Dynamic Multimodal Fusion via Meta-Learning Towards Micro-Video Recommendation

HAN LIU, Shandong University, China
YINWEI WEI, FAN LIU, and WENJIE WANG, National University of Singapore, Singapore
LIQIANG NIE, Harbin Institute of Technology (Shenzhen), China
TAT-SENG CHUA, National University of Singapore, Singapore

Multimodal information (e.g., visual, acoustic, and textual) has been widely used to enhance representation learning for micro-video recommendation. For integrating multimodal information into a joint representation of micro-video, multimodal fusion plays a vital role in the existing micro-video recommendation approaches. However, the static multimodal fusion used in previous studies is insufficient to model the various relationships among multimodal information of different micro-videos. In this article, we develop a novel meta-learning-based multimodal fusion framework called *Meta Multimodal Fusion* (MetaMMF), which dynamically assigns parameters to the multimodal fusion function for each micro-video during its representation learning. Specifically, MetaMMF regards the multimodal fusion of each micro-video as an independent task. Based on the meta information extracted from the multimodal features of the input task, MetaMMF parameterizes a neural network as the item-specific fusion function via a meta learner. We perform extensive experiments on three benchmark datasets, demonstrating the significant improvements over several state-of-the-art multimodal recommendation models, like MMGCN, LATTICE, and InvRL. Furthermore, we lighten our model by adopting canonical polyadic decomposition to improve the training efficiency, and validate its effectiveness through experimental results. Codes are available at https://github.com/hanliu95/MetaMMF.

CCS Concepts: • **Information systems → Recommender systems**; **Multimedia and multimodal retrieval**; • **Computing methodologies → Neural networks**;

Additional Key Words and Phrases: Dynamic multimodal fusion, meta-learning, micro-video recommendation, representation learning

1 INTRODUCTION

Micro-videos have become the emerging mediums of recording and sharing people's lives, due to their rich expression capacity via multimodal information, involving visual, acoustic, and textual modalities. Nowadays, popular micro-video platforms, such as Tiktok and Kwai, are exposed to an influx of users and micro-videos, attracting increasing research attention on micro-video recommendation [3, 20, 30].

Multimodal information (e.g., frames, audios, and captions) associated with items[1] has been widely used for representation learning in micro-video recommendation because of its capability to alleviate the user-item interaction sparsity problem [4, 7]. In other words, recommender systems can more comprehensively understand user preferences and item characters by incorporating multimodal information for representation learning. Existing multimodal recommendation models integrate multimodal information into a joint multimodal representation by employing various multimodal fusion methods. We roughly classify existing multimodal fusion into two categories: linear [43] and nonlinear [7] methods. For example, MMGCN [44] performs the graph convolutional operations on user-item graphs in different modalities, and implements multimodal fusion via the linear combination of multimodal representations. Due to the limitations of linear methods, deep neural networks are employed to fuse multimodal information nonlinearly. For instance, VBPR [15] adopts a neural layer to project the multi-modality features into the latent space. ACF [4] introduces a hierarchical attention mechanism to select informative content by assigning weights to different modalities.

Despite the remarkable performance, the previous methods are only able to perform static multimodal fusion as shown in Figure 1(a), following the assumptions that: (1) for different items, the information from the same modality plays a similar role and the multimodal information obeys the same relationship; and (2) a fusion function can be learned from the vast amount of items, and that the function learned is capable of modeling the relationships among the multimodal information. However, we argue that the above-mentioned assumptions are fragile in micro-video recommendation. In particular, unlike the well-designed video (e.g., movies or documentaries), the modality-specific information of different micro-videos may have very different effects on the expression of the theme, and the relationships among modalities might also be distinct [2]. For example, the information in visual and acoustic modalities is complementary in music micro-videos; while in education micro-videos, acoustic and textual modalities convey the themes in a consistent relationship. With this consideration, we argue that static fusion is under-fitting the varying multimodal fusion for learning representations of micro-videos, leading to suboptimal recommendation performance. Furthermore, we consider that each micro-video should combine its visual, acoustic, and textual modality information in a unique way.

In this article, we propose a dynamic multimodal fusion method that treats the multimodal fusion of each item as an independent task. As shown in Figure 1(b), the item-specific fusion functions are learned for different items independently. For implementation, we resort to the meta-learning techniques to establish the dynamic multimodal fusion as shown in Figure 1(c), due to their following advantages:

- Meta-learning algorithm is capable of generating adaptive parameters according to the context of a single task [10, 47], which is consistent with the target of dynamic multimodal fusion strategy.
- Meta-learning algorithm offers an effective way to perform transfer learning across tasks by means of shared parameters [38], thus enabling us to effectively learn the item-specific fusion functions with the limited training samples.

---

[1]Without special instructions, items refer to micro-videos indiscriminately in this article.
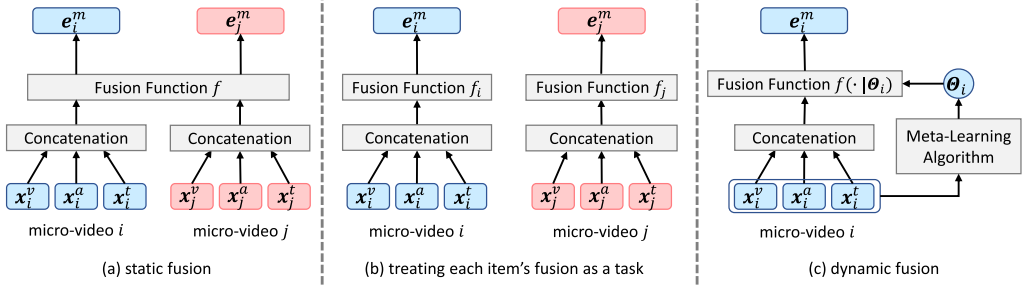
Fig. 1. Illustration of the static and the dynamic multimodal fusion. $\mathbf{x}_i^v$, $\mathbf{x}_i^a$, and $\mathbf{x}_i^t$ denote the features derived from visual, acoustic, and textual modalities of micro-video i, respectively. $\mathbf{e}_i^m$ denotes the multimodal representation of micro-video i through the fusion function. Θ_i denotes the item-specific parameters dynamically generated for the fusion function by a meta-learning algorithm. Similar notations are used for denoting the attributes of micro-video j.

More specifically, we develop a novel meta-learning-based multimodal fusion model, called **Meta Multimodal Fusion (MetaMMF)**, to dynamically integrate multimodal information for micro-video recommendation. In particular, MetaMMF consists of two main components: a meta information extractor and a meta fusion learner. The former is implemented by a **multi-layer perceptron (MLP)**. It projects the multimodal features of the input task (for an item) into the meta information that represents the task features in the form of higher-order abstraction [32]. Based on the meta information, the meta fusion learner parameterizes a neural network as the fusion function for the input task. Namely, the meta fusion learner establishes a mapping from meta information to fusion parameters, while generalizing the learning-to-learn (meta-learning) mechanism to tasks. To parameterize a layer in the neural network, the meta fusion learner exploits a task-shared 3-D tensor to transform the meta information vector into the layer's weight matrix by multiplication. Mathematically, by regarding the 3-D tensor as a set of 2-D matrices, the transformation is actually a linear combination of these 2-D matrices weighted by the elements in meta information, thus satisfying the dynamic nature of the results. Moreover, considering the model complexity problem caused by excessive parameters, we adopt *Canonical Polyadic* (**CP**) Decomposition to lighten our model by decomposing the 3-D tensors into fewer parameters. In this way, MetaMMF can dynamically learn an item-specific fusion function for each item to integrate its multimodal features into the multimodal representation. After combining with the collaborative representation, the joint representation can then be directly used to predict the interactions between the users and micro-videos like MF or passed through an advanced model like GCN. We conduct extensive experiments on three publicly accessible datasets to verify the rationality and effectiveness of our MetaMMF method.

Overall, the three main contributions of our work are summarized as follows:

- We highlight the importance of dynamic multimodal fusion for effective representation learning in micro-video recommendation. Towards this end, we present a meta-learning-based method to fuse multimodal information of each item dynamically. This work represents one of the pioneering efforts in utilizing meta-learning for dynamic multimodal fusion.
- We devise a model MetaMMF for dynamic multimodal fusion. By analyzing the to-be-fused multimodal features of an item, MetaMMF dynamically parameterizes a neural network utilized as the item-specific fusion function. As a technical contribution, we innovatively employ tensor decomposition to downsize the model parameters, resulting in a lighter model with improved training efficiency.

- To validate the effectiveness of our proposed method, we conducted extensive experiments on three real-world datasets. The experimental results demonstrate that MetaMMF achieves state-of-the-art performance in enhancing multimodal representation learning.

2 RELATED WORK

In this section, we review existing work on micro-video recommendation, multimodal fusion, and meta-learning, which are the most relevant to this work.

2.1 Micro-Video Recommendation

Consistent with the general personalized recommendation, the studies in this field parameterize users and micro-videos with embeddings, and reconstruct their interactions to learn parameters. Recently, multiple modalities of micro-videos are widely exploited to enhance their representations in related literature because they can provide rich auxiliary information [25].

One of the early studies, **Visual Bayesian Personalized Ranking (VBPR)** [15] model, projects the modal features of each micro-video into a modal representation, then concatenated with the collaborative representation to form the final item representation. Such a fusion can only capture a linear relationship. With the rise of deep learning, many studies have successfully employed the non-linear fusion of multimodal features. For example, **User-Video Co-Attention Network (UVCAN)** [29] utilizes attention mechanisms at the user and micro-video levels to selectively incorporate modality information for representation learning. Nevertheless, its fusion method, a weighted sum of multimodal features, results in coarse-grained multimodal relationships. More recently, **Graph Convolution Network (GCN)** [13] has gained popularity for recommendation representation learning due to its outstanding performance. MMGCN [44] builds modality-aware GCNs based on a user-item bipartite graph to learn unimodal representations for users and micro-videos. However, its final multimodal fusion relies on a linear combination, neglecting the varying impact of different modalities. **Heterogeneous Hierarchical Feature Aggregation Network (HHFAN)** [3] leverages a modality-aware heterogeneous information graph to explore complex relationships among users, micro-videos, and multimodal information, generating high-quality representations. Nevertheless, the fusion of the neighbor's multimodal features in HHFAN is performed using a static aggregation network. **Invariant Representation Learning (InvRL)** [8] learns invariant representations that account for user attention and mitigate the influence of spurious correlations in micro-video recommendations. The learned invariant representations can consistently predict user-item interactions across different environments. Despite this innovation, its fusion module simply concatenates multimodal features.

