+1)}$ and $E^{id} \in \mathbf{R}^{d_{id}}$. d_v, d_t, and d_{id} are the visual feature dimension, textual feature dimension, and ID feature dimension respectively. n_v is the number of image patches and n_t is the number of word tokens. E_{cls} here represents the embedding of the special token "[CLS]".

Afterward, we use a simple feature transformation matrix to project each input feature into the same dimension d as $\tilde{E}^m = W^m \cdot E^m + b^m$, where $W^m \in \mathbf{R}^{d \times d_m}$, $b^m \in \mathbf{R}^d$, and $m \in \{v, t, id\}$.

ID-aware Multi-modal Transformer (IMT). In this part, we describe the proposed ID-aware Multi-modal Transformer (IMT) module, which consists of multiple standard Transformer layers. Different from traditional multi-modal Transformers designed for visual and textual features, our IMT module integrates the unique ID features in the recommendation system into Transformers. Our goal is to obtain a unified framework that transforms item embeddings from the original generic feature space to one that is suitable for recommendations (especially for modality features).

To achieve this, we first concatenate visual patch features, ID features, and textual token features together as $\tilde{E} = [\tilde{E}^v; \tilde{E}^{id}; \tilde{E}^t] \in \mathbf{R}^{d \times (n_v+n_t+3)}$, where $[\cdot]$ denotes the concatenation operation. Since there are no paddings for visual and ID features, we set the mask value for all visual and ID features to 0 and obtain the attention mask as $\tilde{M} \in \mathbf{R}^{(n_v+n_t+3)\times(n_v+n_t+3)}$. However, in the previous section of the discussion, we discovered that the ID features could influence the optimization direction of the model. To prevent the misleading influence of ID embeddings on modality embeddings, we make the following adjustments to the original attention mask matrix as $\tilde{M}[: n_v + 1, n_v + 1] = 1$, $\tilde{M}[n_t + 2 :, n_v + 1] = 1$. This ensures that the ID embeddings can attend to the modality embeddings, while the modality embeddings cannot attend to the ID embeddings

Similar to the traditional Transformer modeling process, once we have input feature $\tilde{E}$ and revised attention mask matrix $\tilde{M}$, we can feed them directly into the standard Transformer layer as:

$$\hat{E} = \text{IMT}(\tilde{E}, \tilde{M}; \Theta_{\text{IMT}}) \quad (2)$$

where Θ_{IMT} denotes all the learnable parameters in the IMT module and $\hat{E} = [\hat{E}^v; \hat{E}^{id}; \hat{E}^t] \in \mathbf{R}^{d \times (n_v+n_t+3)}$ denotes the encoded item representation. Then we use the "cls" embedding to represent the global feature for image and text. We do not need to include additional positional embeddings in the input embeddings, as the features extracted from the pre-trained models already contain positional information.

To obtain more powerful feature representations for the recommendation domain [24], we empirically employ separate two-layer DNNs with a LeakyRelu activation layer [40] for each output as:

$$\begin{aligned} D^k &= W^k_2 \cdot \text{LeakyRelu}(W^k_1 \cdot \hat{E}^m_{cls} + b^k_1) + b^k_2 \\ D^{id} &= W^{id}_2 \cdot \text{LeakyRelu}(W^{id}_1 \cdot \hat{E}^{id} + b^{id}_1) + b^{id}_2 \end{aligned} \quad (3)$$

where $W^k_i, W^{id}_i \in \mathbf{R}^{d \times d}$, $k \in \{v, t\}$ and $i \in \{1, 2\}$. D represents the final embedding of the item. In detail, D^m_k represents the k-th item embedding whose input is m, where $m \in \{v, t, id\}$.

ID Embedding Initialization. It is noteworthy that the initial embedding table for the ID features is typically randomly generated, which differs significantly from the text and image features extracted by large pre-trained models such as BERT [6] and ViT [7]. In the IMT module, a self-attention mechanism is utilized to compute the similarity between queries and keys. The discrepancy between the ID features and modality features can negatively impact the optimization of the IMT module. To address this, when $d_v = d_t$, we initialize the ID embedding table by averaging E^v_{cls} and E^t_{cls}, thus establishing $d_v = d_t = d_{id}$. When $d_v \neq d_t$, the ID embedding table is initialized using either the text or image features or either the concatenated text and image features, arbitrarily chosen.

3.3 User Sequence Modeling

In sequential recommendation tasks, user sequence features are generated from interacted items. The widely used SASRec [20] model employs a multi-head attention mechanism for user sequence modeling. We adopt SASRec as the backbone network to learn user sequence representations from three input types (image, text, and ID). The user behavior sequence with final item embeddings is represented as $\mathbf{S}^m = \{D^m_{s_1}, D^m_{s_2}, ..., D^m_{s_n}\}$. Using this sequence, we obtain the user sequence feature H^m as follows:

$$H^m = \text{SASRec}_m(\mathbf{S}^m; \Theta_{\text{SASRec}_m}) \quad (4)$$

where $H^m \in \mathbf{R}^d$ is the user preference feature, Θ_{SASRec_m} denotes all the learnable parameters in SASRec_m, m represents the input type.

3.4 Debiased Inbatch Loss

Following prior research [15, 46, 52], we adopt next-item prediction as the recommendation task, with negative log-softmax loss as the guiding loss function, which helps to bring user preference and the target item closer in the feature space. Given the user interaction sequence as $a_1 \rightarrow a_n$, the positive sample that needs to be predicted is a_{n+1}.

For computational efficiency, we utilize all the items from the user interaction sequences in the mini-batch as the candidate item sets. However, this approach leads to a distribution of items in the candidate item sets, known as the Matthew effect, where the majority of items are highly popular with a large number of interactions, causing popular items to become over-represented and leading to under-optimized performance for less popular items. To mitigate this effect, we debias the similarity computation results between users and items based on popularity [4, 43].

