



# ROMA: Recommendation-Oriented Language Model Adaptation Using Multi-Modal Multi-Domain Item Sequences

Xingyu Lu<sup>†</sup>  
Tsinghua Shenzhen International  
Graduate School, Tsinghua University  
Shenzhen, China  
luxy22@mails.tsinghua.edu.cn

Jinpeng Wang<sup>†</sup>  
Tsinghua Shenzhen International  
Graduate School, Tsinghua University  
Shenzhen, China  
wjp20@mails.tsinghua.edu.cn

Jieming Zhu<sup>‡</sup>  
Huawei Noah's Ark Lab  
Shenzhen, China  
jiemingzhu@ieee.org

Zhicheng Zhang  
Tsinghua Shenzhen International  
Graduate School, Tsinghua University  
Shenzhen, China  
zhang-zc24@mails.tsinghua.edu.cn

Deqing Zou  
Tsinghua Shenzhen International  
Graduate School, Tsinghua University  
Shenzhen, China  
zdq23@mails.tsinghua.edu.cn

Hai-Tao Zheng<sup>‡</sup>  
Tsinghua Shenzhen International  
Graduate School, Tsinghua University  
Shenzhen, China  
Peng Cheng Laboratory  
Shenzhen, China  
zheng.haitao@sz.tsinghua.edu.cn

Shu-Tao Xia  
Tsinghua Shenzhen International  
Graduate School, Tsinghua University  
Shenzhen, China  
Peng Cheng Laboratory  
Shenzhen, China  
xiast@sz.tsinghua.edu.cn

Rui Zhang  
School of Computer Science & Tech,  
Huazhong University of Science and  
Technology (www.ruizhang.info)  
Wuhan, China  
rayteam@yeah.net

## Abstract

Sequential recommendation (SR) aims to capture dynamic user preferences from users' historical behaviors. Recently, benefiting from astonishing understanding ability of pre-trained language models (PLMs), text-enhanced sequential recommender becomes a promising direction, which employs PLMs to extract semantic information for user/item representation. Although promising in improving performance and transferability, few existing text-enhanced SR studies have analyzed the differences between PLMs and recommenders, restricting the ability of PLMs for recommendation. In this paper, we make an in-depth comparison and conclude their discrepancies in representation and knowledge level, respectively, caused by different multi-modal content and task-oriented capabilities. Based on this, we propose a **Recommendation-Oriented Language Model Adaptation** framework (named **ROMA**) using multi-modal multi-domain item sequences. To empower PLMs with a rational understanding of user/item modeling and the recommendation task, ROMA partitions a PLM into bottom and top layers, respectively, allowing representation-level and task-level adaptation with

elaborately designed architectures, transferring strategy and learning framework. Our experimental results on public benchmarks demonstrate the effectiveness and transferability of our framework. Additionally, we showcase the application value of ROMA on the recommender system of Huawei's AppGallery through online A/B testing, which shows significant improvements in online metrics.

## CCS Concepts

• **Information systems** → **Recommender systems; Multimedia information systems; Personalization.**

## Keywords

Sequential Recommendation, Pre-trained Language Model, Universal Representation Learning

## ACM Reference Format:

Xingyu Lu<sup>†</sup>, Jinpeng Wang<sup>†</sup>, Jieming Zhu<sup>‡</sup>, Zhicheng Zhang, Deqing Zou, Hai-Tao Zheng<sup>‡</sup>, Shu-Tao Xia, and Rui Zhang. 2025. ROMA: Recommendation-Oriented Language Model Adaptation Using Multi-Modal Multi-Domain Item Sequences. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2 (KDD '25)*, August 3–7, 2025, Toronto, ON, Canada. ACM, New York, NY, USA, 12 pages. <https://doi.org/10.1145/3711896.3737262>

<sup>†</sup> Equal Contribution.

<sup>‡</sup> Corresponding Authors: Hai-Tao Zheng and Jieming Zhu.



This work is licensed under a Creative Commons Attribution 4.0 International License. *KDD '25, Toronto, ON, Canada*

© 2025 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-1454-2/2025/08

<https://doi.org/10.1145/3711896.3737262>

## 1 INTRODUCTION

The goal of Sequential Recommendation (SR) [37, 51] is to suggest items (e.g., products) to users via understanding their preferences behind historical behaviors, which plays an active role in the web industry and has attracted widespread research attention. Learning

accurate representations to depict users and items is critical for SR. Traditional ID-based sequential recommenders only take IDs as input, relying on the collaborative signals between users and items. Although simple, ID-based approaches generally lack cross-domain generalizability due to the independent ID sets among different scenarios, which makes the knowledge hard to transfer. Besides, the long-tailed distribution of user-item interaction and the cold-start issue can threaten the robustness of representation [41, 67].

Recently, the success of pre-trained language models (PLMs) [10, 38] prompt some researches in textual-enhanced SR to learn transferable representations with enriched item textual attributes. Existing PLM adaption methods for recommendation can be divided into representation adaption and model adaption. Representation adaption approaches [19, 20, 50] pre-encode items into feature embedding as side information for recommenders. More holistically, model adaption approaches [32, 39, 70] unify task formulation of recommendation with NLP tasks by converting items into sentences and directly adapt PLMs as content-based recommenders. Despite the promising progress, few researches have noticed the disparities between PLMs and recommenders, which results in the following limitations: (1) Frozen modality encoders used by representation adaption works can remove specific attribute features of items, which is important to navigate user interests, and only contains coarser semantic. (2) Works adapting PLMs as recommenders simply replace an item's *ID* token with its *text* tokens or indices, superficial unification on the input formation overlooks internal differences. Without adaptations according to task characteristics, the performance of PLMs can be curtailed on recommendation tasks.

In this paper, we delineate the disparities between PLMs and SR systems into two facets: representation learning and task-related knowledge. To be specific, representation learning differences highlight the variances of input content between PLMs and recommenders. On the modality level, item images are vital [49, 55] for recommenders to attract users, but PLMs only take text as input. Even if centered on text, the distribution of item descriptions exhibit significant differences with the training corpus of PLMs. A detailed analysis of divergence in part-of-speech distribution between two kind of corpus is provided in the Appendix A.1. The task knowledge facet underscores differences inherent in PLM's pre-training task and recommender's behavior prediction task. Language modeling cares about the linguistic and semantic knowledge of natural language, it does not infer tokens based on user preferences like a recommender. Besides, pre-training of language models takes large-scale corpora to model various domains in a unified manner, while domain-specific patterns are especially important for sequential recommender. Thus, there should be variations in the training paradigms of language modeling and sequential recommendation.

Drawing from the above contrast, this paper is devoted to PLMs for recommendation from representation learning and task knowledge level, trying to improve the universality of adapted PLMs. We summarize the following three challenges for recommendation-oriented PLM adaption: (1) Multi-modal representation learning. Apart from adaption to recommendation context, PLMs are required to utilize image for item modeling for better alignment with actual recommendation scenarios. (2) Domain generality-speciality modeling. Distinct from the general language pre-training, recommendation systems are usually divided into multiple domains for

various item categories and user intentions. Modeling generalities and specialties among domains is necessary to avoid negative transfer. (3) Knowledge balance. Inherited semantic understanding of PLMs is crucial for item modeling. While PLMs further learn behavior prediction from interaction data, it is essential to balance two abilities to avoid catastrophic forgetting and insufficient learning.

We introduce our Recommendation-oriented Pre-training framework for Universal Multi-modal Sequence Representation, ROMA: (1) In light of existing PLM knowledge distribution analysis, ROMA takes a **partition strategy** to innovatively disentangle a language model into the bottom and top layers to balance the inherent language understanding with newly acquired recommendation/modal-ity skills: The bottom and top layers respectively conquer the multi-modal representation learning challenge and generality-speciality modeling challenge. (2) Based on the structure partition, we further incorporate Mixture-of-Adapters (**MoA**) **modules with diverse routing strategies** to address different challenges: For bottom layers, MoA modules with soft routing strategy are deployed to tackle the multi-modal representation learning challenge and adapt PLM to the recommendation context with vision-assisted masked language modeling task. For top layers, we adopt MoA modules with elaborately designed routing strategy for generality-speciality modeling: During pre-training, the routing strategy is set to hard style to assign private adapters to model domain specialty. For fine-tuning, these specialties are adaptively transferred to downstream domains by switching the routing strategy to the soft style. These MoA modules assist top layers to utilize multi-modal representations from the bottom layers for recommendation tasks. As a result, by assigning tasks to appropriate layers and incorporating strategic MoA architectures, ROMA transforms the PLM into a sequential recommender with multi-modal understanding and enhanced general recommendation capabilities.