Although these previous studies have shown performance improvements, their static fusion approach limits the capture of diverse multimodal relationships in micro-videos, resulting in suboptimal representations and negatively affecting recommendation performance. To address this limitation, we propose a novel dynamic multimodal fusion-based recommendation model, enhancing the representation learning of micro-video by dynamically fusing multimodal information. Our model utilizes the multimodal features of micro-videos and leverages meta-learning to dynamically parameterize a fusion neural network for each micro-video. This adaptive approach enables the capture of specific multimodal relationships pertaining to each micro-video. Furthermore, our model is designed to be model-agnostic and lightweight, providing distinct advantages in flexibility and efficiency compared to previous methods.

2.2 Multimodal Fusion

Multimodal fusion is one of the traditional topics in multimodal machine learning. In technical terms, multimodal fusion refers to integrating information from multiple modalities to predict a

class through classification or a continuous value through regression [2]. Most studies on multimodal fusion are model-agnostic, which can be divided into early (i.e., feature-based), late (i.e., decision-based), and hybrid fusion.

Early fusion integrates features extracted from different modalities. For example, ACNet [18] utilizes the attention mechanism to integrate valuable features from RGB and depth branches adaptively. However, the fusion process only considers a complementary relationship and utilizes a static layer with element-wise addition. In contrast, late fusion combines the output results of multiple models for multiple modalities. The related studies focus on fusing unimodal decision values, such as the **neural multimodal cooperative learning (NMCL)** [42] model, which enhances individual features in each modality using cooperative nets and performs late fusion on the prediction results from different modalities. However, these approaches overlook the feature-level interactions between modalities. Hybrid fusion, on the other hand, combines outputs from early fusion and individual unimodal predictors, aiming to leverage the advantages of both above-described methods in a common framework, such as Combine Early and Late Fusion Together (CELFT) [41]. This universal hybrid framework considers the fusion of different levels regardless of the fusion dynamics.

These fusion approaches adhere to an immutable formulation, i.e., the fusion process remains constant regardless of the input. Such fusion strategies are unaware of the multimodal content of items. Considering that the relationships among modalities vary with the multimodal features of each item, we introduce a meta-learning-based multimodal fusion method. Our method dynamically generates item-specific fusion parameters by analyzing the input multimodal features, enabling us to achieve a superior fusion capability.

2.3 Meta-Learning in Recommendation

Meta-learning, also known as learning-to-learn, is inspired by the human learning process, which can quickly learn new tasks based on a small number of examples. Meta-learning tends to train a model that can adapt to a new task that is not used during the training with a few examples [6, 23]. The related studies can be classified into three types: metric-based [36], memory-based [35], and optimization-based meta-learning [9].

Metric-based methods learn a metric or distance function over tasks, while model-based methods aim to design an architecture or training process for rapid generalization across tasks. Lastly, optimization-based methods directly adjust the optimization algorithm to enable quick adaptation with just a few examples. Previous studies [35, 36] on metric-based and memory-based meta-learning convert the recommendation task to the classification problem. The work in [38] implements a meta-learning strategy by learning a neural network classifier whose biases are determined by the item history. The classifier performs recommendations by predicting whether a user consumes an item. A series of recent studies [47] have adopted the optimization-based meta-learning approach and chose **model-agnostic meta-learning (MAML)** [9] for model training. For example, MeLU [23] applies the MAML framework to score the affinities, which rapidly adapts to new users or items based on sparse interaction history.

From the perspective of meta-learning theory, learning the multimodal fusion process of each micro-video can be regarded as an individual task with limited training samples. Inspired by this, we propose a meta-learning strategy to implement dynamic multimodal fusion for learning robust representation of each micro-video in the recommendation scenario.

3 PRELIMINARIES

In this section, we first introduce the task of micro-video recommendation. We then shortly recapitulate the widely used multimodal representation learning based on neural networks, highlighting

the limitation caused by using static multimodal fusion. Finally, we formalize the problem of dynamic multimodal fusion. For ease of reading, we use a bold uppercase letter to denote a matrix, a bold lowercase letter to denote a vector, an italic letter to denote a scalar, and a calligraphic uppercase letter to denote a set.

3.1 Micro-Video Recommendation

Micro-video recommendation focuses on inferring the preference degree of user u on micro-video item i. Basically, all the models can be composed of two parts: representation learning and interaction prediction [27]. The former is responsible for yielding vector representations $\mathbf{e}_u$ and $\mathbf{e}_i$ for user u and item i, respectively. Then, the preference score of user u on item i can be estimated with the function,

$$\hat{y}_{ui} = \rho(\mathbf{e}_u, \mathbf{e}_i), \tag{1}$$

where $\rho(\cdot)$ denotes the prediction function (e.g., inner product). Since each item is associated with multimodal (e.g., visual, acoustic, and textual) features, many efforts have been made to exploit them to enhance the item representation. In recent studies, a typical operation is to represent item i by concatenating its collaborative representation and multimodal representation,

$$\mathbf{e}_i = \begin{bmatrix} \mathbf{e}_i^m \\ \mathbf{e}_i^c \end{bmatrix}, \tag{2}$$

where $\mathbf{e}_i^c \in \mathbb{R}^{d_c}$ indicates the collaborative representation bound up with ID, and $\mathbf{e}_i^m \in \mathbb{R}^{d_m}$ indicates the multimodal representation learned by integrating features from multiple modalities. d_c and d_m denote the dimensions of the two kinds of representation, respectively.

3.2 Multimodal Fusion for Multimodal Representation Learning

In order to obtain multimodal representation, a procedure is required to integrate multimodal features into the same embedding space [24]. Multimodal fusion emerges to resolve how to combine the information from heterogeneous sources and represent information in a format that a computation model can work with. Therefore, it is widely applied to learning joint multimodal representations in micro-video recommendation. For item i, its multimodal representation $\mathbf{e}_i^m$ can be mathematically expressed as,

$$\mathbf{e}_i^m = f\left(\mathbf{x}_i^v, \mathbf{x}_i^a, \mathbf{x}_i^t\right), \tag{3}$$

where $\mathbf{x}_i^v \in \mathbb{R}^{d_v}$, $\mathbf{x}_i^a \in \mathbb{R}^{d_a}$, and $\mathbf{x}_i^t \in \mathbb{R}^{d_t}$ denote the features deriving from visual, acoustic, and textual modalities of item i, respectively. d_v, d_a, and d_t denote the dimensions of these modality features, respectively. $f(\cdot)$ denotes the multimodal fusion function. Considering the ability to learn complex decision boundaries, recent studies have widely adopted neural networks as the multimodal fusion function [31, 33]. For example, VBPR uses one neural layer to implement multimodal fusion by projecting the concatenation of the visual, acoustic, and textual modality features,

$$\mathbf{e}_i^m = \mathbf{W} \begin{bmatrix} \mathbf{x}_i^v \\ \mathbf{x}_i^a \\ \mathbf{x}_i^t \end{bmatrix}, \tag{4}$$

where $\mathbf{W} \in \mathbb{R}^{d_m \times (d_v + d_a + d_t)}$ denotes the weight matrix parameter. However, such multimodal fusion functions with static parameters cannot model the various relationships among multiple modalities in different micro-videos.

3.3 Problem Set-Up of Dynamic Multimodal Fusion

We focus on the dynamic multimodal fusion problem in micro-video recommendation. In this article, static fusion regards the multimodal fusion of all items as the same task. In contrast, we treat multimodal fusion for each item as an individual task in dynamic fusion. We aim to generate the item-specific parameters in a neural network format for each fusion task to achieve dynamic multimodal fusion. As discussed in Section 1, meta-learning is a potential solution due to its ability to learn parameters for multiple tasks and address the few-sample training problem for each task. In particular, we aim to develop a meta-learning model capable of processing an item's multimodal features as input, and generating neural network parameters that can be utilized to fuse the different modalities of the item. Let $\mathcal{X}_i = \{\mathbf{x}_i^v, \mathbf{x}_i^a, \mathbf{x}_i^t\}$ be the set of multimodal features belonging to item i. The learning task of item i is to obtain its multimodal representation as the output of a neural network fusion function $f(\mathcal{X}_i|\Theta_i)$ where the parameters Θ_i are produced from the multimodal features to be fused:

$$\mathbf{e}_i^m = f(\mathcal{X}_i|\mathcal{G}(\mathcal{X}_i)). \tag{5}$$

The meta-learning refers to learning the function $\mathcal{G}(\mathcal{X}_i)$ that takes multimodal features $\mathcal{X}_i$ as input and produces the parameters of the neural network $f(\mathcal{X}_i|\Theta_i)$. In the following section, we describe the framework for learning $\mathcal{G}(\mathcal{X}_i)$ in detail.

4 METHOD

We first present the general MetaMMF framework as illustrated in Figure 2, elaborating on how to learn to parameterize a neural layer for fusing the multimodal features of a micro-video. Based on this, we then extend the single-layer structure into a multi-layer one. To lighten the space complexity of MetaMMF, we present a parameter simplification method based on tensor decomposition. Furthermore, we show that MetaMMF is model-agnostic and can be used as the early step prior to other representation learning functions, such as MF and GCN. Lastly, we provide the optimization of MetaMMF.

4.1 General One-Layer Fusion Framework

As discussed in Section 1, we view dynamic multimodal fusion from a meta-learning perspective, and regard the multimodal fusion of each micro-video as an individual learning task. Formulated as Equation (5), we propose a meta-learning model MetaMMF to learn a neural network for each task in terms of the micro-video's multimodal information, and thus yield an item-specific multimodal fusion function for representation learning. We first illustrate how MetaMMF dynamically learns the parameter of a neural layer, and then generalizes it to a multi-layer network.

Specifically, a meta-learning function $\mathcal{G}$ is responsible for generating the adaptive parameters of each fusion task in a neural network format by processing the meta information. Considering that multimodal features can reflect the relationships among modalities in micro-videos, we employ them as the source of meta information to decide their fusion parameters. In this way, the dynamic parameter generation of a one-layer fusion framework is defined as:

$$\mathbf{W}_i = \mathcal{G}(\mathcal{X}_i), \tag{6}$$

where $\mathbf{W}_i \in \mathbb{R}^{d_m \times (d_v + d_a + d_t)}$ is the weight matrix of a neural layer specific to item i, produced by the meta-learning function $\mathcal{G}$ from the multimodal features of item i. In particular, the design of function $\mathcal{G}$ is twofold, including meta information extractor and meta-fusion learner.