It is noteworthy to consider false negatives in in-batch sampling. When using items that the user has already interacted with as negative samples, the gradient descent direction of the model may

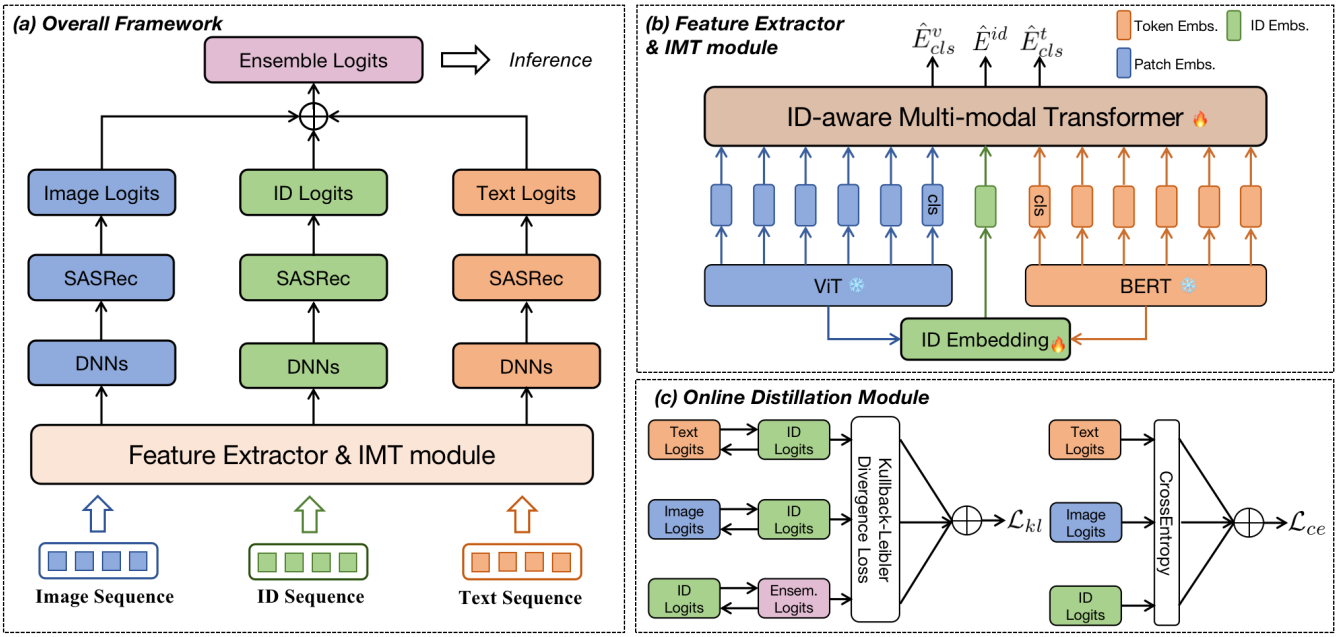


Figure 3: (a) Overall framework of our proposed ODMT model, which illustrates the forward computation flow based on modifications to the Late Fusion approach. (b) shows our proposed IMT module, and (c) represents our proposed Online Distillation module. Different colors represent the information flow of different features (ID, image, and text).

be confused. To address this issue, items in the candidate item set that overlap with the user's interaction sequence should be excluded when predicting the to-click item of a user.

In the batch training process, a set of B training instances is considered, where each instance corresponds to a user sequence and a target next item (positive sample). These instances are encoded as embedding representations $\{(H_{h_1}, D_{d_1}), (H_{h_2}, D_{d_2}), ..., (H_{h_B}, D_{d_B})\}$, where h_i and d_i represent the indices of the user sequence and target item of the i-th pair, respectively. Then we define the candidate item sets which exclude the items that overlap with h_i-th user sequence as $\mathcal{B}_{h_i}$. Finally, we adopt the cross-entropy loss as the objective function:

$$\mathcal{L}_{ce} = -\sum_{i=1}^{B} \log \frac{\exp(s(H_{h_i}, D_{d_i}))}{\exp(s(H_{h_i}, D_{d_i})) + \sum_{d_j \in \mathcal{B}_{h_i}} \exp(s(H_{h_i}, D_{d_j}))} \tag{5}$$

$$s(H_{h_i}, D_{d_i}) = H_{h_i} \cdot D_{d_i} - \log(pop(a_{d_i})) \tag{6}$$

where $pop(a_{d_i})$ represents the frequency of item a_{d_i} appearing in the training set.

3.5 Online Distillation

Similar to Late Fusion, we model different types of user sequences to calculate the similarity between user sequences and the target items as well as candidate items, obtaining logits that represent the user's interest distribution. However, different from Late Fusion, we treat each part as a student network branch and directly calculate the classification loss for each corresponding logit, rather than averaging multi-source logits and obtaining a classification loss. We believe that this independent loss calculation approach will alleviate conflicts between multiple features during the training process. In detail, we denote $\mathbf{z}^{\mathrm{m}}$ and $\mathbf{y}$ as the logits and ground truth, where $m \in \{v, t, id\}$. Late Fusion method obtains the final classification loss as $\mathcal{L}_{ce} = \text{cross\_entropy}((\mathbf{z}^e, \mathbf{y})$, where $\mathbf{z}^e$ is the ensemble logits as $\mathbf{z}^e = \frac{\mathbf{z}^v + \mathbf{z}^t + \mathbf{z}^{id}}{3}$. As for collaborative learning, we calculate classification loss as follows:

$$\mathcal{L}_{ce} = \mathcal{L}_{ce}^v + \mathcal{L}_{ce}^t + \mathcal{L}_{ce}^{id} \tag{7}$$

$$\mathcal{L}_{ce}^m = \text{cross\_entropy}(\mathbf{z}^m, \mathbf{y}),\ m \in \{v, t, id\} \tag{8}$$