To evaluate the effectiveness of ROMA, we conduct extensive experiments on multiple public benchmarks. The experimental results show that ROMA reaches 12.28% average performance improvements. We further validate the application value of ROMA in the Huawei App Store: the online A/B test results show that ROMA achieves a 7.29% increase in average DTR (Download-Through Rate) and a 3.17% increase in ECPM (Effective Cost Per Mille in terms of revenue) for advertisements, which are significant improvements in online metrics. Therefore, ROMA has been used to serve the main traffic of Huawei's App Store recommender system.

In summary, our work makes the following contributions:

- We thoroughly analyze the divergences between PLMs and sequential recommenders, identifying three challenges of adapting a PLM as a universal multi-modal sequential recommender: Modality adaptation, domain transfer and knowledge balance.
- To overcome these challenges, we propose ROMA, which includes a model partition strategy based on knowledge distribution of PLM and two stages with pertinent tasks to mitigate hierarchical divergences. ROMA is the first attempt to address challenges including multi-modal representation learning, knowledge balance and multi-domain transfer for recommendation by fusing MoAs with diverse routing strategies into PLM.
- Extensive experiment results on research benchmarks show that ROMA outperforms all state-of-art baselines and expresses superior generality. Analysis experiment prove ROMA to be a

practical solution for cold-start items and data sparsity. Our code is available at: <https://github.com/Nipers/ROMA>.

## 2 RELATED WORKS

### 2.1 Sequential Recommendation

Sequential Recommendation (SR) aims to learn patterns and characteristics of historical interactions for the prediction of the next interacted item. In order to improve modeling ability of recommender towards item features and user preferences, researchers in SR areas have made a series of innovations around model architecture and training methods. From model architecture perspective, SR models have transitioned from nascent stages of matrix factorization techniques [16, 40] to the contemporary neural architectures like CNN [45, 61], RNN [18, 34], MLP [28, 74], and Transformer architecture [23, 44]. From learning paradigm perspective, tasks like temporal-aware learning [21, 47], self-supervised contrastive objectives [6, 57, 73] are developed to model dynamic and behavior patterns, interest modeling [4, 66] is a popular methodology, in which user preferences are usually implemented by attention or clustering. Recently, technologies in frequency domain [12, 74] are also studied to assist sequential recommendation.

A pertinent topic in the SR is the debate over ID and modality information [62]. ID-based methods possess high efficiency but performs sub-optimally in cold-start and few-shot scenarios due to poor transferability. Existing work such as [36] and [9] attempts to leverage PLMs for improved item representation learning. In contrast, our work aims to adapt PLMs for user-item contrastive learning, thus producing both user and item representations.

### 2.2 Multi-Modal Recommendation

The purpose of Multi-Modal Recommendation (MMR) is to improve the recommendation efficacy and generality [30]. In areas like news recommendation [59, 65], video recommendation [2, 25] and personalized multimodal recommendation [43], visual/textual content is pivotal for matching user desires. Hence, As multi-modal representation learning advances, more modalities are integrated into SR systems. Text has been introduced into SR since the matrix factorization era [17]. Subsequently, a series of works utilize pre-trained modality models to encode item text [64] or image [28].

Early methods like UniSRec [20] and MissRec [50] initialize item embedding with representation from pre-trained models. VQ-Rec [19] takes text representations to characterize item affinity. These methods still employ encoders like two-layer Transformer as recommenders. More recent efforts have sought to incorporate PLM in a more comprehensive way within the SR system. A recent survey [31] provides a detailed summary of above works. Recformer [27] employs natural language descriptions for items and sequences, thus leveraging the entire PLM as a recommender. TASTE [32] also translates items into full text and utilizes PLM's matching ability for ranking, which alleviates cold start problems. However, some research [49, 50] has proven that a wider range of modality information, such as visual characteristics [55], is essential in some scenarios and should not simply be duplicated, while the above methods are all text-only recommenders. Furthermore, they directly take original LLM structure as recommenders, lacking targeted alignment strategy for multi-modal SR. In contrast, ROMA adapts

a PLM as a general multi-modal recommender through a carefully designed model structure and hierarchical adaptation strategy.

There are also methods like TIGER [39], LC-Rec [70] and HSTU [63] that apply extremely large LMs for generative SR, but limited by performance and efficacy, we do not involve them in this research.

### 2.3 Transfer Learning for Recommendation

The target of transfer learning for recommendation is to overcome data sparsity and cold-start problem [11, 35], which is widespread for RS. Early works are mainly about cross- [3] and multi-domain recommendation [54], relying on shared information among domains like overlapped users, items [7, 60] or shared attributes [56] as transfer medium, to mitigate dependence on shared data. Some works [3, 71] design graph architecture and data augmentation to transfer knowledge among domains, but ID-based transfer methods heavily rely on collaborative signals and lack interpretability.

Inspired by the boom of the pre-training paradigm in natural language processing [29, 69] and computer vision [14, 24], several works [20, 27, 50] try to adopt this paradigm into SR to learn general features for sequences and items in different domains. Prevalent customary formation of the "pre-train and fine-tune" paradigm splits all domains into upstream and downstream by data volume. Models are initially pre-trained over a range of upstream domains and subsequently refined using downstream data. Our work adheres to this paradigm, while we make special architecture modifications to enable the commonality-specialty modeling among multiple domains, thereby avoiding negative transfer and domain conflicts.

## 3 METHODOLOGY

In this section, we present the proposed Recommendation-oriented pre-training framework ROMA, which can effectively generate multi-modal user/item representations with excellent transferability towards cold-start items and various domains.

### 3.1 Problem Formulation and Method Overview

Given a user  $u$ 's historical interaction sequence  $s$  with  $T$  interactions denoted as  $[i_1, i_2, \dots, i_T]$ , where  $i_j$  denotes the  $j$ -th interacted item, a sequential recommender manage to predict his/her next interacted item  $i_{T+1}$  from item set  $I$  by calculating the probability scores between users and items. Under the multi-modal recommendation scenario, there are item features across different modalities. Without loss of generality, we involve the most common modalities, text and image to enhance SR in this paper. So in our setting, each item  $i$  is equipped with a dictionary  $D_i$  containing several attribute-value pairs  $\{(k_1, v_1), (k_2, v_2), \dots, (k_n, v_n)\}$ ,  $k$  denotes the attribute's name (e.g., Size, Name, Appearance) and  $v$  denotes the attribute's content (e.g., "Medium", "T-Shirt", a photo of T-Shirt),  $v$  takes the form of image or text and  $k$  is described by text only.

ROMA includes a model partition strategy and two learning stages. We first elaborate in depth on the motivation and objectives of our model partition strategy in Section 3.2.1, then we describe our model architecture in Section 3.2.2. Finally we introduce our pertinently designed tasks in Section 3.3 and two stages of ROMA, pre-training and fine-tuning in Section 3.4 to show how we make adaption on PLM for general sequence representation in SR.

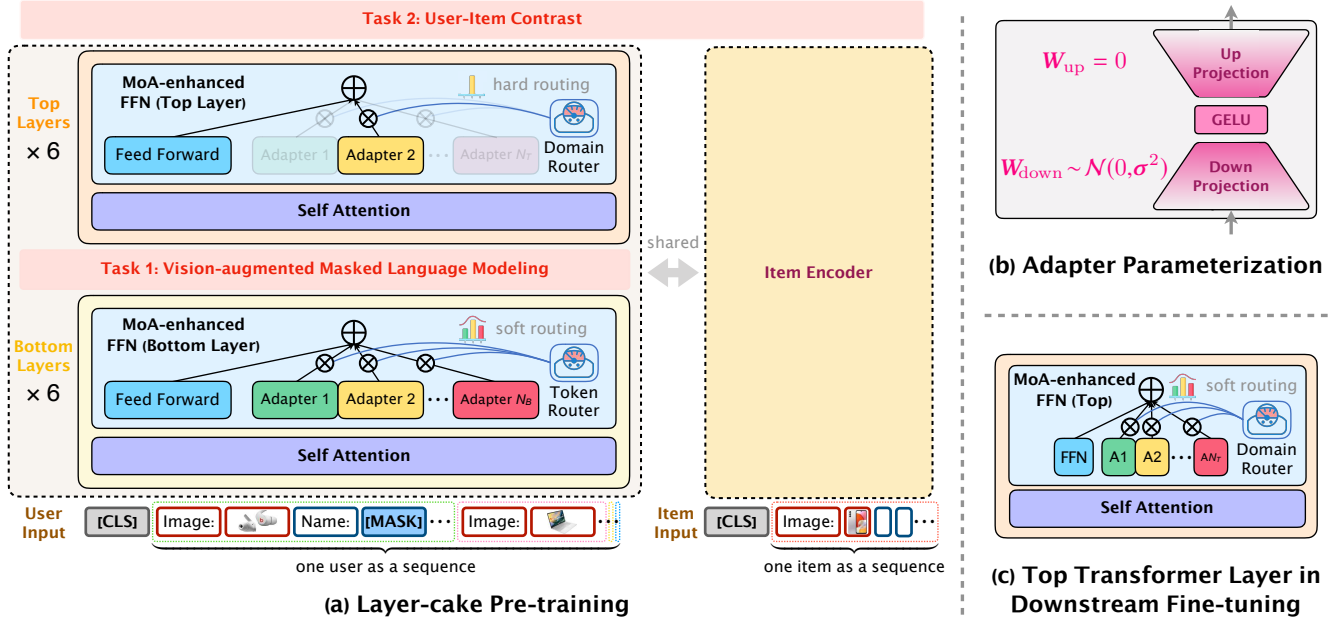


Figure 1: ROMA includes one model partition strategy and two learning stages, pre-training and fine-tuning. The partition strategy divides the original language model into bottom and top parts like a layer cake. The pre-training stage adapts PLM to multi-modal recommendation context with VA-MLM task and models domain generality and specialty with User-Item Contrast task. The fine-tuning stage adaptively transfers knowledge from upstream domains to specific target domains.