4.1.1 Meta Information Extractor. In practice, the multimodal features of micro-videos are highly dimensional, redundant, and noisy, which cannot be fed directly to the meta-learning function. Towards this end, meta information extractor is employed to extract informative and
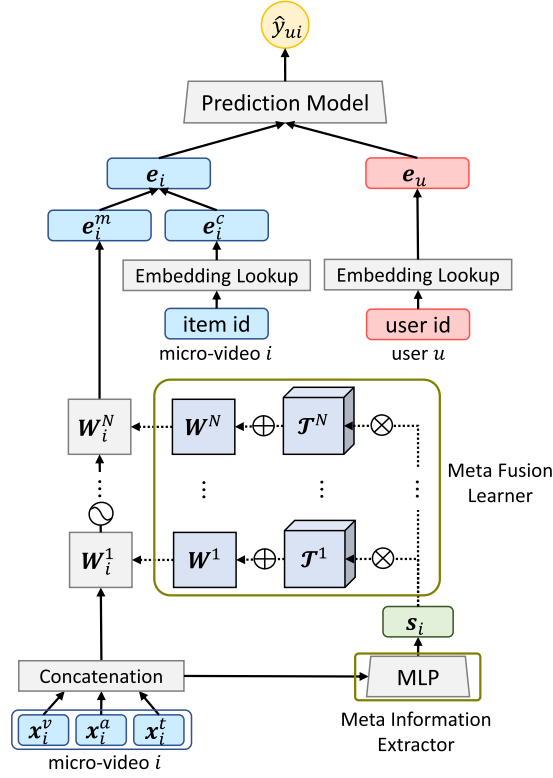
Fig. 2. Schematic illustration of our proposed model.

concise meta information from multimodal features of the target item. In this work, we adopt a **multi-layer perceptron (MLP)** for the extraction,

$$\begin{cases} \phi_{in}(\mathcal{X}_i) = \begin{bmatrix} \mathbf{x}_i^v \\ \mathbf{x}_i^a \\ \mathbf{x}_i^t \end{bmatrix} \\ \mathbf{s}_i = \phi_{out}\left(\phi_{mid}\left(\phi_{in}(\mathcal{X}_i)\right)\right) \end{cases}, \tag{7}$$

where ϕ_{in}, ϕ_{mid}, and ϕ_{out}, respectively, denote the input, hidden, and output layers of MLP, and each layer consists of its own weight matrix, bias vector, and activation function. To be more specific, we uniformly select ReLU as the activation function. After successive multi-layer processing, $\mathbf{s}_i \in \mathbb{R}^{d_s}$ indicates the vector of meta information, wherein the size d_s is far less than the length of input feature $(d_v + d_a + d_t)$. The possibility of other extraction approaches that cannot be ruled out, such as the attention mechanism, will be further explored in future work.

4.1.2 Meta Fusion Learner. Meta fusion learner is responsible for rapidly parameterizing the multimodal function based on the task-dependent meta information. More specifically, meta fusion learner learns the mapping from the meta information $\mathbf{s}_i$, derived from the multimodal features $\mathcal{X}_i$, to the weight matrix parameter $\mathbf{W}_i$. Mathematically, the most direct and practicable way of mapping a 1-D vector into a 2-D weight matrix is to multiply by a 3-D tensor. Hence, in this work,

we partially implement meta fusion learner as:

$$\mathbf{W}_i^* = \mathcal{T} \times_3 \mathbf{s}_i, \tag{8}$$

where $\mathcal{T} \in \mathbb{R}^{d_m \times (d_v + d_a + d_t) \times d_s}$ is the 3-D tensor parameter for meta fusion learner; d_m, $(d_v + d_a + d_t)$, and d_s indicate the argument values of height, width, and depth of $\mathcal{T}$. The operator $\times_3$ denotes that the multiplication is being performed along the tensor's third dimension (d_s). The tensor depth is exactly set equal to the size of meta information vector $\mathbf{s}_i$, in order to make the tensor-vector multiplication along the direction of depth. Thus, the product $\mathbf{W}_i^* \in \mathbb{R}^{d_m \times (d_v + d_a + d_t)}$ is dynamically adaptive to fusing the multimodal features of the target item i.

Beyond this, we also reserve an item-shared weight matrix $\mathbf{W} \in \mathbb{R}^{d_m \times (d_v + d_a + d_t)}$, which is equivalent to the previous static fusion method. This parameter contributes to modeling the elementary and common fusion for all the micro-videos. In comparison, the item-specific weight matrix $\mathbf{W}_i^*$ can be regarded as learning or adaption for the multimodal fusion task of item i. By combining $\mathbf{W}$ and $\mathbf{W}_i^*$, we obtain the parameter underlying a one-layer neural network for addressing the multimodal fusion task of a specific item. The multimodal representation can be learned via such a dynamically parameterized network as:

$$\begin{cases} \mathbf{W}_i = \mathbf{W} + \mathbf{W}_i^* \\ \mathbf{e}_i^m = \mathbf{W}_i \begin{bmatrix} \mathbf{x}_i^v \\ \mathbf{x}_i^a \\ \mathbf{x}_i^t \end{bmatrix} \end{cases} . \tag{9}$$

From the viewpoint of meta-learning, the weight $\mathbf{W}$ denotes the internal parameter that is broadly suitable to many tasks, while the generated parameter $\mathbf{W}_i^*$ can be regarded as fine-tuning the layer weight parameter for quickly adapting to an individual task [9].

4.2 Deep Multi-Layer Fusion Framework

Due to the multi-layer nature of deep neural networks, each successive layer is hypothesized to represent the information more abstractly. It is claimed that most functions that can be represented compactly by deep architectures cannot be represented by a compact shallow architecture [45]. Hence, proposing a multi-layer version of the MetaMMF model is necessary. In brief, multi-layer MetaMMF jointly learns the weight matrix parameters for a succession of neural layers, which are used to fuse the multimodal features step by step. Please kindly note that we omit the bias parameters in a neural layer, without which the performance is still good. Similarly, all the parameters are dynamically generated by taking in the multimodal features of the target item,

$$\{\mathbf{W}_i^n\}_{n=1}^N = \mathcal{G}(\mathcal{X}_i), \tag{10}$$

where N is the number of layers. Specifically, meta information $\mathbf{s}_i$ is likewise extracted from the concatenation of multimodal features as shown in Equation (7). Subsequently, the weight matrices of all layers are determined as follows:

$$\{\mathbf{W}_i^n\}_{n=1}^N = \{\mathbf{W}^n + \mathcal{T}^n \times_3 \mathbf{s}_i\}_{n=1}^N, \tag{11}$$

where the 3-D tensors $\{\mathcal{T}^n\}_{n=1}^N$ and the 2-D matrices $\{\mathbf{W}^n\}_{n=1}^N$ are all trainable parameters for learning the weight matrices $\{\mathbf{W}_i^n\}_{n=1}^N$ of N layers, respectively. All the elements in the sequence of 3-D tensors $\{\mathcal{T}^n\}_{n=1}^N$ have consistent depths, which are equivalent to the length of the meta information vector $\mathbf{s}_i$. Obviously, multi-layer MetaMMF is an extension and stacking of the single-layer ones. In this way, each item can be provided with its individual multi-layer neural network for more complex multimodal fusion. The multimodal representation of item i can be learned via
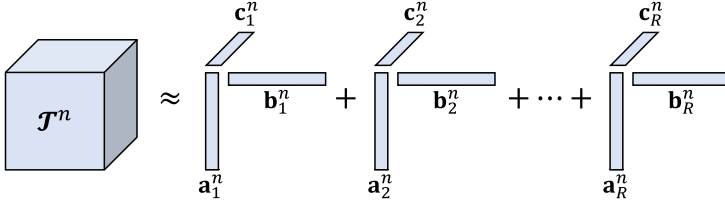
Fig. 3. CP decomposition of a 3-D tensor.

its own N-layer neural network as,

$$\mathbf{e}_i^m = \mathbf{W}_i^N \sigma\left(\mathbf{W}_i^{N-1} \sigma\left(\ldots \sigma\left(\mathbf{W}_i^2 \sigma\left(\mathbf{W}_i^1 \begin{bmatrix} \mathbf{x}_i^v \\ \mathbf{x}_i^a \\ \mathbf{x}_i^t \end{bmatrix}\right)\right)\ldots\right)\right), \tag{12}$$

where σ denotes the activation function of the neural layer to implement the non-linearity of fusion, and we employ LeakyReLU in this work. In terms of the network structure design, we employ a tower pattern. The first hidden layer is the broadest, containing fewer neurons in each subsequent layer. As we progress through the layers, we reduce the number of neurons by half, ensuring that each layer has no less than the output layer size d_m. The rationale behind this design is that deeper layers with fewer hidden neurons can learn more abstract features from the data.

4.3 Model Simplification

Although the multi-layer framework can model a more complex fusion function, the number of parameters inevitably increases as the layers stack. Compared with a static neural network, multi-layer MetaMMF is equipped with extra 3-D tensor parameters for dynamic parameterization, increasing the storage cost and training time. To address this issue, we propose a simplification method using canonical polyadic (CP) decomposition to reduce the number of parameters in MetaMMF. Specifically, CP decomposition factorizes a tensor into a sum of component rank-one tensors, each of which is composed of the outer product of vectors. Through such decomposition, the dimension of the parameters can be greatly reduced. Therefore, we utilize CP decomposition to construct the 3-D tensor parameters used in MetaMMF. As illustrated in Figure 3, we can write the 3-D tensor $\mathcal{T}^n \in \mathbb{R}^{P \times Q \times Z}$ at the nth layer as:

$$\mathcal{T}^n \approx \sum_{r=1}^{R} \mathbf{a}_r^n \circ \mathbf{b}_r^n \circ \mathbf{c}_r^n, \tag{13}$$

where notation $\circ$ denotes the outer product of vectors, R is a positive integer and $\mathbf{a}_r^n \in \mathbb{R}^P$, $\mathbf{b}_r^n \in \mathbb{R}^Q$, and $\mathbf{c}_r^n \in \mathbb{R}^Z$ for $r = 1, \ldots, R$. Elementwise, Equation (13) is written as,

$$t_{pqz}^n \approx \sum_{r=1}^{R} a_{pr}^n b_{qr}^n c_{zr}^n, p = 1, \ldots, P; q = 1, \ldots, Q; z = 1, \ldots, Z. \tag{14}$$

It is proven that the tensor $\mathcal{T}^n$ can be approximatively decomposed into three factor matrices. The factor matrices refer to the combination of the vectors from the rank-one components, i.e., $\mathbf{A}^n = [\mathbf{a}_1^n \ \mathbf{a}_2^n \ \cdots \ \mathbf{a}_R^n]$ and likewise for $\mathbf{B}^n$ and $\mathbf{C}^n$. Using these definitions, we can replace the tensor $\mathcal{T}^n$ with the three matrices $\mathbf{A}^n \in \mathbb{R}^{P \times R}$, $\mathbf{B}^n \in \mathbb{R}^{Q \times R}$, and $\mathbf{C}^n \in \mathbb{R}^{Z \times R}$ to generate dynamic fusion parameters of the n-th layer as,

$$\mathbf{W}_i^n = \mathbf{W}^n + [\![\mathbf{A}^n, \mathbf{B}^n, \mathbf{C}^n]\!] \times_3 \mathbf{s}_i, \tag{15}$$

where $[\![\cdot, \cdot, \cdot]\!]$ indicates the computing process in Equation (13). Comparing the space complexity before and after CP decomposition, we can find it decreasing from $O(P \cdot Q \cdot Z)$ to $O(R \cdot (P + Q + Z))$, since $Q > P \gg R$. When the number of tensors increases with fusion layers, the effect of model simplification will be more significant.