In the knowledge distillation part, we calculate the distillation loss as follows:

$$\begin{aligned} \mathcal{L}_{kl} &= \mathcal{L}_{kl}^v + \mathcal{L}_{kl}^t + \mathcal{L}_{kl}^{id} \\ \mathcal{L}_{kl}^v &= \mathrm{T}^2 KL(\sigma(\mathbf{z}^{id}/\mathrm{T}), \sigma(\mathbf{z}^v/\mathrm{T})) \\ \mathcal{L}_{kl}^t &= \mathrm{T}^2 KL(\sigma(\mathbf{z}^{id}/\mathrm{T}), \sigma(\mathbf{z}^t/\mathrm{T})) \\ \mathcal{L}_{kl}^{id} &= \mathrm{T}^2 KL(\sigma(\mathbf{z}^e/\mathrm{T}), \sigma(\mathbf{z}^{id}/\mathrm{T})) \end{aligned} \tag{9}$$

where T is the temperature parameter, σ is the softmax operation, and $KL(p, q)$ means the KL divergence between the soften outputs p from teacher network and q from student network.

Because at the beginning of the model training, the predictions of each student network are not accurate enough, we need to decrease the weight of the distillation loss during the early training stage. Therefore, we adopt a time-dependent unsupervised ramp-up function $w(\alpha)$[22]. When the training epoch is 0, $w(\alpha)$ is 0. Then, $w(\alpha)$ increases exponentially as the training epoch progresses. When the training epoch reaches α, $w(\alpha)$ takes a value of 1. Then the final total loss is as follows:

$$\mathcal{L}_{total} = \mathcal{L}_{ce} + w(\alpha) \cdot \mathcal{L}_{kl} \tag{10}$$

Table 1: Statistics of all datasets used in our experiment. "#Inter." represents total user-item interactions, and "Avg. u" denotes the average user length.

Dataset	#Users	#Items	#Inter.	Avg. u	Density
Stream	100,000	19,683	687,487	6.875	0.000349
Arts	43,583	58,900	333,693	7.656	0.000130
Office	71,865	91,527	667,461	9.288	0.000101
H&M	50,000	61,042	606,922	12.138	0.000199

4 EXPERIMENTS

4.1 Datasets

We evaluate the performance of each method by using **four** datasets, Stream, Arts, Office, and H&M, which are obtained from **three** different platforms. The Stream dataset is a stream media dataset from the a video content platform that we have crawled by ourselves. Arts and Office are e-commerce datasets from the Amazon platform[1], which are publicly available and commonly used [14, 50, 58]. Arts dataset corresponds to the "Arts, Crafts and Sewing" category of Amazon review datasets, while Office represents the "Office Products" category. H&M is another e-commerce dataset from the H&M platform, which is a public competition dataset provided by Kaggle[2]. The diversity of datasets from different platforms helps to demonstrate the robustness of our proposed methods.

For the Stream dataset [3], we utilize the video cover and the concatenation of video tags and video titles to represent visual and textual information, respectively. For the other three datasets (Arts, Office, and H&M), we use the cover of the product to represent visual information. As for textual information, Arts and Office datasets use the concatenation of "title", "brand", "category", and "description". Different from them, the H&M dataset consists of the concatenation of "prod_name", "product_type_name", "product_group_name", "graphical_appearance_name", and "colour_group_name".

For all datasets, we utilize user sequences with more than 5 interactions and items that have completely matched textual and visual content. Additionally, we keep only the most recent 15 interaction records for each user. Table 1 presents the relevant statistical information of each dataset.

4.2 Evaluation Metric

To evaluate the performance of each model, we follow [15, 28] and adopt the commonly used metrics, Recall@k and NDCG@k (Normalized Discounted Cumulative Gain@k). We report the average results over all users in both metrics, and the higher value indicates better performance. Following [15, 28], we use the last interaction as the prediction, the second-to-last as validation, and the rest for training. We conduct hyper-parameter optimization on the validation set, and choose the combination of parameters that yields the highest Recall@10 as the optimal configuration.

[1]https://jmcauley.ucsd.edu/data/amazon/

[2]https://www.kaggle.com/competitions/h-and-m-personalized-fashion-recommendations

[3]This dataset is provided by the unpublished work [25]. If you require access to the relevant data, please refer to Acknowledgement section and contact the authors directly.

4.3 Implementation Details

To make a fair comparison, we reproduce all of the baselines by utilizing our pipeline framework. Our default loss function for all models is debiased in-batch cross-entropy loss. The visual features are extracted based on the "openai/clip-vit-base-patch32" [27] pre-trained model. The textual features with Chinese words are extracted by using the "hfl/chinese-roberta-wwm-ext" [5] pre-trained model and textual features with English words are extracted by using the "bert-base-uncased" [6] pre-trained model. We conduct the grid search for hyper-parameters, such as hidden size and learning rate. For the general sequential models, FDSA [53] and UniSRec [15], the search range for hidden sizes and learning rates are [128, 256, 512, 768] and [1e-3, 1e-4, 1e-5], respectively. Empirically, SASRec+EF and SASRec+LF both have two Transformer layers for text and image modalities, culminating in a total of four Transformer layers. In our model, we adopt two Transformer layers of IMT. In our experiments, we set the batch size to 128, which includes 128 sequences from different users per batch for training. For specific hyper-parameters unique to each baseline, such as the number of GRU layers in GRU4Rec and the selection of dilated convolution layers in NextItNet, we refer to the settings in RecBole [54].