### 3.2 Partitioned Model Architecture

**3.2.1 Task-oriented Partition Strategy.** To efficiently inject and balance knowledge, the first step of ROMA is to divide the PLM into different regions to respectively conquer two sub-tasks of SR: user/item representation learning and behavior prediction.

Our partition strategy is designed according to PLM’s knowledge characteristic: A taxonomy [15] divides knowledge in PLMs into linguistic and world knowledge. The former incorporates information at lexicon, phrases, and syntax levels, while the latter includes commonsense and facts of the actual world. Several analyses [22, 46] point out that PLMs encode linguistic information at the bottom layers, and underlying semantic features at the top layers.

We first assign the user/item representation task to the bottom layers of PLM, since representation learning focuses on item categories, attribute contents, and inner relations among attributes, which is more related to shallow linguistic knowledge. Then we assign the behavior prediction task to the top layers since behavior prediction depends on underlying causal relations in item sequences which belongs to the logical reasoning in world knowledge.

**3.2.2 Model Architecture.** Based on the partitioning strategy, we make corresponding adjustments to the original language model structure to address the discrepancies at different levels. As shown in Figure 1, ROMA consists of below components:

- **Embedding Layer.** We extend the input embedding layer of BERT [10] to a multi-modal version. Our specific modifications are as follows: (1) Apart from original text tokens, we introduce image tokens of each item encoded by frozen pre-trained vision encoder to build token embedding  $E_t$ . (2) Apart from original

token position embedding  $E_p$ , we introduce item position embedding  $E_{ip}$  like [27] to indicate item’s temporal order in sequence. (3) We adopt attribute type embedding  $E_{at}$  to distinguish different item attributes. For example, tokens of the *title* have different attribute type embedding with tokens of the *image*.

Given a user sequence  $s$  or item  $i$ , we construct their input token sequences  $X_u$  and  $X_i$  in the same way, which are attribute tokens of items following the [CLS] token, the input representation  $h_x^0$  for each token is the sum of its four kinds of embedding.

$$X_u = [[CLS], \tilde{D}_T, \dots, \tilde{D}_1], \quad X_i = [[CLS], \tilde{D}_i], \quad (1)$$

$$h_x^0 = E_t(x) + E_p(x) + E_{ip}(x) + E_{at}(x), \quad (2)$$

where item  $i$ ’s attribute tokens  $\tilde{D}$  are the concatenation of all attribute names and values in  $D_i$  and  $x$  is a token in  $X$ . For items with no image, we take a specified token to represent void image.

- **MoA-Enhanced Transformer Layer.** We integrate a Mixture-of-Adapters (MoA) design into PLM’s original Transformer layers. After adjustment, each layer of our model is composed of a multi-head self-attention (MSA) module [48] and a *MoA-enhanced* feed-forward network (MoA-FFN). The forward process of the  $l$ -th layer for illustration is defined by

$$\tilde{H}^l = \text{LN}(\text{MSA}(H^l) + H^l), \quad (3)$$

$$H^{l+1} = \text{LN}(\text{MoA-FFN}(\tilde{H}^l) + \tilde{H}^l), \quad (4)$$

where  $\text{LN}(\cdot)$  denotes the layer normalization operator,  $H^l$  and  $H^{l+1}$  denote the input and output tensors of the  $l$ -th layer. Following common MoA designs [26, 42], our MoA module includes a gated router  $R$  and several adapters, where each adapter

$\phi$  consists of a linear down-projection, the GELU activation, and a linear up-projection, as shown in Figure 1(b). The gated router  $R$  is implemented by a linear projection  $W_g$  and a softmax layer to adaptively assign a weight for every adapter. Equation (5) describes two kinds of MoA with different routing strategies.

$$R(x) = \text{softmax}(W_g x), \quad (5)$$

$$\text{MoA}(x) = \begin{cases} \sum_{n=1}^N R(x)_n \cdot \phi_n(x), & \text{soft style;} \\ \phi_i(x), & \text{hard style.} \end{cases} \quad (6)$$

The MoA-enhanced feed-forward network is defined by

$$\text{MoA-FFN}(x) = \text{FFN}(x) + \text{MoA}(x). \quad (7)$$

FFN refers to FFN modules of the original Transformer layer. During pre-training stage, since the bottom and top layers are assigned with different functionalities (Section 3.2.1), they are equipped with different MoA designs: (1) Bottom layers mitigate the modality low-level linguistic gap and the multi-modality gap, and the routing mechanism of MoA is set to the **soft** style. (2) Top layers are expected to mitigate the multi-domain gap for recommendation. We assign one adapter to each pre-training domain with the **hard** style, which captures domain-specific knowledge to model specialties. During fine-tuning stage, we switch the **hard** routing mechanism of the top layer adapters to the **soft** mechanism to enable flexible knowledge transfer.

- **User/Item Representation.** As described in Equation (1), the input tokens of user/item are composed of [CLS] and attribute tokens from one or several flattened item attribution dictionaries. After encoding with all MOA-FFN layers, we gain a sequence of token representations  $H = [h_{CLS}, h_{t_1}, \dots, h_{t_l}]$ . Here we follow document retrieval research to pool the representation sequence by taking  $h_{CLS}$  as final user representation  $h_u$  or item representation  $h_i$ . In this way, we unify user/item modeling with sequence representation. With obtained representations, we apply cosine similarity to calculate the probability score between  $u$  and  $i$ :

$$\cos(u, i) = \frac{h_u^T h_i}{\|h_u\| \cdot \|h_i\|}. \quad (8)$$

### 3.3 Training Tasks for Bottom and Top Layers

In this section, we introduce two tasks specifically used for the bottom and top layers. Based on the analysis in Section 3.2, two tasks, VA-MLM and UIC, are assigned to bottom and top layers respectively to overcome corresponding challenges in user/item modeling and behavior prediction.

- **VA-MLM Task for Bottom Layers.** Since textual attributes comprise the majority of item properties, and images are subject to instances of absence, we regard visual information as supplementary to the textual data. We propose Vision-Augmented Masked Language Modeling (VA-MLM) task for user/item modeling. Particularly, we adopt the MLM task of BERT to choose 15% word tokens at random while keeping image tokens unchanged, a chosen word token could be replaced by (1) [MASK] token with 80% likelihood, (2) a random word token with 10% likelihood, (3) original token with 10% likelihood. The VA-MLM loss

is formulated as:

$$p = \text{softmax}(lm\_head(h_x^b)), \quad (9)$$

$$\mathcal{L}_{\text{VA-MLM}} = -\lambda \sum_{i=0}^{|\mathcal{V}|} y_i \log(p_i), \quad (10)$$

where  $lm\_head$  is a two-layer MLP,  $\mathcal{V}$  is the vocabulary and  $\lambda$  is the weight for VA-MLM loss, the hidden state  $h_x^b$  of word token  $x$  comes from **bottom** layers so that VA-MLM only trains parameters in these layers. Semantic of the image is conducted by attention mechanism to assist word token prediction, VA-MLM thereby introduces vision modality into item semantic modeling and adapts PLM to multi-modal recommendation context.

- **UIC Task for Top Layers.** Following precedent works [20, 27, 50], we adopt User-Item Contrastive (UIC) task to teach PLM to predict user behavior. The positive sample for user  $u$  is the real next item  $i^+$ . The negative sample set  $\mathcal{N}$  is a subset of  $I$ , different sampling strategies are used in our training pipeline.

$$\mathcal{L}_{\text{UIC}} = -\log \frac{\exp(\cos(u, i^+)/\tau)}{\sum_{i \in \mathcal{N}} \exp(\cos(u, i)/\tau)}. \quad (11)$$

where  $\cos(u, i)$  is the similarity score of  $u$  and  $i$  defined in Equation (8) and  $\tau$  is the temperature coefficient. The UIC loss is calculated with the output of the **top** layers so that parameters in these layers can concentrate on behavior prediction.

### 3.4 Training Pipeline

The pipeline of ROMA is composed of pre-training and fine-tuning stage. In two stages, the training task and architecture for bottom layers are identical, but specialized transition of routing mechanism is deployed for MoAs top layers to mitigate negative transfer.