4.4 Model-Agnostic Combination with MF & GCN

As described above, MetaMMF can dynamically provide a multimodal fusion function in the form of a neural network for each item, which integrates the features from multiple modalities into a joint multimodal representation. Benefiting from the end-to-end training of neural networks, MetaMMF can be seamlessly combined with other representation learning frameworks where the joint multimodal representations are used for initialization. In this way, the combined model can take advantage of both multimodal and other useful information, such as collaborative filtering (CF) information and graph structure information. To demonstrate the model-agnostic characteristic of MetaMMF, we present two combined methods to address the micro-video recommendation task.

4.4.1 MetaMMF Plus MF. **Matrix factorization (MF)** is one of the most simple and effective recommendation methods, only utilizing CF signals. MF learns two independent embedding matrices to represent users and items, whose multiplication is used to reconstruct the observed ratings and predict the undiscovered ratings. The joint multimodal representation learned by MetaMMF can be directly injected into the MF framework, and thus combining the multimodal information with the CF signal. Therefore, we propose a combined method named **MetaMMF_MF**. As in VBPR [15], we extend the item collaborative representation $\mathbf{e}_i^c$ with the joint multimodal representation $\mathbf{e}_i^m$ as the complete item representation $\mathbf{e}_i$, like Equation (2). Following the MF framework, we estimate the affinity score between the target user and item by the inner product of their representations,

$$\hat{y}_{ui} = \mathbf{e}_u^\top \mathbf{e}_i = \mathbf{e}_u^\top \begin{bmatrix} \mathbf{e}_i^m \\ \mathbf{e}_i^c \end{bmatrix}, \tag{16}$$

wherein $\mathbf{e}_u \in \mathbb{R}^{(d_m + d_c)}$ and $\mathbf{e}_i^c \in \mathbb{R}^{d_c}$ are both trainable embedding parameters corresponding to user u and item i, responsible for capturing the CF signals.

4.4.2 MetaMMF Plus GCN. **Graph Convolution Network (GCN)** is a state-of-the-art representation learning technology that injects high-order connectivity signals into representations by recursively passing and aggregating messages based on the graph structure. In the recommendation task, GCN is always applied to the user-item interaction graph where the nodes correspond to users and items, connected by historical interactions [26, 28]. Each node is initialized with an embedding, recursively updated by the convolution operation on its neighbor nodes. In most cases, the node initialization involves individual features, e.g., identity and content features. Hence, nodes can finally obtain higher-quality representations by aggregating information from multi-hop neighbors.

However, such graph convolution operation only conducts on unimodal representation. For information of multiple modalities, some methods construct multiple interaction graphs for all modalities, such as MMGCN [44]. In comparison, MetaMMF can integrate the information from multiple modalities into a joint representation, directly serving as the node initialization of an interaction graph. Therefore, we employ MetaMMF in the initial layer of GCN, and propose a combined method called **MetaMMF_GCN**. Specifically, user u and item i can be initialized as:

$$\mathbf{e}_u^{(0)} = \mathbf{e}_u, \ \mathbf{e}_i^{(0)} = \begin{bmatrix} \mathbf{e}_i^m \\ \mathbf{e}_i^c \end{bmatrix} \tag{17}$$

where $\mathbf{e}_u^{(0)}$ and $\mathbf{e}_i^{(0)}$ denote the initial user and item representations at the 0th layer of GCN. Similarly, $\mathbf{e}_u^{(0)}$ is an entirely trainable embedding of user u, while $\mathbf{e}_i^{(0)}$ is obtained by concatenating the joint multimodal representation derived from MetaMMF and the trainable collaborative representation of item i. Based upon the representations at the 0th layer, we recursively formulate the item i's representation at the lth layer as:

$$\mathbf{e}_i^{(l)} = \sigma \left(\mathbf{W}_1 \mathbf{e}_i^{(l-1)} + \mathbf{W}_2 \cdot \text{mean}_{u \in \mathcal{N}_i} \mathbf{e}_u^{(l-1)} \right), \tag{18}$$

where $\mathcal{N}_i$ denotes the neighborhood set of item i, i.e., the users whom item i directly interacted with. σ denotes the LeakyReLU activation function. Note that we utilize the most common graph convolution operation in [13], and other more complicated choices are left to be explored. Analogously, we can obtain the representation $\mathbf{e}_u^{(l)}$ for user u by propagating information from its connected items. For convenience, we denote the final representations at depth L as $\mathbf{e}_u^{(L)}$ and $\mathbf{e}_i^{(L)}$. The inner product of the final user and item representations is used to predict their affinity score like most GCN-based methods do.

4.5 Optimization

In this work, we optimize the proposed models based on implicit feedback, such as clicks, views, and purchases. Compared to explicit ratings, implicit feedback is easier to collect but more challenging to model user preferences [12], due to its scarcity of negative feedback. To learn model parameters, we optimize the pairwise BPR loss [34], which is often used in recommender systems. It considers the relative order between observed and unobserved user-item interactions. Specifically, BPR assumes that the items in observed interactions reflect a user's preference more than those in unobserved ones and should be assigned with higher prediction scores. The objective function is as follows:

$$\mathcal{L} = \sum_{(u,i,j) \in \mathcal{D}} -\ln \sigma(\hat{y}_{ui} - \hat{y}_{uj}) + \lambda ||\Phi||_2^2, \tag{19}$$

where $\mathcal{D} = \{(u,i,j)|(u,i) \in \mathcal{R}^+, (u,j) \in \mathcal{R}^-\}$ denotes the pairwise training data, $\mathcal{R}^+$ indicates the observed interactions, and $\mathcal{R}^-$ is the unobserved interactions; $\sigma(\cdot)$ is the sigmoid function; Φ denotes all trainable model parameters, and λ controls the L_2 regularization strength to prevent overfitting. We adopt mini-batch Adam [21] to optimize the prediction model and update the model parameters. In particular, for a batch of randomly sampled triples $(u,i,j) \in \mathcal{D}$, we establish their representations after MetaMMF and the subsequent MF or GCN, and then update model parameters by using the gradients of the loss function.

5 EXPERIMENTAL SETUP

In this section, the evaluation datasets are presented first, followed by an explanation of the experimental settings and an elaboration of baseline methods.

5.1 Datasets

As micro-videos contain rich multimodal information—frames, soundtracks, and descriptions, the experimental datasets should include such multimodal data for each item. Following MMGCN [44] and GRCN [43], we performed experiments on three publicly accessible datasets designed for

Table 1. Statistics of the Evaluation Datasets

Dataset	#Users	#Items	#Interactions	Density	d_v	d_a	d_t
MovieLens	55,485	5,986	1,239,508	0.37%	2048	128	100
TikTok	36,656	76,085	726,065	0.03%	128	128	128
Kwai	7,010	86,483	298,492	0.05%	2048	–	128

(d_v, d_a, and d_t denote the dimensions of visual, acoustic, and textual modality feature data, respectively.)

micro-video personalized recommendation, including MovieLens,[2] TikTok,[3] and Kwai.[4] Table 1 summarizes the statistics of the datasets.

- *MovieLens:* This dataset is widely adopted to evaluate personalized recommendation. To adapt it for micro-video recommendation, we collected the descriptions of movies from the MovieLens-10M, and crawled the movies' trailers from Youtube.[5] Then, the pre-trained ResNet50 [14] extracted the visual features from these trailers' keyframes. FFmpeg[6] and VG-Gish [17] were adopted to separate audio tracks and learn the acoustic features, respectively. Sentence2Vector [1] was utilized to derive the textual features from the movies' descriptions. Our experiments treated all ratings as the implicit feedback between the corresponding user-item pairs.
- *TikTok:* This dataset is released by the micro-video sharing platform TikTok, which allows users to create and share micro-videos. It consists of users, micro-videos, and their interactions (e.g., clicks). The micro-video features in each modality are extracted and published without providing the raw data. Particularly, the textual features are extracted from the micro-video captions.
- *Kwai:* As a micro-video service provider, Kwai released a sizable micro-video dataset. The dataset contains users, micro-videos, and user behavior records with timestamps. To evaluate the proposed method from implicit feedback, we collected some click records of the corresponding users and micro-videos in a certain period. The acoustic features are unavailable, in contrast to the datasets mentioned above.

For each dataset, we randomly split the historical interactions of each user into three folds: 80% for training, 10% for validation, and the remaining 10% for testing. For each observed user-item interaction in the training set, we treated it as a positive instance. Then we adopted the negative sampling strategy to pair it with negative items the user did not interact with in the training set. This approach contrasts with that of [43], which used the entire dataset for negative sampling. To ensure a fair comparison, the negative sampling setup remains consistent across our methods and the compared baselines. The constructed triples are used for parameter optimization. The validation and testing sets are used to tune the hyper-parameters and evaluate the performance in the experiments, respectively.

5.2 Experimental Settings

Evaluation Metrics. For each user in the testing set, we regarded the items that the user did not interact with as the negative ones. Then each recommendation method predicts the user's preference

---

[2]https://movielens.org/
[3]https://www.tiktok.com/
[4]https://www.kwai.com/
[5]https://www.youtube.com/
[6]http://ffmpeg.org/

scores over all items, except the ones used in the training set. To measure the performance of top-K recommendation and preference ranking, we ranked the items in descending order and adopted the widely used evaluation metrics: Precision@K, Recall@K, HR@K (Hit Ratio), and NDCG@K (Normalized Discounted Cumulative Gain). By default, we set $K = 10$ and 20, respectively. We reported the average metrics of all users in the testing set.

Model Implementation. We implemented the two combined models proposed above MetaMMF_MF and MetaMMF_GCN via the development tools Pytorch[7] and Pytorch Geometric.[8] The user/item representation size is fixed at 64. We optimized our model using the Adam optimizer with a batch size of 3,000, searching in batches of {500, 1,000, 2,000, 3,000}. Besides, we initialized all model parameters using the well-known Xavier approach. To tune the hyperparameters, we applied a grid search based on the results from the validation set. Specifically, we searched for the optimal learning rate from {1e-4, 5e-4, 1e-3, 5e-3} and ultimately set it to 1e-3. The coefficient λ of L_2 normalization was searched within {1e-8, 1e-7, . . . , 1}, and the optimal value was determined to be 1e-6. As another critical model hyperparameter, the length of the meta information vector, denoted by d_s, was fine-tuned among the values of {2, 3, . . . , 10}, and eventually fixed at 5. Moreover, if Precision@K on the validation set does not rise for ten successive epochs, early stopping is adopted. Experimentally, the optimal layer number for MetaMMF on the MovieLens dataset is 4. The sequence of neuron sizes from the input layer to the output layer is as follows: $2276 \rightarrow 1024 \rightarrow 512 \rightarrow 256 \rightarrow 32$. On the TikTok and Kwai datasets, the optimal layer number for MetaMMF is 1. The sequences of neuron sizes are $384 \rightarrow 32$ and $2176 \rightarrow 32$, respectively.