4.4 Comparison with SOTA Methods

Based on the input types, we divide SOTA methods into two categories: 1) **general sequential recommendation models** that only take item ID information as input (e.g. GRU4Rec [12], SASRec [20], NextItNet [45]), and 2) **multi-modal sequential recommendation models** that take both item ID information and modality information (visual and textual) as input (e.g. FDSA [52], UniSRec [15], SASRec+EF, SASRec+LF): **(a). GRU4Rec** is a session-based recommendation algorithm that uses recurrent neural networks (GRUs) to model user behavior; **(b). SASRec** is a self-attention-based sequential recommendation algorithm that uses a multi-head self-attention mechanism to capture user preferences; **(c). NextItNet** is a neural network-based sequential recommendation algorithm that uses dilated convolutions to capture long-term dependencies between items; **(d). FDSA** is a feature-driven and self-attention-based sequential recommendation algorithm that uses feature-driven attention mechanisms to capture user preferences; **(e). UniSRec** is a universal sequence representation learning algorithm for recommendation that harnesses the descriptive text associated with an item to learn transferable representations across different domains and platforms; **(f). SASRec+EF (Our Extension)** is an extension of SASRec that takes id, text, and image as input and uses Transformer layers as the FST module with Early Fusion; **(g). SASRec+LF (Our Extension)** is an extension of SASRec that takes id, text, and image as input and uses Transformer layers as the FST module with Late Fusion.

For the vanilla UniSRec [15] and FDSA [53] models, only textual information is utilized. To make a fair comparison, we reproduce these models by incorporating image information, leveraging the inherent extensibility of the UniSRec and FDSA models.

Table 2 demonstrates the superiority of the **Sequential Model with Modality Feature** over the **General ID-based Sequential Model** across all datasets and evaluation metrics, which underscores the potential benefits of incorporating modality information

Table 2: Overall performance of our model and the baselines on four multi-modal recommendation datasets. Best performances are noted in bold, and the second-best are underlined.

Dataset	Metrics	General ID-based Sequential Model			Sequential Model with Modality Feature					Improv.
		GRU4Rec	SASRec	NextItNet	FDSA	UniSRec	SASRec+EF	SASRec+LF	ODMT	
Stream	Recall@10	0.0914	0.0935	0.0845	0.0885	0.1067	0.1112	0.1090	**0.1194**	**7.373%**
	NDCG@10	0.0492	0.0507	0.0454	0.0476	0.0585	0.0611	0.0607	**0.0672**	**9.946%**
	Recall@20	0.1323	0.1344	0.1253	0.1287	0.1506	0.1572	0.1545	**0.1668**	**6.132%**
	NDCG@20	0.0595	0.0610	0.0557	0.0577	0.0696	0.0727	0.0721	**0.0791**	**8.863%**
Arts	Recall@10	0.0535	0.0617	0.0510	0.0640	0.0860	0.0962	0.0992	**0.1127**	**13.552%**
	NDCG@10	0.0380	0.0454	0.0363	0.0471	0.0612	0.0669	0.0709	**0.0787**	**11.127%**
	Recall@20	0.0703	0.0787	0.0661	0.0809	0.1095	0.1241	0.1264	**0.1410**	**11.601%**
	NDCG@20	0.0422	0.0497	0.0401	0.0514	0.0672	0.0739	0.0765	**0.0852**	**11.308%**
Office	Recall@10	0.0703	0.0769	0.0710	0.0816	0.0971	0.1068	0.1085	**0.1175**	**8.299%**
	NDCG@10	0.0542	0.0606	0.0533	0.0635	0.0758	0.0814	0.0830	**0.0893**	**7.558%**
	Recall@20	0.0844	0.0914	0.0852	0.0970	0.1154	0.1281	0.1302	**0.1408**	**8.200%**
	NDCG@20	0.0578	0.0642	0.0568	0.0674	0.0804	0.0867	0.0885	**0.0952**	**7.575%**
H&M	Recall@10	0.0380	0.0502	0.0275	0.0571	0.0877	0.1115	0.1138	**0.1235**	**8.522%**
	NDCG@10	0.0206	0.0296	0.0141	0.0334	0.0531	0.0682	0.0701	**0.0771**	**9.988%**
	Recall@20	0.0582	0.0705	0.0435	0.0830	0.1209	0.1490	0.1513	**0.1629**	**7.679%**
	NDCG@20	0.0257	0.0347	0.0181	0.0400	0.0615	0.0777	0.0796	**0.0870**	**9.368%**

to enhance recommendation accuracy. Notably, SASRec+EF and SASRec+LF within the **Sequential Model with Modality Feature** outperform UniSRec and FDSA in terms of Recall@10 and NDCG@10 scores, indicating that utilizing Transformers as FST module may lead to more effective item representation modeling and improves recommendation accuracy. Furthermore, the fusion strategy of Late Fusion is proven to result in better overall performance. Based on the SASRec+LF framework, our proposed ODMT model aims to achieve multi-source information representation learning in a unified manner, which leverages the strengths of contemporary multi-modal Transformer models and online distillation methods. Experimental results demonstrate that our proposed ODMT model can achieve better performance across all four datasets and four metrics, surpassing not only SASRec+LF but also all other baseline models.

Table 3: Ablation analysis results on two downstream datasets. "R@10" is short for Recall@10, and "N@10" is short for NDCG@10.

Dataset	Stream		Arts	
Method	**R@10**	**N@10**	**R@10**	**N@10**
Text Initialization	0.1191	0.0666	0.1107	0.0777
Image Initialization	0.1161	0.0647	0.1051	0.0738
w/o Initialization	0.1144	0.0638	0.1025	0.0722
w/o ID mask	0.1155	0.0646	0.1001	0.0700
w/o IMT (1)	0.1088	0.0607	0.1068	0.0752
w/o IMT (2)	0.1146	0.0636	0.1096	0.0770
w/o Online Distillation	0.1121	0.0626	0.1019	0.0723
w/o ID	0.1125	0.0623	0.1075	0.0747
ODMT (full framework)	**0.1194**	**0.0672**	**0.1127**	**0.0787**

4.5 Ablation Study

In this study, we conduct an analysis to evaluate the impact of each module on the final performance of our proposed ODMT model. To compare the performance of our model with other variants, we prepare **8** different models, including:

(1) Text Initialization, which initializes the ID embedding table using only textual features; **(2) Image Initialization**, which initializes the ID embedding table using only visual features; **(3) w/o Initialization**, which abandons initialization for the ID embedding table with text and image features; **(4) w/o ID mask**, which removes the limitation of modality features being unable to attend to ID features in the IMT module; **(5) w/o IMT (1)**, which replaces the two layers of IMT with two standard Transformer layers for text input and two standard Transformer layers for image input, with the depth of Transformers in both cases remaining the same; **(6) w/o IMT (2)**, which replaces the two layers of IMT with one standard Transformer layer for text input and one standard Transformer layer for image input, while keeping the total number of Transformers the same; **(7) w/o Online Distillation**, which uses a traditional Late Fusion loss calculation method and removes the distillation loss; **(8) w/o ID**, which removes the ID component in ODMT and replaces it with the corresponding ID-free version.