- **Upstream Pre-training.** In this stage, data from all upstream domains is mixed to train the model. Since items from different domains have no overlap, in-batch negatives have a small probability to be false-negative, we adopt in-batch negatives for the UIC task. During pre-training, in top layers, with **hard**-style router, data from one specific domain only goes through adapter private to this domain, which turns the FFN modules into shared parameters for domain generalities and turns newly added adapters into private parameters for each domain.
- **Downstream Fine-tuning.** Considering similar divergences between upstream and downstream domains, we continue to use the VA-MLM and UIC tasks to fine-tune model. However, since training data for fine-tuning only comes from target domain, in-batch negatives are more likely to be popular items, which causes more false negatives. While encoding all items as negatives by step is impractical due to the calculation consumption. For the balance between efficiency and precise supervision, in the initial few epochs, we randomly select a subset of uninteracted items as UIC negatives. After the model converges, we generate [CLS] embedding for all items to construct an item embedding table. With the frozen table, no more encoding operation is required, making it efficient to take all uninteracted items as negatives for better performance. It can also speed up the inference process.

As we mentioned before, we convert the routing strategy of top MoAs to avoid negative transfer: We prompt the router  $R$  to

adaptively assign weights for each adapter. With suitable weight, knowledge from beneficial upstream domains can be transferred to downstream domains, as we will show in Section 5.3.

## 4 EXPERIMENTS

### 4.1 Experimental Setup

**4.1.1 Datasets.** We choose different categories from **Amazon Review** dataset [33] for open-source experiments. In the pre-training and fine-tuning stage, we uniformly employ the leave-one-out strategy to divide datasets into training/validation/test splits.

**For pre-training,** eight domains with relatively rich interaction data are chosen: “Automotive”, “Home and Kitchen”, “Cell Phones and Accessories”, “Clothing, Shoes and Jewelry”, “Electronics”, “Grocery and Gourmet Food”, “Movies and TV” and “CDs and Vinyl”. We mix data from all these domains for joint pre-training.

**For fine-tuning,** we select six domains with sparser data including “Industrial and Scientific”, “Pet Supplies”, “Musical Instruments”, “Video Games”, “Arts, Crafts and Sewing”, “Office Products” as downstream domains to evaluate the final performance of ROMA.

We follow [27] to use the five-core datasets provided by Amazon and retrieve item text attributes including *title*, *categories* and *brand* from the metadata. We further crawl item images through URLs in the metadata as item image attributes. While text attributes are fully available, some images are missed because of expired URLs. We retain items without images in datasets for the fairness of comparison. The final dataset statistics are shown in Table 1. As we will show below, ROMA exhibits robust improvement in the face of modality absence, which is common in practical application.

**Table 1: Statistics of Pre-processed Datasets. “Cover.” denotes the image coverage among the item set. “Avg. SL” denotes the average length of interaction sequences.**

Datasets	#Users	#Items	#Img. (Cover./%)	#Inters.	Avg. SL.
<i>Pre-training</i>	3,608,532	1,022,309	724,562 (70.88%)	33,572,032	9.30
Scientific	11,041	5,327	3,490 (65.52%)	76,896	6.96
Instruments	27,530	10,611	6,289 (59.27%)	231,312	8.40
Pet	47,569	37,970	30,611 (80.62%)	420,662	8.84
Arts	56,210	22,855	13,418 (58.71%)	492,492	8.76
Games	55,223	17,389	14,967 (86.07%)	496,315	8.99
Office	101,501	27,932	20,542 (73.54%)	798,914	7.87

**4.1.2 Metrics.** We adopt three metrics  $NDCG@K$ ,  $Recall@K$  and  $MRR$ , which are widely used in SR to evaluate the recommendation performance of methods.  $K$  is set to 10 for showcases.

**4.1.3 Baselines.** For a comprehensive comparison, we have selected the following three groups of baselines.

(1) ID-based Methods

- **GRU4Rec** [18] is a session-based recommender adopting RNNs to model user behavior sequences.
- **SASRec** [23] firstly employs a self-attentive Transformer layer as the sequence encoder to capture item dependency.
- **BERT4Rec** [44] applies a bi-directional attention mechanism and item clozing task for sequence representation learning.

(2) Text-Enhanced Methods

- **UniSRec** [20] replaces ID embeddings with text representations and adapts models to target domains with MoE adapters.

- **Recformer** [27] trains Longformer [1] with MLM and sequence-item contrast task to encode item word tokens and conducts two-stage training (pre-train and fine-tune).
- **TedRec** [58] converts ID and text features into the frequency domain and completes feature fusion at the sequence level.

(3) Multi-Modal Methods

- **MISSRec** [50] first extracts text and image features of items and then builds an interest-aware encoder-decoder model to pre-train universal sequence representation for recommendation.
- **MMSASRec** is a modified version of SASRec. We add text and visual features projected by a linear layer to ID features to fuse multi-modal information into original SASRec.

**4.1.4 Implementation Details.** Due to limited computational resources, we choose tiny MiniLMv2 [53] as the base PLM, we adopt a multi-modal encoder universal for all vision-language tasks, BEiT3 [52], to encode item images. ROMA is also applicable to other base models (Section 5.4). We set the maximum number of tokens to 32 for a single attribute and 1,024 for items and user sequences. The max length of each user sequence is limited to 50 items for all methods. The temperature coefficient  $\tau$  is 0.05 and the loss weight  $\lambda$  is set to 0.1. The number of adapters in the bottom layers is set to 8 and all adapters’ inner dimension is set to 128. The batch size is 256 for pre-training and 32 for fine-tuning. We train ROMA for 20 epochs in pre-training stage and at most 15 epochs for fine-tuning.

We adopt the hyper-parameter settings reported in baselines’ papers to ensure performance. We implement MISSRec, UniSRec and all ID-based recommenders with open-source RECOLE framework [68]. For a fair comparison, we pre-train and fine-tune UniSRec, MISSRec and Recformer with the same datasets as ROMA.

### 4.2 Research Questions

We try to answer the following research questions of our framework with empirical evaluation result and further analysis:

- RQ1:** Can ROMA achieve competitive results with other state-of-the-art SR methods in downstream domains?
- RQ2:** Can the universal representations of ROMA benefit recommendation performance on cold-start items?
- RQ3:** Is the routing-based transfer strategy effective in modeling domain affinities and mitigating negative transfer?
- RQ4:** How do other designs impact the efficacy of ROMA?

## 5 EVALUATION RESULTS

### 5.1 Comparison with State-of-the-arts (RQ1)

We report overall experiment results on six downstream datasets in Table 2. From the result we conclude that (1) modality-enhanced approaches exhibit superior overall performance compared to ID-based methods, substantiating the advantages of content-based recommendation in terms of generality and interpretability. (2) Our proposed ROMA achieves the best performance on all datasets except the sub-optimal  $Recall@10$  on *Instruments*, which indicates the effectiveness of ROMA to adapt PLM for recommendation. In comparison to baselines, ROMA realizes an enhancement of approximately 10% across metrics in all domains, and in *Games* domain, the improvement exceeds 20%. (4) ROMA exhibits better

**Table 2: Performance comparison of ROMA and other baselines. “R@K” is short for “Recall@K” and “N@K” is short for “NDCG@K”. Optimal and sub-optimal performance is denoted in bold and underlined fonts, respectively.**