5.3 Baseline Comparison

To demonstrate the effectiveness of our method on micro-video recommendation, we compared it with the following state-of-the-art methods. These baselines can be briefly divided into three groups: MF-based (i.e., VBPR), GCN-based (i.e., GraphSAGE, NGCF, LightGCN, and GAT), and multimodal (MMGCN, GRCN, LATTICE, and InvRL) methods.

- **VBPR [15]:** This is a benchmark model in the multimodal recommendation. It incorporates content information with the MF framework to predict the interactions between users and items. For adapting to micro-video recommendation, we concatenated the multimodal features of micro-videos as content information.
- **GraphSAGE [13]:** With the trainable aggregation function, this model can pass the message along the graph structure and collect them to update each node's representation. It considers both the structure information and the distribution of node features in the neighborhood. For making a fair comparison, we integrated multimodal features as node features to learn the representations.
- **NGCF [40]:** This model presents a novel recommendation framework to integrate the interaction information into the representation learning process. By exploiting the high-order connectivity from user-item interactions, the model encodes the CF signals into the representations. For a fair comparison, we fed the multimodal features of micro-videos into the framework to predict the user-item interactions.
- **LightGCN [16]:** This light yet effective model includes the essential ingredients of GCN for recommendation. This model adopts the simple weighted sum aggregator as a graph convolution operation and combines the representations obtained at each layer to form the final representation.

- **GAT [39]:** This GCN-based method is able to automatically learn and specify different weights to the neighbors of each node. With the learned weights, it alleviates the noisy information from the neighbors to improve the GCN performance.
- **MMGCN [44]:** This is a framework designed for micro-video recommendation. It allocates an independent GCN for each modality, which learns the modality-specific user preferences and item representations via the propagation of modality information on the user-item graph. The outputs of these GCNs are integrated as the final representations for recommendation.
- **GRCN [43]:** This is a state-of-the-art method for multimodal recommendation with implicit feedback. It adaptively adjusts the structure of the interaction graph according to the status of model training, instead of keeping the structure fixed. Based on the refined graph, it applies a graph convolution layer to distill informative signals on user preference.
- **LATTICE [46]:** This is an advanced method for multimodal recommendation, which discovers the item relationship about multimodal features to learn a graph structure. Along the graph structure, graph convolutions can aggregate informative high-order affinities from neighbors for each item. Finally, the model makes recommendation by combining downstream CF methods.
- **InvRL [8]:** This is a state-of-the-art method for invariant representation learning in multimodal recommendation. InvRL can eliminate spurious correlations using heterogeneous environments to learn consistent invariant item representations across different environments. These invariant representations are then used to make accurate predictions of user-item interactions.

Baseline Implementation. To ensure the consistency of baseline implementation, we adopted the publicly available implementations of GraphSAGE,[9] NGCF,[10] LightGCN[11], GAT,[12] MMGCN,[13] GRCN,[14] LATTICE,[15] and InvRL.[16]. Without specification, the default size of user/item representation is 64. For all the GCN-based baselines, we uniformly set a two-layer structure with parameters initialized from Xavier, and used the Adam optimizer with a well-chosen mini-batch size for model optimization. The learning rate is tuned amongst {1e-4, 5e-4, 1e-3, 5e-3, 1e-2}.

6 EXPERIMENTAL RESULTS

To validate the effectiveness of our proposed method, we conducted extensive quantitative and qualitative experiments to answer the following research questions:

- **RQ1:** Can our proposed method outperform the state-of-the-art baselines in the micro-video recommendation task?
- **RQ2:** How does the depth of fusion layers affect MetaMMF?
- **RQ3:** How do the representations benefit from the dynamic multimodal fusion?
- **RQ4:** How do the CP decomposition of parameters affect the convergence and the performance of MetaMMF?
- **RQ5:** How do the static weights affect MetaMMF?

---

[9]https://github.com/williamleif/GraphSAGE
[10]https://github.com/xiangwang1223/neural_graph_collaborative_filtering
[11]https://github.com/gusye1234/LightGCN-PyTorch
[12]https://github.com/PetarV-/GAT
[13]https://github.com/weiyinwei/MMGCN
[14]https://github.com/weiyinwei/GRCN
[15]https://github.com/CRIPAC-DIG/LATTICE
[16]https://github.com/nickwzk/InvRL

6.1   Performance Comparison (RQ1)

Table 2 presents the results of our methods and baselines over the experimental datasets. Besides, it reports the improvements and statistical significance test, which are calculated between our proposed method and the strongest baseline (highlighted with underline). From the comparison results, we noted the following findings:

- VBPR obtains poor performance on three datasets. This indicates that the static fusion function is insufficient to fit the distinct multimodal fusion of different micro-videos, and thus easily leads to suboptimal multimodal representations. Similarly under the MF framework, MetaMMF_MF consistently outperforms VBPR across all cases, demonstrating the rationality and effectiveness of dynamic multimodal fusion. Despite obtaining rather competitive performance, MetaMMF_MF is still slightly inferior to the GCN-based methods due to the limitation of MF.

- Compared with the MF-based methods, it is obvious that representation learning can be further boosted via aggregation and propagation on the graph structure, since GCN-based methods show better performance on micro-video recommendation. More specifically, NGCF generally outperforms GraphSAGE since it injects high-order connectivities among users and items into its propagation layers. LightGCN achieves certain improvements over NGCF after removing some useless nonlinear operations in the original model. Meanwhile, GAT obtains remarkable performance attributed to the attention mechanism which can specify the adaptive weights of neighbor nodes.

- As a model specifically designed for micro-video recommendation, MMGCN slightly outperforms the general GCN-based methods in most cases. One possible reason is that MMGCN sufficiently leverages the multimodal information by message passing of each modality and information interchange across modalities. Furthermore, GRCN adaptively adjusts the false-positive edges of the interaction graph to discharge the ability of GCN, thus outperforming MMGCN across all datasets. In addition to constructing the user-item graph for representation learning, LATTICE also creates modality-aware graphs that accurately capture the similarity of modal features between items, resulting in improved performance. Furthermore, the recently presented InvRL is confirmed to be the strongest baseline. InvRL is groundbreaking in that it learns invariant item representations to mitigate the influences of spurious correlations that are not adequately addressed by other methods.

- MetaMMF_GCN consistently yields the best performance across all datasets. In particular, the improvements of MetaMMF_GCN over the strongest competitor w.r.t. NDCG@10 are 5.27%, 6.54%, and 4.46% in MovieLens, TikTok, and Kwai, respectively. Additionally, we conducted one-sample t-tests, which reveal that the improvements of MetaMMF_GCN over the strongest baseline are statistically significant (p-value < 0.05). The primary difference between MetaMMF and baselines lies in whether the multimodal fusion is dynamic or static. Hence, we attribute the significant improvements to our dynamic fusion strategy. MetaMMF can effectively model the complex relationships among the modalities in different items during representation learning by dynamically learning the item-specific fusion networks. This underscores the importance of dynamic multimodal fusion in micro-video recommendation.

- Moreover, MetaMMF_GCN significantly outperforms its MF version. It is reasonable since MetaMMF_GCN further boosts the multimodal representations learned from MetaMMF via the message passing function of GCN, while MetaMMF_MF directly uses the raw representations. This demonstrates that MetaMMF can be combined with other representation learning frameworks, and achieve better performance by exploiting the advantages of both frameworks to the full.

Table 2. Overall Performance Comparison between our Model and the Baselines on Three Datasets

Methods	MovieLens							
	P@10	P@20	R@10	R@20	HR@10	HR@20	N@10	N@20
VBPR	0.0449	0.0348	0.1927	0.2875	0.1512	0.2340	0.1207	0.1494
MetaMMF_MF	0.0457	0.0353	0.1941	0.2912	0.1539	0.2375	0.1268	0.1562
GraphSAGE	0.0493	0.0363	0.1979	0.3049	0.1601	0.2445	0.1351	0.1653
NGCF	0.0539	0.0378	0.2132	0.3173	0.1658	0.2542	0.1366	0.1685
LightGCN	0.0557	0.0391	0.2318	0.3359	0.1753	0.2646	0.1525	0.1845
GAT	0.0564	0.0404	0.2250	0.3344	0.1813	0.2721	0.1523	0.1855
MMGCN	0.0569	0.0420	0.2310	0.3522	0.1900	0.2851	0.1535	0.1977
GRCN	0.0572	0.0424	0.2487	0.3558	0.1925	0.2867	0.1653	0.1999
LATTICE	0.0575	0.0426	0.2500	0.3562	0.1939	0.2871	0.1669	0.2010
InvRL	0.0575	0.0424	0.2518	0.3584	0.1934	0.2869	0.1666	0.2008
MetaMMF_GCN	**0.0599**	**0.0442**	**0.2613**	**0.3702**	**0.2018**	**0.2974**	**0.1757**	**0.2090**
% Improv.	4.17%	3.76%	3.77%	3.29%	4.07%	3.59%	5.27%	3.98%
p-value	1.12e-4	1.97e-4	1.85e-2	4.26e-3	2.38e-4	1.68e-3	3.05e-3	5.73e-3
Methods	TikTok							
	P@10	P@20	R@10	R@20	HR@10	HR@20	N@10	N@20
VBPR	0.0138	0.0111	0.0600	0.0945	0.0496	0.0799	0.0397	0.0500
MetaMMF_MF	0.0143	0.0119	0.0673	0.1062	0.0521	0.0854	0.0422	0.0531
GraphSAGE	0.0150	0.0124	0.0718	0.1134	0.0539	0.0893	0.0436	0.0560
NGCF	0.0172	0.0141	0.0845	0.1335	0.0617	0.1014	0.0513	0.0658
LightGCN	0.0184	0.0150	0.0921	0.1455	0.0705	0.1160	0.0565	0.0726
GAT	0.0187	0.0152	0.0933	0.1474	0.0707	0.1163	0.0573	0.0736
MMGCN	0.0186	0.0148	0.0927	0.1436	0.0706	0.1149	0.0570	0.0724
GRCN	0.0195	0.0162	0.0948	0.1480	0.0717	0.1162	0.0586	0.0745
LATTICE	0.0199	0.0164	0.0977	0.1535	0.0714	0.1177	0.0583	0.0754
InvRL	0.0203	0.0166	0.1002	0.1559	0.0726	0.1179	0.0612	0.0779
MetaMMF_GCN	**0.0217**	**0.0172**	**0.1065**	**0.1612**	**0.0775**	**0.1237**	**0.0652**	**0.0816**
% Improv.	6.90%	3.61%	6.29%	3.40%	6.75%	4.92%	6.54%	4.75%
p-value	3.71e-3	7.81e-3	1.65e-3	2.08e-2	6.64e-4	3.92e-3	3.17e-3	6.95e-3
Methods	Kwai							
	P@10	P@20	R@10	R@20	HR@10	HR@20	N@10	N@20
VBPR	0.0108	0.0088	0.0302	0.0441	0.0220	0.0359	0.0221	0.0283
MetaMMF_MF	0.0139	0.0102	0.0388	0.0536	0.0285	0.0418	0.0336	0.0384
GraphSAGE	0.0135	0.0105	0.0351	0.0490	0.0285	0.0429	0.0318	0.0364
NGCF	0.0138	0.0106	0.0378	0.0553	0.0282	0.0434	0.0334	0.0391
LightGCN	0.0153	0.0114	0.0417	0.0606	0.0314	0.0468	0.0357	0.0420
GAT	0.0146	0.0111	0.0419	0.0609	0.0299	0.0455	0.0360	0.0417
MMGCN	0.0140	0.0114	0.0431	0.0614	0.0309	0.0466	0.0361	0.0419
GRCN	0.0154	0.0116	0.0456	0.0649	0.0320	0.0479	0.0367	0.0428
LATTICE	0.0156	0.0117	0.0460	0.0656	0.0323	0.0482	0.0379	0.0435
InvRL	0.0159	0.0119	0.0463	0.0668	0.0324	0.0489	0.0381	0.0444
MetaMMF_GCN	**0.0166**	**0.0122**	**0.0480**	**0.0680**	**0.0340**	**0.0500**	**0.0398**	**0.0453**
% Improv.	4.40%	2.52%	3.67%	1.80%	4.94%	2.25%	4.46%	2.03%
p-value	3.44e-3	1.03e-2	4.50e-3	6.64e-3	1.05e-2	1.19e-2	4.51e-3	5.66e-3