As shown in Table 3, each new component contributes to the final performance. In the ID initialization part, utilizing average features of both text and image modalities for initializing the ID embedding table results in the best performance, with text features following closely. However, random initialization of the ID embedding table has a detrimental impact on prediction results, particularly affecting the optimization of the IMT module. "w/o ID" shows an obvious performance reduction compared to full ODMT, which highlights the effective accommodation of both ID features and multi-modal features in our framework.

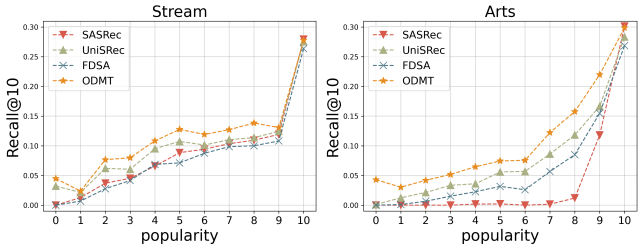


Figure 4: Performance comparison of sequential models regarding item popularity, where the x-axis represents different groups divided based on quantile statistics. Group 0 refers to items that have not appeared in the training set, while the item count is kept consistent across other groups.

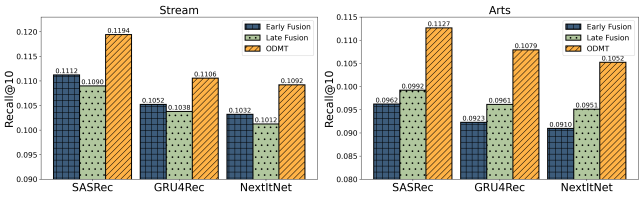


Figure 5: Performance comparison of sequential models with different backbones.

4.6 In-depth Analysis

4.6.1 Performance Comparison w.r.t Item Popularity. The item popularity distribution follows a Matthew effect, with a majority of users showing interest in only a small portion of items. Figure 4 indicates existing models predict popular items well but struggle with long-tail items. Moreover, Figure 4 illustrates the consistent improvement of our proposed method compared to the SOTA models in item groups with varying popularity. Notably, in item group 0, which consists of items not present in the training set, the ID-based SASRec fails to make accurate predictions. In contrast, our proposed model demonstrates effective mitigation of the cold-start problem, showcasing superior performance compared to other multi-modal sequential recommendation models.

4.6.2 Different Sequential Models as Backbone. Since our method is model-agnostic, we conduct experiments based on various sequential models to test the robustness. Figure 5 shows that ODMT can surpass other methods consistently. This highlights the generality of ODMT, which can be seamlessly integrated as a plug-and-play module into any general sequential model.

5 RELATED WORK

5.1 Multi-modal Recommendation

The development of computer vision [17], natural language processing [23], and multi-modal learning [16, 18, 19, 47] has provided better representations for heterogeneous data structures. Recently, multi-modal representations have been widely used in the field of recommendation systems, where collaborative filtering paradigms still dominate, with multi-modal features typically incorporated as side information in the model framework [33, 35, 36, 42, 58]. In the field of sequential recommendation, several studies have found that multi-modal features can also yield significant improvements [14, 15, 28]. Especially, [46] even achieved comparable results to traditional ID features using only multi-modal features, which underscores the significance of modal features in sequential recommendation and their substitutability for traditional ID features. We attribute this to the characteristics of sequential recommendation models, which heavily rely on item representations for modeling user preferences, unlike collaborative filtering which requires additional learning of user representations. Through extensive experimental exploration of multi-modal sequential recommendation, we observe the issue of conflicts between multi-modal and ID features during training. Based on these observations, we design two modules that leverage the characteristics of ID and multi-modal features to make them compatible and mutually beneficial.

5.2 Knowledge Distillation

Knowledge distillation [13, 44] aims to guide the learning of a student model by using a pretrained teacher model, allowing the student model to achieve better predictive performance with smaller model sizes. In recent years, online distillation has gained attention due to its end-to-end training strategy, which eliminates the need for a pre-trained teacher model. Unlike the traditional "teacher-student" paradigm of knowledge distillation, online distillation allows for mutual learning between all sub-networks [1, 53] or the use of ensemble methods to obtain a teacher output that combines multiple prediction results [8, 21, 59], which in turn guide the learning of all sub-networks. Online distillation typically requires that each sub-network can independently complete downstream prediction tasks (usually classification tasks) and that the prediction results of different sub-networks have diverse characteristics [3]. In sequential recommendation, we find that using ID features or multi-modal features (such as image or text) can independently complete recommendation predictions. Additionally, due to the heterogeneity of the input, the outputs of different features are also diverse. Based on these findings, we propose a multi-modal online distillation framework for sequential recommendation models.

6 CONCLUSION

This paper investigates the impact of item representation learning on downstream recommendation tasks, exploring disparities in information fusion at different stages. Empirical experiments show significant influence on recommendation performance. To enhance recommendation accuracy, the paper proposes two novel modules: ID-aware Multi-modal Transformer for feature interaction and online distillation training for multi-faceted user interest distribution and improved prediction robustness. Experimental results on four datasets demonstrate the effectiveness of these modules.