Model Type →		ID-based			Text-enhanced			Multi-modal			
Dataset	Metric	GRU4Rec	SASRec	BERT4Rec	UniSRec	Recformer	TedRec	MISSRec	MMSASRec	ROMA	(Imprv.)
Scientific	N@10	0.0414	0.0655	0.0336	0.0788	<u>0.1027</u>	0.0908	0.0793	0.0977	<b>0.1139</b>	10.91%
	R@10	0.0952	0.1206	0.0552	0.1376	<u>0.1448</u>	0.1256	0.1407	0.1373	<b>0.1619</b>	11.81%
	MRR	0.0641	0.0541	0.0317	0.0679	<u>0.0951</u>	0.0859	0.0675	0.0869	<b>0.1058</b>	11.25%
Games	N@10	0.0424	0.0442	0.0281	0.0532	0.0702	0.0631	0.0531	<u>0.0732</u>	<b>0.0891</b>	21.72%
	R@10	0.0816	0.0971	0.0552	0.1128	0.1092	0.1135	0.1142	<u>0.1143</u>	<b>0.1426</b>	24.76%
	MRR	0.0390	0.0374	0.0266	0.0454	0.0659	0.0575	0.0449	<u>0.0681</u>	<b>0.0823</b>	20.85%
Instruments	N@10	0.0648	0.0664	0.0574	0.0759	0.0841	0.0870	0.0765	<u>0.0842</u>	<b>0.0959</b>	13.90%
	R@10	0.0894	0.1171	0.0805	0.1290	0.1085	0.1204	<b>0.1324</b>	0.1126	<u>0.1295</u>	-
	MRR	0.0624	0.0570	0.0552	0.0659	0.0815	<u>0.0832</u>	0.0668	0.0809	<b>0.0919</b>	10.46%
Arts	N@10	0.0677	0.0744	0.0594	0.0851	<u>0.1220</u>	0.1065	0.0852	0.1161	<b>0.1343</b>	10.08%
	R@10	0.0952	0.1124	0.0840	0.1477	0.1645	0.1455	0.1506	<u>0.1649</u>	<b>0.1735</b>	5.22%
	MRR	0.0641	0.0678	0.0564	0.0726	0.1138	0.1008	0.0719	<u>0.1155</u>	<b>0.1283</b>	11.08%
Office	N@10	0.0775	0.0832	0.0671	0.0855	<u>0.1141</u>	0.1096	0.0890	0.1135	<b>0.1288</b>	12.88%
	R@10	0.1084	0.1215	0.0900	0.1358	0.1403	0.1418	0.1384	<u>0.1428</u>	<b>0.1596</b>	11.76%
	MRR	0.0887	0.0751	0.0631	0.0745	0.1089	0.1041	0.0783	<u>0.1105</u>	<b>0.1237</b>	11.95%
Pet	N@10	0.0853	0.0833	0.0596	0.0796	<u>0.0978</u>	0.0973	0.0841	0.0944	<b>0.1049</b>	7.26%
	R@10	0.1084	0.1173	0.0917	0.1238	0.1214	0.1235	<u>0.1249</u>	0.1188	<b>0.1307</b>	4.64%
	MRR	0.0833	0.0780	0.0537	0.0716	0.0935	<u>0.0936</u>	0.0772	0.0888	<b>0.1013</b>	8.23%

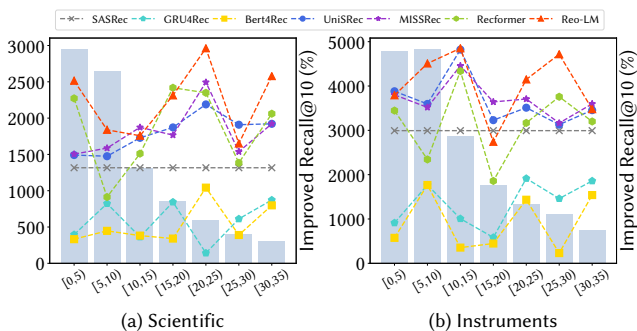
ranking ability: In most domains, relatively improvement on *NDCG* and *MRR* is higher than on *Recall*.

We attribute ROMA’s success to these factors: (1) The VA-MLM task enables ROMA to utilize multiple modalities for comprehensive user/item modeling. This explains its greater enhancements in domains like *Games* where visual features are pivotal. (2) Partition of model and tasks in ROMA endows a more effective injection of modeling and prediction abilities, which benefits the adaptation of PLM for recommendation and brings better ranking ability. (3) the MoA routing strategy grants ROMA with non-conflicted knowledge transfer from upstream to downstream domains, enabling the pre-trained model effectively to adapt to downstream domains.

## 5.2 Cold-start Analysis (RQ2)

We make a comparison about the performances of methods towards cold-start problem. We choose the smallest *Scientific* and *Instruments* domain and divide their test set into different groups by item frequency in the training set. The performances on these groups are shown in Figure 2.

It turns out that modality-enhanced methods outperform ID-based methods under most settings, and the relative improvement is more stable for colder items. This aligns with the observation that ID-based methods struggle to model less popular items. Moreover, ROMA achieves the best few-shot performance among all methods in most cases, which indicates the efficacy of multiple modalities for cold-start problem and the mastery of adapting PLM in item modeling over feature-based methods like MISSRec.



**Figure 2: Performance comparison w.r.t. cold-start items. The bar graph denotes the number of items with different interaction numbers. The line chart denotes relative improvement ratios compared with the baseline method SASRec.**

## 5.3 Analysis of Domain Affinity (RQ3)

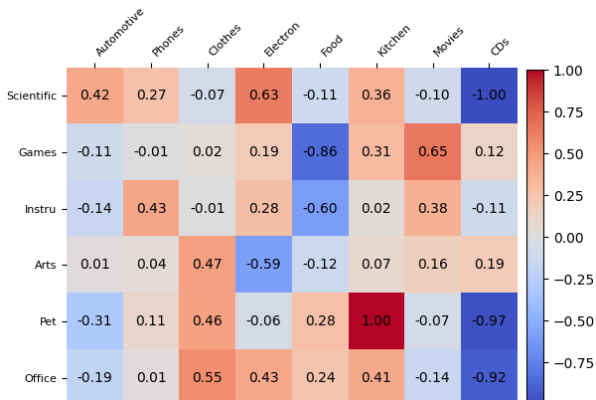
**5.3.1 Adapter weights can capture domain affinities.** To investigate the efficacy of our routing strategy, we analyze how fine-tuned MoA gates allocate weights to various upstream domains: For each downstream domain, we sample 1,000 sequences and gather weights assigned to adapters on all sequence tokens, then we average and normalize these weights by corresponding upstream domains.

We take normalized weights to indicate affinities between upstream and downstream domains and depict domain correlation in Figure 3. We observe that affinities learned by MoA gates are in a coherent alignment with the intuitive proximity between these domains. For example, for *Games* domain, the most-related upstream domain is *Movies*, which is reasonable since lots of games are adapted from films and television works. There also exists counter-intuitive cases like the negative affinity between *CDs* and

**Table 3: Ablation analysis on three downstream datasets. We choose domains with least, medium and most interactions for comprehensive ablation. The best and the second-best performance are denoted in bold and underlined fonts, respectively.**

Variants	Scientific			Games			Office		
	N@10	R@10	MRR	N@10	R@10	MRR	N@10	R@10	MRR
(0) ROMA	<b>0.1139</b>	<u>0.1619</u>	<b>0.1058</b>	<b>0.0891</b>	<u>0.1426</u>	<b>0.0823</b>	<b>0.1288</b>	<u>0.1596</u>	<b>0.1237</b>
(1) w/o VA-MLM	0.1099	0.1599	0.1024	0.0817	0.1324	0.0757	0.1269	0.1573	0.1220
(2) w/o user-item contrast	0.1080	0.1548	0.1003	0.0847	0.1363	0.0785	0.1120	0.1442	0.1063
(3) w/o partition	0.1086	0.1534	0.1015	0.0829	0.1343	0.0766	0.1253	0.1566	0.1201
(4) w/o top MoA	0.1102	0.1582	0.1023	0.0863	0.1400	0.0794	0.1278	0.1576	0.1229
(5) w/o image	0.1080	0.1584	0.0993	0.0857	0.1382	0.0786	0.1283	0.1581	0.1235
(6) w/o real-time negatives	0.1068	0.1501	0.0990	0.0840	0.1412	0.0763	0.1131	0.1373	0.1084
(7) w/ RoBERTa-Base	0.1113	<b>0.1649</b>	0.1030	0.0881	<b>0.1435</b>	0.0814	0.1223	<b>0.1643</b>	0.1171
(8) w/ CLIP-Large-P14	<u>0.1120</u>	0.1612	<u>0.1036</u>	<u>0.0884</u>	0.1416	<u>0.0817</u>	<u>0.1287</u>	0.1592	<u>0.1237</u>

*Instruments*, which may come from user group divergence. Overall, Figure 3 suggests that ROMA’s routing strategy can efficiently transfer knowledge from closely related upstream domains.

**Figure 3: We normalize the weights allocated to upstream domain adapters by fine-tuned downstream models to analysis the domain affinities learned by transfer strategy.**

#### 5.4 Ablation Study (RQ4)

We construct 8 variants to analyze each component’s contribution.

In variant (1) and (2), we separately remove the VA-MLM task and the UIC task in Section 3.3. Performance degradation from task absence confirms the validity and necessity of our adaption towards two-aspect divergences. Besides, variants (1) and (2) both outperform ROMA without pre-training in Table 4, further indicating the effect of single task for recommendation-oriented adaption.

Then we examine the effectiveness of our architecture designs. Variant (3) cancels the partition strategy and moves the VA-MLM task to the top of the model. The triggered performance decline verifies that our hierarchical adaption strategy improves model’s learning efficiency for different skills. Variant (4) removes MoA modules in top layers, it performs worse without measure for the

negative transfer. And MoAs have more assistance for sparser domains, which have stronger dependency on transferred knowledge.

Variant (5) removes image attributes to limit the input to be text-only. The decreased results show the value of multi-modality for RS. The small performance gap on the *Office* domain could be caused by high image similarity between *Office* items.