6.2 Effect of Layer Number on MetaMMF (RQ2)

To verify the fusion advantage of MetaMMF, we compared it with neural network-based early fusion [31], late fusion [19], and hybrid fusion [22] methods. We connected the output of these fusion methods with MF and GCN, respectively. Additionally, we examined the impact of layer numbers on MetaMMF's performance, considering the crucial role played by the meta-learned neural layer in MetaMMF. In particular, we varied the model depth in the $\{1, 2, 3, 4\}$ range. The experimental results of these methods are summarized in Table 3, where MetaMMF_MF-3 and MetaMMF_GCN-3 denote our two methods with three meta-learned fusion layers, and similar notations are used for the others. By analyzing the results presented in Table 3, we have made the following findings:

- On the datasets with less sparsity (e.g., MovieLens), increasing the depth of MetaMMF substantially improves the performance when combined with both MF and GCN. The phenomenon validates that multi-layer MetaMMF is capable of modeling fine-grained multimodal fusion for items and thus enhances representation learning. On the sparse datasets (e.g., TikTok and Kwai), increasing the number of meta-learned fusion layers has no positive effect on the performance of MetaMMF. Especially when combined with MF, one-layer fusion achieves the best performance, which is consistent with the prior VBPR [15]. The probable reason is that MF easily suffers from over-fitting with sparse data, and redundant fusion layers actually make this problem worse. When followed by GCN, stacking fusion layers also has little impact on model performance. This is because the message passing of GCN can effectively alleviate the data sparsity problem with just one fusion layer [40].
- Among the three kinds of fusion methods, early fusion consistently outperforms late fusion. Early fusion is the feature-level fusion that directly integrates multimodal features to learn the representation. By contrast, late fusion is at the decision level, overlooking the feature-level fusion among the modalities. This shows that late fusion is inapplicable to micro-video recommendation based on representation learning. Although the hybrid fusion method exploits both fusion methods in a common framework, it is inferior to the single early fusion in some cases. The possible reason is that the late fusion part drags down the performance.
- We performed one-sample t-tests on various datasets and metrics to compare the best fusion strategies (highlighted with underlines) with our methods that employ optimal layers (highlighted in bold). The results are recorded in Table 3, where a p-value < 0.05 indicates a statistically significant improvement. In the MF framework, we observed that MetaMMF significantly outperforms the early, late, and hybrid fusion methods when utilizing the optimal number of fusion layers. On the other hand, in the GCN framework, MetaMMF consistently outperforms all other fusion strategies, regardless of the number of fusion layers employed. This suggests that MetaMMF is better suited for collaborating with complex models, such as GCN, than other static fusion strategies. Our findings confirm the effectiveness of MetaMMF, demonstrating that dynamic multimodal fusion can significantly enhance the micro-video recommendation task.

6.3 Effect of Dynamic Multimodal Fusion (RQ3)

In this section, we attempt to understand how our dynamic multimodal fusion strategy facilitates representation learning in the embedding space. Therefore, we randomly selected six users from TikTok associated with their relevant items. Note that the items are from the test set, which are not paired with users in the training phase. By employing the t-Distributed Stochastic Neighbor Embedding (t-SNE) [37] in 2-dimension, Figures 4(a)-4(d) visualize the representations derived from

Table 3. Effect of Fusion Neural Layer Numbers

Methods	MovieLens		TikTok		Kwai	
	P@10	R@10	P@10	R@10	P@10	R@10
Early_MF	0.0448	0.1923	0.0137	0.0639	0.0118	0.0326
Late_MF	0.0399	0.1730	0.0131	0.0598	0.0108	0.0304
Hybrid_MF	0.0407	0.1766	0.0134	0.0651	0.0117	0.0321
MetaMMF_MF-1	0.0443	0.1844	**0.0143**	**0.0673**	**0.0139**	**0.0388**
MetaMMF_MF-2	0.0447	0.1884	0.0134	0.0609	0.0122	0.0353
MetaMMF_MF-3	0.0455	0.1930	0.0128	0.0580	0.0119	0.0349
MetaMMF_MF-4	**0.0457**	**0.1941**	0.0124	0.0584	0.0110	0.0337
% Improv.	2.01%	0.94%	4.38%	3.38%	17.80%	19.02%
p-value	3.37e-4	1.02e-4	3.13e-4	2.38e-4	6.74e-3	3.54e-3
Early_GCN	0.0573	0.2446	0.0186	0.0925	0.0151	0.0406
Late_GCN	0.0568	0.2264	0.0188	0.0890	0.0140	0.0386
Hybrid_GCN	0.0572	0.2477	0.0189	0.0884	0.0151	0.0396
MetaMMF_GCN-1	0.0584	0.2554	**0.0217**	**0.1065**	**0.0166**	**0.0480**
MetaMMF_GCN-2	0.0592	0.2587	0.0214	0.1047	0.0160	0.0452
MetaMMF_GCN-3	0.0595	0.2590	0.0211	0.1043	0.0153	0.0439
MetaMMF_GCN-4	**0.0599**	**0.2613**	0.0211	0.1040	0.0151	0.0438
% Improv.	4.54%	5.49%	14.81%	15.14%	9.93%	18.23%
p-value	3.03e-4	1.97e-3	2.01e-3	1.66e-3	2.62e-3	3.99e-3
Methods	MovieLens		TikTok		Kwai	
	HR@10	N@10	HR@10	N@10	HR@10	N@10
Early_MF	0.1508	0.1204	0.0495	0.0392	0.0242	0.0282
Late_MF	0.1341	0.1120	0.0471	0.0391	0.0219	0.0225
Hybrid_MF	0.1370	0.1141	0.0482	0.0401	0.0239	0.0286
MetaMMF_MF-1	0.1491	0.1208	**0.0521**	**0.0422**	**0.0285**	**0.0336**
MetaMMF_MF-2	0.1516	0.1228	0.0515	0.0414	0.0249	0.0318
MetaMMF_MF-3	0.1534	0.1252	0.0482	0.0379	0.0243	0.0283
MetaMMF_MF-4	**0.1539**	**0.1268**	0.0462	0.0307	0.0235	0.0272
% Improv.	2.06%	5.32%	5.25%	5.24%	17.77%	17.48%
p-value	4.90e-4	6.09e-4	6.76e-4	5.47e-4	1.23e-2	7.05e-3
Early_GCN	0.1929	0.1654	0.0704	0.0568	0.0310	0.0346
Late_GCN	0.1837	0.1595	0.0675	0.0539	0.0287	0.0324
Hybrid_GCN	0.1925	0.1671	0.0678	0.0543	0.0309	0.0348
MetaMMF_GCN-1	0.1965	0.1709	**0.0775**	**0.0652**	**0.0340**	**0.0398**
MetaMMF_GCN-2	0.1992	0.1731	0.0767	0.0644	0.0327	0.0380
MetaMMF_GCN-3	0.2003	0.1736	0.0756	0.0640	0.0314	0.0375
MetaMMF_GCN-4	**0.2018**	**0.1757**	0.0757	0.0630	0.0308	0.0374
% Improv.	4.61%	5.15%	10.09%	14.79%	9.68%	14.37%
p-value	7.52e-4	2.59e-3	4.58e-3	5.64e-3	1.87e-3	5.43e-3

VBPR, MetaMMF_MF, InvRL, and MetaMMF_GCN, respectively. By analyzing how their representations distribute, we have two key observations:

- The connectivities of users and items are well reflected in the embedding space, i.e., embedded into the nearest parts of the space, resulting in discernible clustering. Compared to
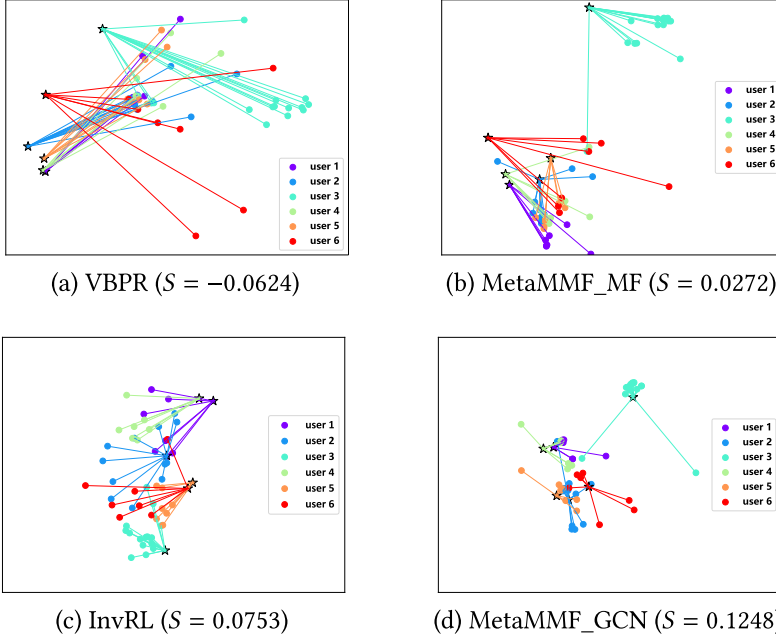
Fig. 4. The visualization depicts the t-SNE transformed representations obtained from our methods and baselines. Each star corresponds to a user from the TikTok dataset, while points with the same color signify relevant items. A link between a star and a point represents their interaction. For optimal viewing, please refer to the colored version. The notation S indicates the mean silhouette coefficient of the clustering result for the sample representations.