7 ACKNOWLEDGEMENT

This work is supported by the Advanced Research and Technology Innovation Centre (ARTIC), the National University of Singapore under Grant (project number: A-8000969-00-00). This research is also supported by the National Natural Science Foundation of China (9227010114) and the University Synergy Innovation Program of Anhui Province (GXXT-2022-040). Futhermore, we extend our gratitude to the Lab for Representation Learning at Westlake University (fajieyuan@westlake.edu.cn) for supporting dataset.

REFERENCES

[1] Rohan Anil, Gabriel Pereyra, Alexandre Passos, Robert Ormandi, George E Dahl, and Geoffrey E Hinton. 2018. Large scale distributed neural network training through online distillation. *arXiv preprint arXiv:1804.03235* (2018).

[2] Paul Baltescu, Haoyu Chen, Nikil Pancha, Andrew Zhai, Jure Leskovec, and Charles Rosenberg. 2022. ItemSage: Learning Product Embeddings for Shopping Recommendations at Pinterest. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2703–2711.

[3] Defang Chen, Jian-Ping Mei, Can Wang, Yan Feng, and Chun Chen. 2020. Online knowledge distillation with diverse peers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 3430–3437.

[4] Jin Chen, Defu Lian, Yucheng Li, Baoyun Wang, Kai Zheng, and Enhong Chen. 2022. Cache-Augmented Inbatch Importance Resampling for Training Recommender Retriever. *arXiv preprint arXiv:2205.14859* (2022).

[5] Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Shijin Wang, and Guoping Hu. 2020. Revisiting Pre-Trained Models for Chinese Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*. Association for Computational Linguistics, Online, 657–668. https://www.aclweb.org/anthology/2020.findings-emnlp.58

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* (2020).

[8] Qiushan Guo, Xinjiang Wang, Yichao Wu, Zhipeng Yu, Ding Liang, Xiaolin Hu, and Ping Luo. 2020. Online knowledge distillation via collaborative learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 11020–11029.

[9] Ruining He and Julian McAuley. 2016. VBPR: visual bayesian personalized ranking from implicit feedback. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 30.

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

[11] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*. 173–182.

[12] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939* (2015).

[13] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531* (2015).

[14] Yupeng Hou, Zhankui He, Julian McAuley, and Wayne Xin Zhao. 2022. Learning Vector-Quantized Item Representation for Transferable Sequential Recommenders. *arXiv preprint arXiv:2210.12316* (2022).

[15] Yupeng Hou, Shanlei Mu, Wayne Xin Zhao, Yaliang Li, Bolin Ding, and Ji-Rong Wen. 2022. Towards Universal Sequence Representation Learning for Recommender Systems. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 585–593.

[16] Wei Ji, Long Chen, Yinwei Wei, Yiming Wu, and Tat-Seng Chua. 2022. Mrtnet: Multi-resolution temporal network for video sentence grounding. *arXiv preprint arXiv:2212.13163* (2022).

[17] Wei Ji, Xi Li, Lina Wei, Fei Wu, and Yueting Zhuang. 2020. Context-aware graph label propagation network for saliency detection. *IEEE Transactions on Image Processing* 29 (2020), 8177–8186.

[18] Wei Ji, Renjie Liang, Lizi Liao, Hao Fei, and Fuli Feng. 2023. Partial Annotation-based Video Moment Retrieval via Iterative Learning. In *Proceedings of the 31th ACM international conference on Multimedia*.

[19] Wei Ji, Renjie Liang, Zhedong Zheng, Wenqiao Zhang, Shengyu Zhang, Juncheng Li, Mengze Li, and Tat-seng Chua. 2023. Are binary annotations sufficient? video moment retrieval via hierarchical uncertainty-based active learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 23013–23022.

[20] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*. IEEE, 197–206.

[21] Jangho Kim, Minsung Hyun, Inseop Chung, and Nojun Kwak. 2021. Feature fusion for online mutual knowledge distillation. In *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 4619–4625.

[22] Samuli Laine and Timo Aila. 2016. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242* (2016).

[23] Jingye Li, Kang Xu, Fei Li, Hao Fei, Yafeng Ren, and Donghong Ji. 2021. MRN: A locally and globally mention-based reasoning network for document-level relation extraction. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. 1359–1370.

[24] Fan Liu, Huilin Chen, Zhiyong Cheng, Anan Liu, Liqiang Nie, and Mohan Kankanhalli. 2022. Disentangled Multimodal Representation Learning for Recommendation. *IEEE Transactions on Multimedia* (2022).

[25] Yongxin Ni, Yu Cheng, Junjie Shan, Xiangyan Liu, Juncheng Fu, Youhua Li, Xiangnan He, Yongfeng Zhang, and Fajie Yuan. 2023. A Content-Driven Micro-Video Recommendation Dataset at Scale.

[26] Xingyu Pan, Yushuo Chen, Changxin Tian, Zihan Lin, Jinpeng Wang, He Hu, and Wayne Xin Zhao. 2022. Multimodal Meta-Learning for Cold-Start Sequential Recommendation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 3421–3430.

[27] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*. PMLR, 8748–8763.

[28] Ahmed Rashed, Shereen Elsayed, and Lars Schmidt-Thieme. 2022. CARCA: Context and Attribute-Aware Next-Item Recommendation via Cross-Attention. *arXiv preprint arXiv:2204.06519* (2022).

[29] 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 *Proceedings of the 28th ACM international conference on information and knowledge management*. 1441–1450.

[30] Jie Wang, Fajie Yuan, Mingyue Cheng, Joemon M Jose, Chenyun Yu, Beibei Kong, Zhijin Wang, Bo Hu, and Zang Li. 2022. TransRec: Learning Transferable Recommendation from Mixture-of-Modality Feedback. *arXiv preprint arXiv:2206.06190* (2022).

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

[32] Wei Wei, Chao Huang, Lianghao Xia, and Chuxu Zhang. 2023. Multi-Modal Self-Supervised Learning for Recommendation. *arXiv preprint arXiv:2302.10632* (2023).