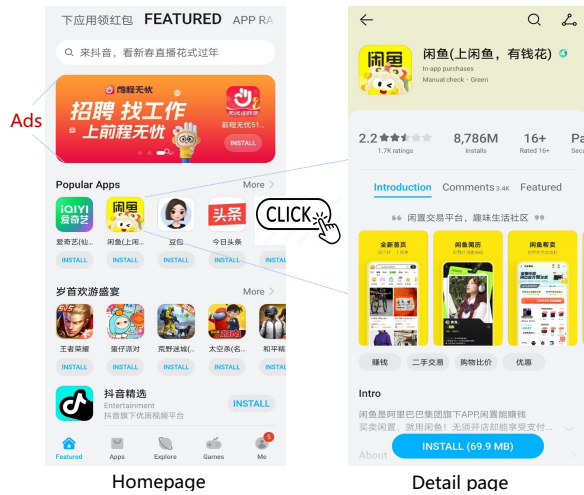
Variant (6) cancels the real-time negatives in fine-tuning, and only adopts the frozen embedding table as negatives, the performance decay indicates that random real-time negatives can reach a better balance between computation cost and performance, which is especially pivotal for domains with more items and interactions.

For variant (7) and (8), we replace the PLM and vision encoder with larger RoBERTa-Base and inferior CLIP-Large-P-32, respectively. With RoBERTa, ROMA receives the highest *Recall* rate, but exhibits worse on ranking metric *NDCG* and *MRR*, which may result from excessive concentration on text. And with CLIP, ROMA only receives modest metric decline. We demonstrate that our ROMA framework is robust for different PLMs and vision encoders.

#### 5.5 Industrial Deployment and Evaluation

In this section, we describe the deployment of our ROMA within a large-scale commercial recommender system at Huawei. Specifically, we showcase its application in advertisement recommendation on the homepage of the Huawei AppGallery. As illustrated in Figure 4, when a user opens the AppGallery homepage, they are presented with a horizontally scrolling banner of advertisements at the top. Additionally, clicking on any banner or app icon directs the user to a detailed page featuring app screenshots. Users can download and install advertised apps by clicking the ‘Install’ button. The recommender system currently serves a large base of smartphone users daily, with recommended advertisements and banner images updated frequently to ensure content freshness and relevance.

To evaluate the effectiveness of ROMA for industrial use, we deployed it in our App recommender system. We treated advertisements and native app interactions as two separate domains and collected both textual information and images as multi-modal data. The ROMA model was pre-trained using the BGE language



**Figure 4: An Illustration of Advertisement Recommendation at Huawei AppGallery.**

model [5], generating user and item embeddings with a dimensionality of 64. These embeddings were then used as pretrained features in the downstream ranking model (a DIN-like model [72]) for advertisement recommendation. We conducted online A/B testing using 10% of randomly sampled user traffic per group over a one-week period. During this time, we observed a 7.29% increase in average DTR (Download-Through Rate in terms of downloads) and a 3.17% increase in ECPM (Effective Cost Per Mille in terms of revenue) for advertisements, which are significant improvements in our scenario. Currently, ROMA has been fully deployed as a common multi-modal embedding extraction service, supporting the main traffic of Huawei’s AppGallery recommender system.

## 6 CONCLUSION

In this paper, we proposed a framework named ROMA to effectively adapt a PLM as a multi-modal sequential recommender. Based on our in-depth analysis of the divergences between PLMs and recommenders, we apply a partition strategy to divide the original model into the bottom and top layers and conquer disparities at different levels. To integrate the vision modality and adapt to recommendation context, we train bottom layers with vision-augmented mask language modeling task. To master behavior prediction and avoid domain conflicts, we apply MoA modules with delicate routing strategies in user-item contrastive task to train top layers. During pre-training and fine-tuning, rational negative sampling strategies ensure the balance between efficiency and efficacy.

With the above designs, ROMA achieves impressive improvements on open-source benchmarks. Further analysis reveals that ROMA’s pre-training always achieves positive transfer, and the partition strategy grants ROMA low saturation and better layer attention patterns. Besides, ROMA exhibits better performance for cold-start items than other methods. Finally, the ablation study shows the effectiveness of ROMA’s each component, it also demonstrates ROMA’s universality towards different foundation models.

In the appendix, we also compare the pre-training effects of ROMA with other methods, along with some case study experiments that explore the mechanism of ROMA.

## ACKNOWLEDGMENTS

We want to thank the anonymous reviewers and the meta-reviewer for their valuable comments and suggestions. This research is supported by National Natural Science Foundation of China (Grant No. 62276154, No.624B2088 and No.62171248), the Natural Science Foundation of Guangdong Province (Grant No. 2023A1515012914 and 440300241033100801770), Basic Research Fund of Shenzhen City (Grant No. JCYJ20210324120012033, JCYJ20240813112009013 and GJHZ20240218113603006), the Major Key Project of PCL (NO. PCL2024A08). We gratefully acknowledge the support of MindSpore<sup>1</sup>, which is a new deep learning framework.

## References

- [1] Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The Long-Document Transformer. *arXiv:2004.05150* (2020).
- [2] Desheng Cai, Shengsheng Qian, Quan Fang, and Changsheng Xu. 2021. Heterogeneous cross-hierarchical feature aggregation network for personalized micro-video recommendation. *IEEE TMM* (2021).
- [3] Jiangxia Cao, Xin Cong, Jiawei Sheng, Tingwen Liu, and Bin Wang. 2022. Contrastive Cross-Domain Sequential Recommendation. In *CIKM*.
- [4] Yukuo Cen, Jianwei Zhang, Xu Zou, Chang Zhou, Hongxia Yang, and Jie Tang. 2020. Controllable multi-interest framework for recommendation. In *KDD*.
- [5] Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. BGE M3-Embedding: Multi-Lingual, Multi-Functionality, Multi-Granularity Text Embeddings Through Self-Knowledge Distillation. *CoRR abs/2402.03216* (2024).
- [6] Yongjun Chen, Zhiwei Liu, Jia Li, Julian McAuley, and Caiming Xiong. 2022. Intent contrastive learning for sequential recommendation. In *WWW*.
- [7] Mingyue Cheng, Fajie Yuan, Qi Liu, Xin Xin, and Enhong Chen. 2021. Learning transferable user representations with sequential behaviors via contrastive pre-training. In *ICDM*.
- [8] Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. 2019. What does bert look at? an analysis of bert’s attention. In *ACL*.
- [9] Boqi Dai, Zhaocheng Du, Jieming Zhu, Jintao Xu, Deqing Zou, Quanyu Dai, Zhenhua Dong, Rui Zhang, and Hai-Tao Zheng. 2024. UniEmbedding: Learning Universal Multi-Modal Multi-Domain Item Embeddings via User-View Contrastive Learning. In *CIKM*.
- [10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL*.
- [11] 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 *MM*.
- [12] Xinyu Du, Huanhuan Yuan, Pengpeng Zhao, Jianfeng Qu, Fuzhen Zhuang, Guanfeng Liu, Yanchi Liu, and Victor S Sheng. 2023. Frequency enhanced hybrid attention network for sequential recommendation. In *SIGIR*.
- [13] Mor Geva, Avi Caciularu, Kevin Ro Wang, and Yoav Goldberg. 2022. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. In *EMNLP*.
- [14] Kai Han, Yunhe Wang, Hanting Chen, Xinghao Chen, Jianyuan Guo, Zhenhua Liu, Yehui Tang, An Xiao, Chunjing Xu, Yixing Xu, et al. 2022. A survey on vision transformer. *IEEE transactions on pattern analysis and machine intelligence* 45, 1 (2022), 87–110.
- [15] Xu Han, Zhengyan Zhang, Ning Ding, Yuxian Gu, Xiao Liu, Yuqi Huo, Jiezhong Qiu, Yuan Yao, Ao Zhang, Liang Zhang, et al. 2021. Pre-trained models: Past, present and future. *AI Open* 2 (2021), 225–250.
- [16] Ruining He and Julian McAuley. 2016. Fusing similarity models with markov chains for sparse sequential recommendation. In *ICDM*.
- [17] Ruining He and Julian McAuley. 2016. VBPR: visual bayesian personalized ranking from implicit feedback. In *AAAI*.
- [18] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based recommendations with recurrent neural networks. In *ICLR*.
- [19] Yupeng Hou, Zhankui He, Julian McAuley, and Wayne Xin Zhao. 2023. Learning Vector-Quantized Item Representation for Transferable Sequential Recommenders. In *WWW*.
- [20] 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 *SIGKDD*.
- [21] Hengchang Hu, Wei Guo, Xu Liu, Yong Liu, Ruiming Tang, Rui Zhang, and Min-Yen Kan. 2024. User Behavior Enriched Temporal Knowledge Graphs for