VBPR and InvRL, our MetaMMF_MF and MetaMMF_GCN models exhibit even more noticeable clustering, with points of the same color (i.e., items interacted with by the same users) forming clusters. To evaluate the clustering performance of our methods and baselines, we computed their mean silhouette coefficients (S) for all samples shown in Figure 4. This coefficient measures the effectiveness and reasonableness of the clustering, with a value ranging between -1 and 1. A larger value indicates that samples from the same class are closer, while samples from different classes are farther apart, implying better clustering performance. Our methods' representations achieve better clustering performance than the competitors, as seen in the S values in Figure 4. This indicates that our MetaMMF model enhances representation learning effectively. We attribute this success to our model's dynamic multimodal fusion, which utilizes and integrates items' multimodal features in an adaptive approach.
- Jointly analyzing the same users across Figure 4, we find that the embeddings of their historical items tend to be closer in our methods. Meanwhile, the dynamic multimodal fusion prompts the item embeddings to distribute more discriminately in the visualization. It verifies that our proposed methods have an advantage in capturing the user-item correlation.

6.4 Effect of Parameter Decomposition (RQ4)

We investigate how the parameter decomposition (i.e., CP decomposition of 3-D tensor parameters) affects model MetaMMF from two aspects of performance and convergence. In particular, we first study how the hyper-parameter R (i.e., the number of components in Equation(13)) from CP
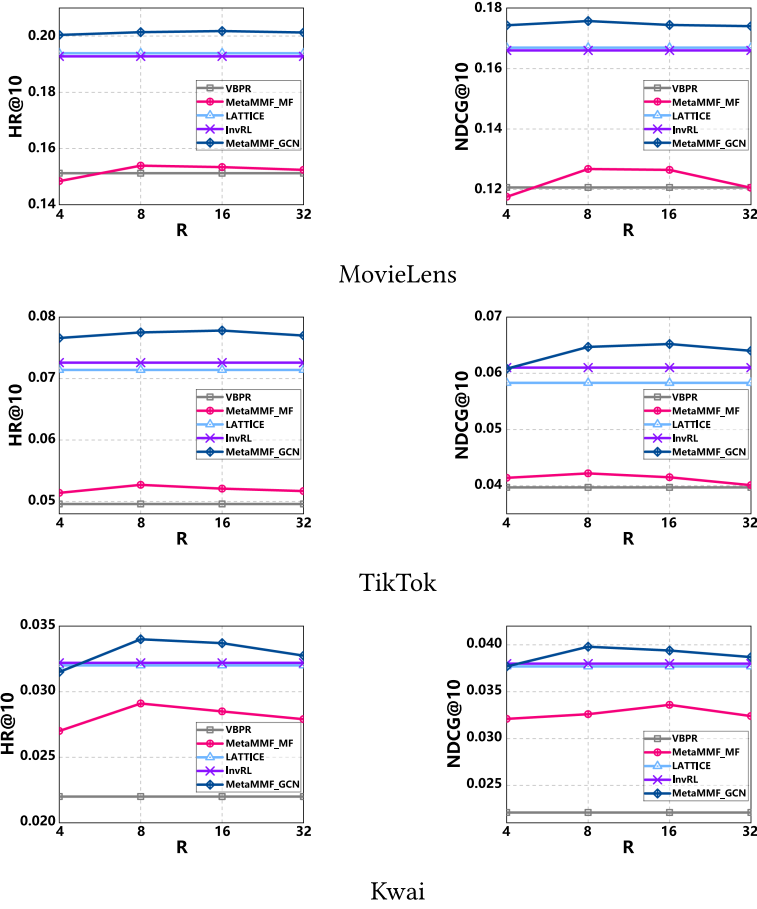
Fig. 5. Effect of hyper-parameter R in CP decomposition.

decomposition affects the performance, and then explore the influence of CP decomposition on training epochs.

6.4.1 Effect of Hyper-parameter R. Figure 5 shows the performance of our methods MetaMMF_MF and MetaMMF_GCN with different numbers of decomposed components, as well as their most competitive baselines, against HR@10 and NDCG@10 evaluation protocols on different datasets. The value of hyper-parameter R is varied in the range of $\{4, 8, 16, 32\}$, while the number of fusion layers is optimal. From Figure 5, we can observe that:

- The hyper-parameter R in CP decomposition indeed affects the performance of our methods. Generally, a small decomposition value leads to the worst performance across all the datasets, since such large-scale decomposition cannot restore the fitting ability of the original 3-D tensor parameters. By increasing the value of R, the CP decomposition can offer better performance while effectively reducing our model's space complexity. However, blindly increasing the R value will not further enhance the performance, but instead burdening the model. The possible reason is that R has already been higher than the rank of the original 3-D tensor, and CP decomposition provides excess capacity.
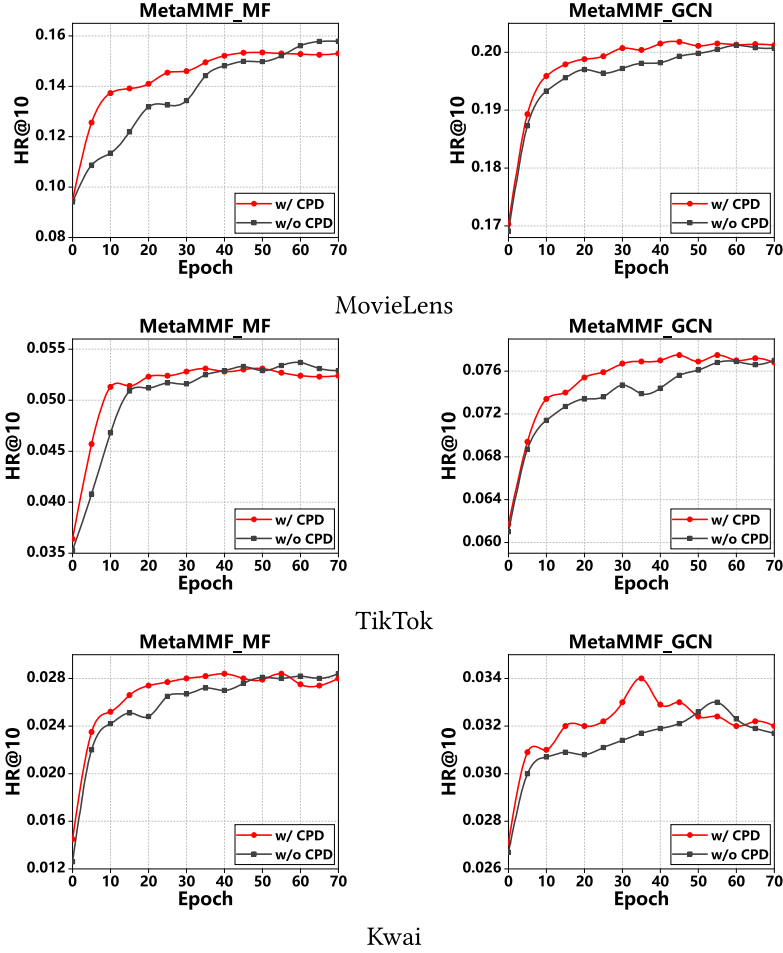
Fig. 6. Test performance of each epoch of MetaMMF_MF and MetaMMF_GCN with and without CP decomposition.

- Despite the influence of R, our methods are consistently superior to VBPR and LATTICE, respectively. This demonstrates that CP decomposition can replace the original 3-D tensor parameters, and retain the ability of dynamic multimodal fusion within our model. With regard to the optimal value of R, we find that MetaMMF_MF and MetaMMF_GCN always achieve the best performance on datasets when the value of R is set to 8 or 16. In such a case, CP decomposition is able to reduce the model to nearly one percent of its original size. Hence, although MetaMMF uses 3-D tensor parameters to replace the 2-D matrix parameters in neural layers, the number of model parameters does not exceed by a wide margin attributing to CP decomposition.

6.4.2   Effect of CP Decomposition on Model Convergence. Considering that CP decomposition effectively reduces the parameters in MetaMMF, we can infer that the simplified model can be trained to converge more easily. To validate this inference, we conducted experiments to explore how CP decomposition affects the convergence rate of the model. Figure 6 shows the test performance w.r.t. HR@10 of MetaMMF_MF and MetaMMF_GCN with (w/) and without (w/o) CP

Table 4. Training Time Comparison of MetaMMF with and without CP Decomposition

Methods	CPD	MovieLens		TikTok		Kwai	
		Converge (eps)	Training time (min/ep)	Converge (eps)	Training time (min/ep)	Converge (eps)	Training time (min/ep)
MetaMMF_MF	w/	45~50	6.07	30~35	2.33	35~40	0.88
	w/o	65~70	8.74	55~60	2.62	50~55	1.05
MetaMMF_GCN	w/	40~45	11.69	40~45	5.08	30~35	1.42
	w/o	55~60	12.41	55~60	5.32	50~55	1.47

decomposition as the training epoch progressing. Due to the space limitation, we omitted the performance w.r.t. other evaluation metrics which has a similar trend. As shown in Figure 6, we can see that:

- On three datasets, the two methods exhibit faster convergence with CP decomposition compared with using the complete 3-D tensor parameters. The gap of convergence rate caused by CP decomposition is more obvious on the MovieLens dataset where MetaMMF is equipped with more fusion layers. This phenomenon is reasonable since CP decomposition can substantially reduce the number of to-be-optimized parameters and thus improve the efficiency of model training.
- Besides, CP decomposition slightly influences the performance of MetaMMF_MF while enhancing the performance of MetaMMF_GCN. The reason might be that the complete MetaMMF is more likely to suffer from over-fitting when combined with a complex GCN model, and CP decomposition resolves this issue by simplifying the model complexity. This demonstrates that CP decomposition benefits MetaMMF in multiple aspects, including size, convergence, and model performance.

To demonstrate the time efficiency resulting from CP decomposition transparently, we measured the training time of two models, MetaMMF_MF and MetaMMF_GCN, both with (w/) and without (w/o) CP decomposition. For a fair comparison, all the models were trained on Ubuntu 16.04.5 with NVIDIA TITAN Xp and Python3.7. Table 4 displays the range of training epochs (eps) when the various variant models achieve convergence. Additionally, we recorded the training time in minutes taken by the variants to complete the unit training epoch (min/ep). From Table 4, it is evident that the decomposed models converge in fewer epochs and require less training time to complete one training epoch. Therefore, we conclude that CP decomposition decreases training time not only by earlier convergence but also by reducing the time taken per epoch. The reason is that CP decomposition effectively reduces the number of parameters that must be optimized.