[33] Yinwei Wei, Wenqi Liu, Fan Liu, Xiang Wang, Liqiang Nie, and Tat-Seng Chua. 2023. LightGT: A Light Graph Transformer for Multimedia Recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1508–1517.

[34] Yinwei Wei, Xiang Wang, Xiangnan He, Liqiang Nie, Yong Rui, and Tat-Seng Chua. 2021. Hierarchical user intent graph network for multimedia recommendation. *IEEE Transactions on Multimedia* 24 (2021), 2701–2712.

[35] Yinwei Wei, Xiang Wang, Liqiang Nie, Xiangnan He, and Tat-Seng Chua. 2020. Graph-refined convolutional network for multimedia recommendation with implicit feedback. In *Proceedings of the 28th ACM international conference on multimedia*. 3541–3549.

[36] 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 *Proceedings of the 27th ACM international conference on multimedia*. 1437–1445.

[37] Yinwei Wei, Xiang Wang, Liqiang Nie, Shaoyu Li, Dingxian Wang, and Tat-Seng Chua. 2022. Causal inference for knowledge graph based recommendation. *IEEE Transactions on Knowledge and Data Engineering* (2022).

[38] Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. 2021. Mm-rec: multimodal news recommendation. *arXiv preprint arXiv:2104.07407* (2021).

[39] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. 2019. Session-based recommendation with graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 346–353.

[40] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. 2015. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853* (2015).

[41] Jiajing Xu, Andrew Zhai, and Charles Rosenberg. 2022. Rethinking Personalized Ranking at Pinterest: An End-to-End Approach. In *Proceedings of the 16th ACM Conference on Recommender Systems*. 502–505.

[42] Zhengyi Yang, Xiangnan He, Jizhi Zhang, Jiancan Wu, Xin Xin, Jiawei Chen, and Xiang Wang. 2023. A Generic Learning Framework for Sequential Recommendation with Distribution Shifts. In *SIGIR*. 331–340.

[43] Xinyang Yi, Ji Yang, Lichan Hong, Derek Zhiyuan Cheng, Lukasz Heldt, Aditee Kumthekar, Zhe Zhao, Li Wei, and Ed Chi. 2019. Sampling-bias-corrected neural modeling for large corpus item recommendations. In *Proceedings of the 13th ACM Conference on Recommender Systems*. 269–277.

[44] Guanghao Yin, Wei Wang, Zehuan Yuan, Chuchu Han, Wei Ji, Shouqian Sun, and Changhu Wang. 2022. Content-variant reference image quality assessment via knowledge distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 3134–3142.

[45] Fajie Yuan, Alexandros Karatzoglou, Ioannis Arapakis, Joemon M Jose, and Xiangnan He. 2019. A simple convolutional generative network for next item recommendation. In *Proceedings of the twelfth ACM international conference on web search and data mining*. 582–590.

[46] Zheng Yuan, Fajie Yuan, Yu Song, Youhua Li, Junchen Fu, Fei Yang, Yunzhu Pan, and Yongxin Ni. 2023. Where to Go Next for Recommender Systems? ID-vs. Modality-based recommender models revisited. *arXiv preprint arXiv:2303.13835*

(2023).
[47] Ao Zhang, Hao Fei, Yuan Yao, Wei Ji, Li Li, Zhiyuan Liu, and Tat-Seng Chua. 2023. Transfer visual prompt generator across llms. *arXiv preprint arXiv:2305.01278* (2023).
[48] An Zhang, Wenchang Ma, Xiang Wang, and Tat-Seng Chua. 2022. Incorporating Bias-aware Margins into Contrastive Loss for Collaborative Filtering. In *NeurIPS*.
[49] An Zhang, Jingnan Zheng, Xiang Wang, Yancheng Yuan, and Tat seng Chua. 2023. Invariant Collaborative Filtering to Popularity Distribution Shift. In *WWW*.
[50] Jinghao Zhang, Yanqiao Zhu, Qiang Liu, Mengqi Zhang, Shu Wu, and Liang Wang. 2022. Latent Structure Mining with Contrastive Modality Fusion for Multimedia Recommendation. *IEEE Transactions on Knowledge and Data Engineering* (2022).
[51] Lingzi Zhang, Xin Zhou, and Zhiqi Shen. 2023. Multimodal Pre-training Framework for Sequential Recommendation via Contrastive Learning. *arXiv preprint arXiv:2303.11879* (2023).
[52] Tingting Zhang, Pengpeng Zhao, Yanchi Liu, Victor S Sheng, Jiajie Xu, Deqing Wang, Guanfeng Liu, Xiaofang Zhou, et al. 2019. Feature-level Deeper Self-Attention Network for Sequential Recommendation.. In *IJCAI*. 4320–4326.
[53] Ying Zhang, Tao Xiang, Timothy M Hospedales, and Huchuan Lu. 2018. Deep mutual learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4320–4328.
[54] Wayne Xin Zhao, Shanlei Mu, Yupeng Hou, Zihan Lin, Yushuo Chen, Xingyu Pan, Kaiyuan Li, Yujie Lu, Hui Wang, Changxin Tian, et al. 2021. Recbole: Towards a unified, comprehensive and efficient framework for recommendation algorithms. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 4653–4664.
[55] Hongyu Zhou, Xin Zhou, and Zhiqi Shen. 2023. Enhancing Dyadic Relations with Homogeneous Graphs for Multimodal Recommendation. *arXiv preprint arXiv:2301.12097* (2023).
[56] Hongyu Zhou, Xin Zhou, Zhiwei Zeng, Lingzi Zhang, and Zhiqi Shen. 2023. A Comprehensive Survey on Multimodal Recommender Systems: Taxonomy, Evaluation, and Future Directions. *arXiv preprint arXiv:2302.04473* (2023).
[57] Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. 2020. S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization. In *Proceedings of the 29th ACM international conference on information & knowledge management*. 1893–1902.
[58] Xin Zhou, Hongyu Zhou, Yong Liu, Zhiwei Zeng, Chunyan Miao, Pengwei Wang, Yuan You, and Feijun Jiang. 2022. Bootstrap latent representations for multi-modal recommendation. *arXiv preprint arXiv:2207.05969* (2022).
[59] Xiatian Zhu, Shaogang Gong, et al. 2018. Knowledge distillation by on-the-fly native ensemble. *Advances in neural information processing systems* 31 (2018).