<sup>1</sup><https://www.mindspore.cn>

- Sequential Recommendation. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*. 266–275.
- [22] Ganesh Jawahar, Benoît Sagot, and Djamel Seddah. 2019. What does BERT learn about the structure of language?. In *ACL*.
- [23] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *ICDM*.
- [24] Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. 2022. Transformers in vision: A survey. *ACM computing surveys (CSUR)* 54, 10s (2022), 1–41.
- [25] Chenyi Lei, Yong Liu, Lingzi Zhang, Guoxin Wang, Haihong Tang, Houqiang Li, and Chunyan Miao. 2021. Semi: A sequential multi-modal information transfer network for e-commerce micro-video recommendations. In *KDD*. 3161–3171.
- [26] Dmitry Lepikhin, Hyoukjoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. 2021. Gshard: Scaling giant models with conditional computation and automatic sharding. In *ICLR*.
- [27] Jiacheng Li, Ming Wang, Jin Li, Jinmiao Fu, Xin Shen, Jingbo Shang, and Julian McAuley. 2022. Text Is All You Need: Learning Language Representations for Sequential Recommendation. In *SIGKDD*.
- [28] Jiahao Liang, Xiangyu Zhao, Muyang Li, Zijian Zhang, Wanyu Wang, Haochen Liu, and Zitao Liu. 2023. MMMLP: Multi-modal Multilayer Perceptron for Sequential Recommendations. In *WWW*.
- [29] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *Comput. Surveys* 55, 9 (2023), 1–35.
- [30] Qijiong Liu, Jieming Zhu, Yanting Yang, Quanyu Dai, Zhaocheng Du, Xiao-Ming Wu, Zhou Zhao, Rui Zhang, and Zhenhua Dong. 2024. Multimodal Pretraining, Adaptation, and Generation for Recommendation: A Survey. In *KDD*.
- [31] Qijiong Liu, Jieming Zhu, Yanting Yang, Quanyu Dai, Zhaocheng Du, Xiao-Ming Wu, Zhou Zhao, Rui Zhang, and Zhenhua Dong. 2024. Multimodal pretraining, adaptation, and generation for recommendation: A survey. In *SIGKDD*.
- [32] Zhenghao Liu, Sen Mei, Chenyan Xiong, Xiaohua Li, Shi Yu, Zhiyuan Liu, Yu Gu, and Ge Yu. 2023. Text Matching Improves Sequential Recommendation by Reducing Popularity Biases. In *CIKM*.
- [33] Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *EMNLP*.
- [34] Yabo Ni, Dan Ou, Shichen Liu, Xiang Li, Wenwu Ou, Anxiang Zeng, and Luo Si. 2018. Perceive your users in depth: Learning universal user representations from multiple e-commerce tasks. In *KDD*.
- [35] 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 *CIKM*.
- [36] Zexuan Qiu, Jieming Zhu, Yankai Chen, Guohao Cai, Weiwen Liu, Zhenhua Dong, and Irwin King. 2024. EASE: Learning Lightweight Semantic Feature Adapters from Large Language Models for CTR Prediction. In *CIKM*.
- [37] Massimo Quadrona, Paolo Cremonesi, and Dietmar Jannach. 2018. Sequence-aware recommender systems. *ACM CSUR* (2018).
- [38] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training. (2018).
- [39] Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Hulikal Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Tran, Jonah Samost, et al. 2024. Recommender systems with generative retrieval. In *Neurips*.
- [40] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized markov chains for next-basket recommendation. In *WWW*.
- [41] Andrew I Schein, Alexandrin Popescul, Lyle H Ungar, and David M Pennock. 2002. Methods and metrics for cold-start recommendations. In *SIGIR*.
- [42] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *ICLR*.
- [43] Xiaoteng Shen, Rui Zhang, Xiaoyan Zhao, Jieming Zhu, and Xi Xiao. 2024. Pmg: Personalized multimodal generation with large language models. In *WWW*.
- [44] 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 *CIKM*.
- [45] Jiayi Tang and Ke Wang. 2018. Personalized top-n sequential recommendation via convolutional sequence embedding. In *WSDM*.
- [46] Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovered the classical NLP pipeline. In *ACL*.
- [47] Changxin Tian, Zihan Lin, Shuqing Bian, Jinpeng Wang, and Wayne Xin Zhao. 2022. Temporal Contrastive Pre-Training for Sequential Recommendation. In *CIKM*.
- [48] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*.
- [49] 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).
- [50] Jinpeng Wang, Ziyun Zeng, Yunxiao Wang, Yuting Wang, Xingyu Lu, Tianxiang Li, Jun Yuan, Rui Zhang, Hai-Tao Zheng, and Shu-Tao Xia. 2023. MISSRec: Pre-training and Transferring Multi-modal Interest-aware Sequence Representation for Recommendation. In *MM*.
- [51] Shoujin Wang, Liang Hu, Yan Wang, Longbing Cao, Quan Z. Sheng, and Mehmet Orgun. 2019. Sequential Recommender Systems: Challenges, Progress and Prospects. In *IJCAI*.
- [52] Wenhui Wang, Hangbo Bao, Li Dong, Johan Björck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, Subhojit Som, et al. 2022. Image as a foreign language: Beit pretraining for all vision and vision-language tasks. *arXiv preprint arXiv:2208.10442* (2022).
- [53] Wenhui Wang, Hangbo Bao, Shaohan Huang, Li Dong, and Furu Wei. 2021. MiniLMv2: Multi-Head Self-Attention Relation Distillation for Compressing Pretrained Transformers. In *Findings of ACL-IJCNLP*.
- [54] Yichao Wang, Hui Feng Guo, Bo Chen, Weiwen Liu, Zhirong Liu, Qi Zhang, Zhicheng He, Hongkun Zheng, Weiwei Yao, Muyu Zhang, et al. 2022. Causalint: Causal inspired intervention for multi-scenario recommendation. In *SIGKDD*.
- [55] Chuhan Wu, Fangzhao Wu, Tao Qi, Chao Zhang, Yongfeng Huang, and Tong Xu. 2022. MM-Rec: Visiolinguistic Model Empowered Multimodal News Recommendation. In *SIGIR*.
- [56] Chaojun Xiao, Ruobing Xie, Yuan Yao, Zhiyuan Liu, Maosong Sun, Xu Zhang, and Leyu Lin. 2021. UPRec: User-aware Pre-training for Recommender Systems. *arXiv preprint arXiv:2102.10989* (2021).
- [57] Xu Xie, Fei Sun, Zhaoyang Liu, Shiwen Wu, Jinyang Gao, Jiandong Zhang, Bolin Ding, and Bin Cui. 2022. Contrastive learning for sequential recommendation. In *ICDE*.
- [58] Lanling Xu, Zhen Tian, Bingqian Li, Junjie Zhang, Daoyuan Wang, Hongyu Wang, Jinpeng Wang, Sheng Chen, and Wayne Xin Zhao. 2024. Sequence-level Semantic Representation Fusion for Recommender Systems. In *CIKM*. 5015–5022.
- [59] Jiahao Xun, Shengyu Zhang, Zhou Zhao, Jieming Zhu, Qi Zhang, Jingjie Li, Xiuqiang He, Xiaofei He, Tat-Seng Chua, and Fei Wu. 2021. Why do we click: visual impression-aware news recommendation. In *MM*.
- [60] Fajie Yuan, Xiangnan He, Alexandros Karatzoglou, and Liguang Zhang. 2020. Parameter-efficient transfer from sequential behaviors for user modeling and recommendation. In *SIGIR*.
- [61] Fajie Yuan, Alexandros Karatzoglou, Ioannis Arapakis, Joemon M Jose, and Xiangnan He. 2019. A simple convolutional generative network for next item recommendation. In *WSDM*.
- [62] 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. In *SIGIR*.
- [63] Jiaqi Zhai, Lucy Liao, Xing Liu, Yueming Wang, Rui Li, Xuan Cao, Leon Gao, Zhaojie Gong, Fangda Gu, Michael He, et al. 2024. Actions Speak Louder than Words: Trillion-Parameter Sequential Transducers for Generative Recommendations. In *ICML*.
- [64] Lingzi Zhang, Xin Zhou, Zhiwei Zeng, and Zhiqi Shen. 2024. Are id embeddings necessary? whitening pre-trained text embeddings for effective sequential recommendation. In *ICDE*.
- [65] Qi Zhang, Jingjie Li, Qinglin Jia, Chuyuan Wang, Jieming Zhu, Zhaowei Wang, and Xiuqiang He. 2021. UNBERT: User-News Matching BERT for News Recommendation. In *IJCAI*.
- [66] Shengyu Zhang, Lingxiao Yang, Dong Yao, Yujie Lu, Fuli Feng, Zhou Zhao, Tat-Seng Chua, and Fei Wu. 2022. Re4: Learning to Re-contrast, Re-attend, Re-construct for Multi-interest Recommendation. In *WWW*.
- [67] Yang Zhang, Fuli Feng, Xiangnan He, Tianxin Wei, Chonggang Song, Guohui Ling, and Yongdong Zhang. 2021. Causal intervention for leveraging popularity bias in recommendation. In *SIGIR*.
- [68] Wayne Xin Zhao, Yupeng Hou, Xingyu Pan, Chen Yang, Zeyu Zhang, Zihan Lin, Jingsen Zhang, Shuqing Bian, Jiakai Tang, Wenqi Sun, et al. 2022. RecBole 2.0: Towards a More Up-to-Date Recommendation Library. In *CIKM*.
- [69] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223* (2023).
- [70] Bowen Zheng, Yupeng Hou, Hongyu Lu, Yu Chen, Wayne Xin Zhao, and Ji-Rong Wen. 2024. Adapting large language models by integrating collaborative semantics for recommendation. In *ICDE*.
- [71] Xiaolin Zheng, Jiajie Su, Weiming Liu, and Chaochao Chen. 2022. DDGHM: dual dynamic graph with hybrid metric training for cross-domain sequential recommendation. In *Multimedia*.
- [72] 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 *KDD*.
- [73] 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 *CIKM*.
- [74] Kun Zhou, Hui Yu, Wayne Xin Zhao, and Ji-Rong Wen. 2022. Filter-enhanced MLP is all you need for sequential recommendation. In *WWW*.