6.5 Effect of Static Weights (RQ5)

Additionally, we conducted ablation experiments to investigate the impact of static weights on our dynamic model. Table 5 presents the performance of our two methods, with (w/) and without (w/o) static weights, in terms of HR@10 and NDCG@10 on three datasets. The results indicate that the performance of MetaMMF is negatively affected when the static weights are eliminated. This observation emphasizes the importance of shared and static fusion parameters in our dynamic fusion model. To figure out why, we think the static parameters serve as fundamental and universal multimodal fusion components for all micro-videos. From a meta-learning perspective, the static component can be viewed as learning the initialization parameters to generalize most tasks, which lays the groundwork for subsequent fine-tuning to adapt to each unique task.

Table 5. Performance Comparison of MetaMMF with and without Static Weights

Methods	Static weights	MovieLens		TikTok		Kwai	
		HR@10	N@10	HR@10	N@10	HR@10	N@10
MetaMMF_MF	w/	0.1539	0.1268	0.0521	0.0422	0.0285	0.0336
	w/o	0.1460	0.0987	0.0520	0.0403	0.0273	0.0324
MetaMMF_GCN	w/	0.2018	0.1757	0.0775	0.0652	0.0340	0.0398
	w/o	0.2016	0.1753	0.0769	0.0638	0.0312	0.0370

7 CONCLUSION AND FUTURE WORK

In this work, we upgraded the multimodal fusion from static to dynamic to enhance the representation learning in the micro-video recommendation. We devised a novel meta-learning model MetaMMF, which realizes the target by learning the item-specific multimodal fusion functions for different items. The core of MetaMMF is the meta fusion learner which can adaptively parameterize a neural network for an item's multimodal fusion based on the meta information derived from its multimodal features. Extensive experiments on three real-world datasets demonstrate the rationality and effectiveness of leveraging dynamic fusion to learn multimodal representations. In the future, we will exploit MetaMMF to dynamically integrate user preferences on multiple modalities and produce higher-quality user representations, which will be beneficial to recommendation performance. Moreover, we are interested in introducing MetaMMF to address the cold-start problem of multimodal recommendation [7].

This work is an initial attempt at dynamic multimodal fusion, limited to a neural network framework and the micro-video recommendation scenario. Specifically, there are many other types of fusion functions, such as multiple kernel learning methods [11] and graphical models. Furthermore, multimodal fusion is widely applied in various tasks, such as emotion recognition [5], multimedia event detection [22], and multimedia retrieval. We expect the potential of dynamic multimodal fusion can be further explored towards other model structures and tasks.

REFERENCES

[1] Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A simple but tough-to-beat baseline for sentence embeddings. In *ICLR*. 1–16.

[2] Tadas Baltrušaitis, Chaitanya Ahuja, and Louis-Philippe Morency. 2018. Multimodal machine learning: A survey and taxonomy. *TPAMI* 41, 2 (2018), 423–443.

[3] Desheng Cai, Shengsheng Qian, Quan Fang, and Changsheng Xu. 2021. Heterogeneous hierarchical feature aggregation network for personalized micro-video recommendation. *TMM* 24 (2021), 805–818.

[4] Jingyuan Chen, Hanwang Zhang, Xiangnan He, Liqiang Nie, Wei Liu, and Tat-Seng Chua. 2017. Attentive collaborative filtering: Multimedia recommendation with item-and component-level attention. In *SIGIR*. 335–344.

[5] Roddy Cowie, Ellen Douglas-Cowie, Nicolas Tsapatsoulis, George Votsis, Stefanos Kollias, Winfried Fellenz, and John G. Taylor. 2001. Emotion recognition in human-computer interaction. *SPM* 18, 1 (2001), 32–80.

[6] Yue Cui, Hao Sun, Yan Zhao, Hongzhi Yin, and Kai Zheng. 2021. Sequential-knowledge-aware next POI recommendation: A meta-learning approach. *TOIS* 40, 2 (2021), 1–22.

[7] 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*. 3469–3477.

[8] Xiaoyu Du, Zike Wu, Fuli Feng, Xiangnan He, and Jinhui Tang. 2022. Invariant representation learning for multimedia recommendation. In *MM*. 619–628.

[9] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*. 1126–1135.

[10] Yuqian Fu, Li Zhang, Junke Wang, Yanwei Fu, and Yu-Gang Jiang. 2020. Depth guided adaptive meta-fusion network for few-shot video recognition. In *MM*. 1142–1151.

[11] Mehmet Gönen and Ethem Alpaydın. 2011. Multiple kernel learning algorithms. *JMLR* 12 (2011), 2211–2268.

[12] Yangyang Guo, Zhiyong Cheng, Liqiang Nie, Yinglong Wang, Jun Ma, and Mohan Kankanhalli. 2019. Attentive long short-term preference modeling for personalized product search. *TOIS* 37, 2 (2019), 1–27.

[13] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *NeurIPS*. 1024–1034.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR*. 770–778.

[15] Ruining He and Julian McAuley. 2016. VBPR: Visual bayesian personalized ranking from implicit feedback. In *AAAI*. 144–150.

[16] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *SIGIR*. 639–648.

[17] Shawn Hershey, Sourish Chaudhuri, Daniel PW Ellis, Jort F Gemmeke, Aren Jansen, R Channing Moore, Manoj Plakal, Devin Platt, Rif A Saurous, Bryan Seybold, et al. 2017. CNN architectures for large-scale audio classification. In *ICASSP*. 131–135.

[18] Xinxin Hu, Kailun Yang, Lei Fei, and Kaiwei Wang. 2019. Acnet: Attention based network to exploit complementary features for rgbd semantic segmentation. In *ICIP*. 1440–1444.

[19] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using click through data. In *CIKM*. 2333–2338.

[20] Hao Jiang, Wenjie Wang, Yinwei Wei, Zan Gao, Yinlong Wang, and Liqiang Nie. 2020. What aspect do you like: Multi-scale time-aware user interest modeling for micro-video recommendation. In *MM*. 3487–3495.

[21] Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.

[22] Zhen-zhong Lan, Lei Bao, Shoou-I Yu, Wei Liu, and Alexander G Hauptmann. 2014. Multimedia classification and event detection using double fusion. *MTA* 71, 1 (2014), 333–347.

[23] Hoyeop Lee, Jinbae Im, Seongwon Jang, Hyunsouk Cho, and Sehee Chung. 2019. Melu: Meta-learned user preference estimator for cold-start recommendation. In *KDD*. 1073–1082.

[24] Fan Liu, Zhiyong Cheng, Huilin Chen, Anan Liu, Liqiang Nie, and Mohan Kankanhalli. 2022. Disentangled multimodal representation learning for recommendation. *TMM* (2022), 1–11.

[25] Fan Liu, Zhiyong Cheng, Changchang Sun, Yinlong Wang, Liqiang Nie, and Mohan Kankanhalli. 2019. User diverse preference modeling by multimodal attentive metric learning. In *MM*. 1526–1534.

[26] Fan Liu, Zhiyong Cheng, Lei Zhu, Zan Gao, and Liqiang Nie. 2021. Interest-aware message-passing GCN for recommendation. In *WWW*. 1296–1305.

[27] Han Liu, Xiangnan He, Fuli Feng, Liqiang Nie, Rui Liu, and Hanwang Zhang. 2018. Discrete factorization machines for fast feature-based recommendation. In *IJCAI*. 3449–3455.

[28] Han Liu, Yinwei Wei, Jianhua Yin, and Liqiang Nie. 2022. HS-GCN: Hamming spatial graph convolutional networks for recommendation. *TKDE* (2022).

[29] Shang Liu, Zhenzhong Chen, Hongyi Liu, and Xinghai Hu. 2019. User-video co-attention network for personalized micro-video recommendation. In *WWW*. 3020–3026.

[30] Jingwei Ma, Guang Li, Mingyang Zhong, Xin Zhao, Lei Zhu, and Xue Li. 2018. LGA: Latent genre aware micro-video recommendation on social media. *MTA* 77, 3 (2018), 2991–3008.

[31] Youssef Mroueh, Etienne Marcheret, and Vaibhava Goel. 2015. Deep multimodal learning for audio-visual speech recognition. In *ICASSP*. IEEE, 2130–2134.

[32] Tsendsuren Munkhdalai and Hong Yu. 2017. Meta networks. In *ICML*. 2554–2563.

[33] Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y. Ng. 2011. Multimodal deep learning. In *ICML*. 689–696.

[34] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *UAI*. 452–461.

[35] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. 2016. Meta-learning with memory-augmented neural networks. In *ICML*. 1842–1850.

[36] Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In *NeurIPS*. 4080–4090.

[37] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *JMLR* 9, 11 (2008).

[38] Manasi Vartak, Arvind Thiagarajan, Conrado Miranda, Jeshua Bratman, and Hugo Larochelle. 2017. A meta-learning perspective on cold-start recommendations for items. In *NeurIPS*. 6907–6917.

[39] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *ICLR*. 1–12.

[40] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *SIGIR*. 165–174.

[41] Yifan Wang, Xing Xu, Wei Yu, Ruicong Xu, Zuo Cao, and Heng Tao Shen. 2021. Combine early and late fusion together: A hybrid fusion framework for image-text matching. In *ICME*. 1–6.

[42] Yinwei Wei, Xiang Wang, Weili Guan, Liqiang Nie, Zhouchen Lin, and Baoquan Chen. 2019. Neural multimodal cooperative learning toward micro-video understanding. *TIP* 29 (2019), 1–14.

[43] Yinwei Wei, Xiang Wang, Liqiang Nie, Xiangnan He, and Tat-Seng Chua. 2020. Graph-refined convolutional network for multimedia recommendation with implicit feedback. In *MM*. 3541–3549.

[44] 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 *MM*. 1437–1445.

[45] Bengio Yoshua and Lecun Yann. 2007. Scaling learning algorithms towards AI. *Large-scale Kernel Machines* 34, 5 (2007), 1–41.

[46] Jinghao Zhang, Yanqiao Zhu, Qiang Liu, Shu Wu, Shuhui Wang, and Liang Wang. 2021. Mining latent structures for multimedia recommendation. In *MM*. 3872–3880.

[47] Yongchun Zhu, Ruobing Xie, Fuzhen Zhuang, Kaikai Ge, Ying Sun, Xu Zhang, Leyu Lin, and Juan Cao. 2021. Learning to warm up cold item embeddings for cold-start recommendation with meta scaling and shifting networks. In *SIGIR*. 1167–1176.