A APPENDIX

A.1 Dataset Details

Figure 6 shows some samples of **four** datasets from **three** platforms, namely Stream, Amazon and H&M, respectively. In contrast to the majority of existing research that predominantly focuses on experiments conducted on Amazon datasets [14, 15, 24, 28], our study employs a diverse range of experimental datasets with the aim of validating the robustness of our proposed method.

As for the Stream dataset, it is assembled by gathering publicly accessible user comments on videos from the stream media platform between February 2020 and September 2022. Specifically, we first collect short videos from the homepage to ensure the channel diversity. Subsequently, more videos are included via crawling related videos from the associated page of each video during the first stage. Finally, we establish the user set by acquiring publisher IDs from the comment sections under all videos. Note that there are no privacy issues since all videos and user comments are publicly available.

A.2 Effect of Different Transformer Layers

We investigate the number of layers in the multi-modal Transformer of the IMT module, where the number of multi-heads corresponds to the number of Transformer layers. For comparison, we also explore the SASRec+LF method under the same conditions. Figure 7 indicates that 2 Transformer layers are sufficient to achieve good performance, which is our default setting. Comparing our model with SASRec+LF, ODMT consistently outperforms SASRec+LF across different numbers of Transformer layers.

A.3 Effect of Parameters in Knowledge Distillation

The online distillation part of our framework involves two important parameters: 1) **Temperature parameter (T)** serves as a scaling factor that controls the softness or hardness of predicted probabilities. It affects the balance between exploration and exploitation during the distillation process by controlling the level of smoothing or sharpening in the probability distributions. We set T = 0.5 on the Stream dataset, T = 0.3 on the Arts dataset, T = 0.4 on the Office dataset and T = 0.5 on the H&M dataset. 2) **Weight factor (α)** controls the weight of the distillation loss in the whole loss function. A larger value of this parameter results in a smaller weight for the distillation loss at a fixed epoch, and the weight increases exponentially with the epoch. This allows the model to prioritize learning from the distillation loss during the late stage of training, mitigating error accumulation from poor student model performance in the early stages. We set α = 50 on the Stream dataset, α = 20 on the Arts dataset, α = 40 on the Office dataset, and α = 60 on the H&M dataset. Table 4 illustrates the impact of the two parameters on the experimental results on Stream and Arts datasets.

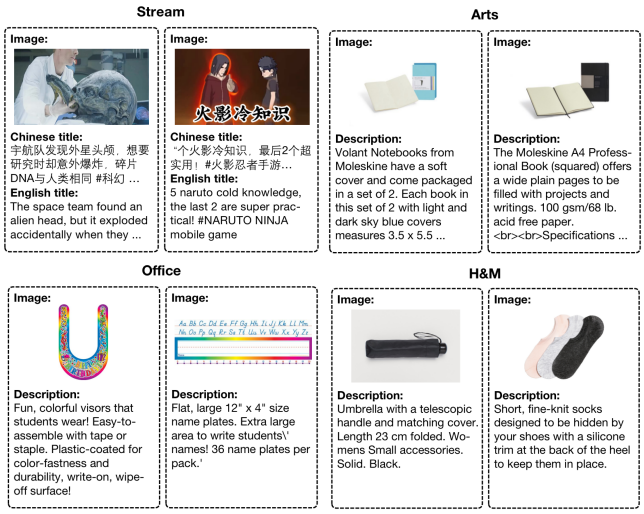


Figure 6: Selected sample cases from four different datasets.

Table 4: Performance comparison of different hyper-parameters settings in the online distillation module. "None" denotes the weight of distillation loss is 0, primarily used for control purposes in comparison with other parameters. "R@10" is short for Recall@10, "N@10" is short for NDCG@10. The best performance is highlighted in bold.

Stream					
T (α=50)	R@10	N@10	α (T=0.5)	R@10	N@10
None	0.1159	0.0645	None	0.1159	0.0645
0.1	0.1162	0.0648	10	0.1177	0.0654
0.2	0.1159	0.0650	20	0.1177	0.0659
0.3	0.1171	0.0652	30	0.1186	0.0662
0.4	0.1183	0.0661	40	0.1179	0.0660
0.5	**0.1194**	**0.0672**	**50**	**0.1194**	**0.0672**
0.6	0.1194	0.0669	60	0.1178	0.0657
Arts					
T (α=20)	R@10	N@10	α (T=0.3)	R@10	N@10
None	0.1091	0.0766	None	0.1091	0.0766
0.1	0.1120	0.0784	10	0.1120	0.0783
0.2	0.1105	0.0771	**20**	**0.1127**	**0.0787**
0.3	**0.1127**	**0.0787**	30	0.1113	0.0777
0.4	0.1120	0.0785	40	0.1103	0.0772
0.5	0.1091	0.0761	50	0.1118	0.0780
0.6	0.1058	0.0738	60	0.1107	0.0776

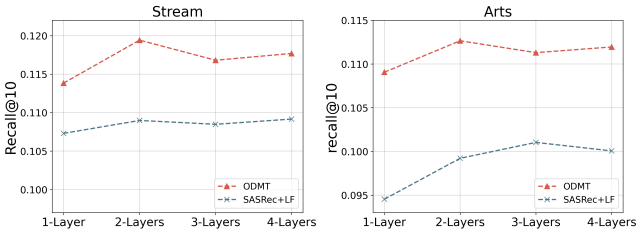


Figure 7: Performance comparison of different numbers of Transformer layers.