## A APPENDIX

### A.1 Visualization of Corpora Divergence

To more intuitively display the divergence in corpora between recommendation and general domains, we provide a visual exhibition about the proportion of words with various parts of speech from two representative datasets, Amazon Review and pre-training corpus of RoBERTa, in Figure 5.

It is evident that in the context of product descriptions, the recommendation corpus exhibits a markedly higher prevalence of nouns, numerals, and adjectives compared to the general domain. Conversely, the occurrence of verbs, adverbs, and other parts of speech is comparatively lower. This observation superficially underscores the distinct disparities between the recommendation text and general text in aspects such as textual composition, sentence construction, and semantic conveyance. Consequently, it is imperative to undertake semantic-level adaptations to enhance the modeling accuracy of products within recommendation system.

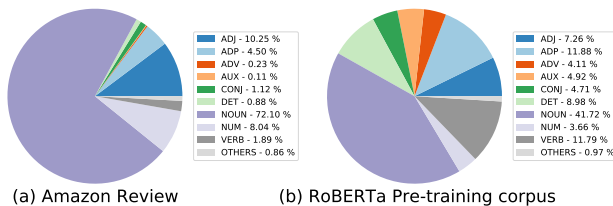


Figure 5: Proportion about words with different POS from Amazon Review dataset and pre-training corpus of RoBERTa.

### A.2 Effect of Pre-training (RQ2)

To examine the efficacy of ROMA’s pre-training, we make comparisons about the performance improvement brought by pre-training among several typical methods. The results are shown in Table 4.

Table 4: Analysis of pre-training effect for three types of methods. The green block means the corresponding setting performs better, and grey means worse.

Domain	SASRec w/o Pre-training			SASRec w/ Pre-training		
	R@10	N@10	MRR	R@10	N@10	MRR
Scientific	0.1206	0.0655	0.0541	0.1143	0.0717	0.0652
Games	0.0971	0.0442	0.0374	0.0918	0.0476	0.0434
Office	0.1215	0.0832	0.0751	0.1180	0.0854	0.0796

Domain	MISSRec w/o Pre-training			MISSRec w/ Pre-training		
	R@10	N@10	MRR	R@10	N@10	MRR
Scientific	0.1347	0.0731	0.0610	0.1407	0.0793	0.0675
Games	0.1180	0.0548	0.0460	0.1142	0.0531	0.0449
Office	0.1310	0.0870	0.0778	0.1384	0.0890	0.0783

Domain	ROMA w/o Pre-training			ROMA w/ Pre-training		
	R@10	N@10	MRR	R@10	N@10	MRR
Scientific	0.1380	0.0974	0.0912	0.1619	0.1139	0.1058
Games	0.1352	0.0847	0.0782	0.1426	0.0891	0.0823
Office	0.1529	0.1248	0.1205	0.1596	0.1288	0.1237

We can conclude that: (1) As an ID-based method, the enhancement of SASRec’s pre-training is subtle and variable across metrics. (2) As a modality-enhanced method, although MISSRec’s pre-training achieves relatively substantial improvements, its performance is not consistent and causes negative transfer to the *Game* domain. (3) In contrast, ROMA surpasses in both the magnitude and universality of improvement. This indicates that the routing strategy of our MoA modules in the top layers achieve an anticipated effect to mitigate conflicted knowledge from upstream domains.

Besides, under no pre-training setting, ROMA also generally performs better, demonstrating the advantage of content-based RS and the exceptional comprehension capability of PLM.

### A.3 Case study for cross-domain knowledge transfer

Compared to ID-based methods, modality-based approaches are more abstract and complex in cross-domain transfer. We conduct a case study to illustrate the knowledge our method transfers from pre-training to assist recommendation in downstream domains.

We select the *Games* domain, which shows the greatest relative improvement among downstream domains, and the *Movies* domain, which is allocated with the highest weight among domain adapters from *Games*, as subjects for analysis. The two sequences in Figure 6 from different users demonstrate how ROMA utilizes modal information for knowledge transfer: During pre-training, sequences in the *Movies* domain that contain both *Dragon Ball* and *Gundam* related products inspire ROMA with the association between these two topics. During fine-tuning in the *Games* domain, this association is transferred and enables ROMA to recommend products of one topic for users who are interested in the other.

Based on our statistics, there are 53 examples containing both above topics in the *Movies* domain and 12 in the *Games* domain. There might be thousands of similar cases between different domain pairs, here, we just select one intuitive example as illustration. A deeper investigation of the transfer mechanisms is left for future.

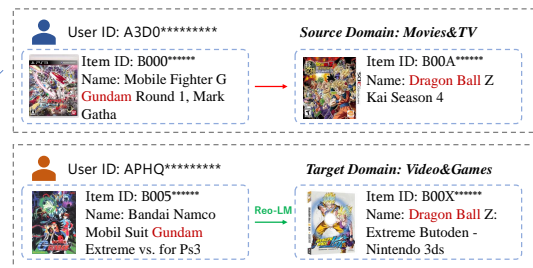
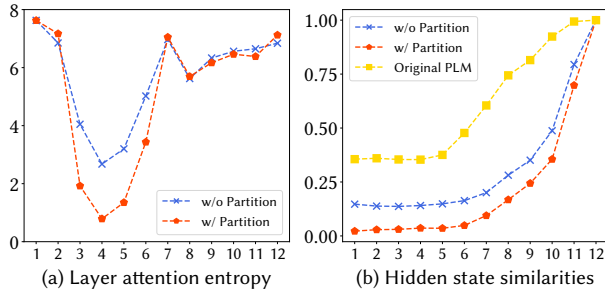


Figure 6: A case study for the knowledge transfer: The correlation between items of similar modality features is transferred from Movies domain to the Games domain.

### A.4 Influence of Partition Strategy

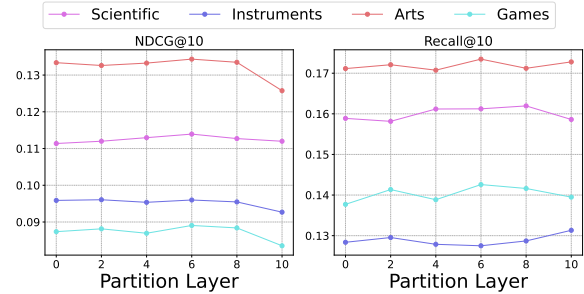
**A.4.1 Influence on model behavior.** We exhibit the knowledge injection effects of ROMA’s partition strategy through the attention pattern and saturation status. We sample 1,000 sequences to plot [CLS] attention entropy [8] of various models by layers in Figure 7 (a). Higher entropy signifies more widespread attention. Compared

to no partition, the attention entropy of ROMA’s medium layers is much lower while the top is higher. This confirms that the top layers are driven to focus on global information by the behavior prediction task. And bottom layers attend to features within local item, there is less information conduction among items, which causes lower entropy. In Figure 7(b), we depict cosine similarities between hidden states of the last layer and former layers. We observe the saturation status [13] in the original PLM. And lower similarities indicate ROMA inject behavior prediction ability into saturated high layers more efficiently than ROMA without partition.



**Figure 7: Influence of the partition strategy to model’s layer attention entropy and similarity saturation status.**

*A.4.2 Influence on model stability.* In our principal experiment, we intuitively partition the model into two parts with equal layers. We investigate the influence of the partition location on ROMA’s performance. Without loss of generality, we test ROMA’s performance when the model is divided at all even layers. We plot how *NDCG* and *Recall* metrics vary with the change of partition location in Figure 8. It turns out that ROMA’s performance is minimally affected by the partition location; we do not find our partition strategy to perform particularly well at any specific layer, but dividing at the middle layers tends to yield better results.



**Figure 8: Influence of the partition location to the stability of ROMA’s performance.**