Adaptive Multi-Modalities Fusion in Sequential Recommendation Systems

Hengchang Hu†‡
hengchang.hu@u.nus.edu
National University of Singapore
Singapore

Yong Liu
liu.yong6@huawei.com
Huawei Noah’s Ark Lab
Singapore

Wei Guo
guowei67@huawei.com
Huawei Noah’s Ark Lab
Singapore

Min-Yen Kan
kannya@comp.nus.edu.sg
National University of Singapore
Singapore

ABSTRACT
In sequential recommendation, multi-modal information (e.g., text or image) can provide a more comprehensive view of an item’s profile. The optimal stage (early or late) to fuse modality features into item representations is still debated. We propose a graph-based approach (named MMSR) to fuse modality features in an adaptive order, enabling each modality to prioritize either its inherent sequential nature or its interplay with other modalities. MMSR represents each user’s history as a graph, where the modality features of each item in a user’s history sequence are denoted by cross-linked nodes. The edges between homogeneous nodes represent intra-modality sequential relationships, and the ones between heterogeneous nodes represent inter-modality interdependence relationships. During graph propagation, MMSR incorporates dual attention, differentiating homogeneous and heterogeneous neighbors. To adaptively assign nodes with distinct fusion orders, MMSR allows each node’s representation to be asynchronously updated through an update gate. In scenarios where modalities exhibit stronger sequential relationships, the update gate prioritizes updates among homogeneous nodes. Conversely, when the interdependence relationships between modalities are more pronounced, the update gate prioritizes updates among heterogeneous nodes. Consequently, MMSR establishes a fusion order that spans a spectrum from early to late modality fusion. In experiments across six datasets, MMSR consistently outperforms state-of-the-art models, and our graph propagation methods surpass other graph neural networks. Additionally, MMSR naturally manages missing modalities. The code is available at: https://github.com/HoldenHu/MMSR.

KEYWORDS
Sequential Recommendation, Graph Neural Network, Multi-Modal

CCS CONCEPTS
• Information systems → Recommender systems.

†Work done when the author is a research intern at Huawei Noah’s Ark Lab, Singapore.
‡Corresponding author.

ACM Reference Format:

1 INTRODUCTION
Recommendation systems leverage user–item interactions to predict future user consumption. Collaborative approaches focus on determining similarity between users/items, while sequential approaches uncover sequential patterns among items. Modality information, such as images or text, has been extensively studied in collaborative recommendation [3, 43, 46, 57], but its potential in sequential recommendation (SR) remains largely unexplored.

In collaborative recommendation, modalities are represented as high-dimensional feature vectors, which are captured through pre-trained models like BERT [9] for texts and ResNet [13] for images. However, incorporating multiple modalities into SR poses two key challenges: (1) Identifying sequential patterns within each modality, as they may exhibit distinct patterns; (2) Capturing the complex interplay between modalities that can influence users’ sequential behavior. For example, many consumers may purchase a suit and then subsequently buy a tie (Figure 1, left). Recognizing meaningful sequential image patterns between suits and ties allows for robust recommendations, independent of specific ID patterns. Moreover, at the item-level, an item is not solely defined by a single modality. Considering different images of suits, the interaction between these images and other modalities (e.g., textual descriptions) also plays a role in influencing a user’s selection.

In Sequential Recommendation, existing approaches for merging different channels of features include early [19, 25, 39] and late fusion [54], which determine whether merging occurs before or after sequential modeling. However, considering the above challenges, both have limitations — early fusion is less sensitive to the interactions between intra-channel features, while late fusion is less sensitive to the interactions among different channels of features.

We conduct a case study as evidence. We utilized both fusion strategies on GRU4Rec [18] and SASRec [22] for pre-experiments on the Amazon dataset [14]. To minimize interference, we had two
settings: randomly shuffling the item-level sequence (disordered) and maintaining the sequence while randomly displacing certain modalities (mismatched). We found late fusion models are more sensitive to the disordered version (resulting in a significant performance drop). In contrast, early fusion is less sensitive to sequential patterns within each channel. Under mismatched conditions, this reversed, with early fusion experiencing a larger performance drop. This indicates that late fusion is less sensitive to restricted modality matching.

These findings reveal that fusion order is crucial. While holistic fusion methods like Trans2D [40] suggest features can be fused without a strict order, they do not address the heterogeneity of feature channels or consider fusion order impact. Therefore, we propose a graph-based holistic fusion method for a flexible modality feature fusion. As existing fusion methods target attribute features [28, 49], we introduce our Multi-Modality enriched Sequential Recommendation (MMSR) framework which focuses on modality feature fusion. Our MMSR framework comprises three stages: representation, where item features in each channel are represented as nodes; fusion, which aggregates features from different channels using graph techniques; and prediction, which generates the final representations. To overcome the limitations of existing methods, we aim to tackle the aforementioned two challenges by: (1) Preserving modalities’ temporal order during fusion, and (2) Enabling effective interactions between multiple modalities.

We represent each user’s behavior history with a graph, where the modality features of items are nodes. We consider three feature channels: item identifier, visual, and textual modalities. Each graph maintains their temporal order as homogeneous relations while capturing cross-modal interactions as heterogeneous relations. Still, challenges persist in graph construction, aggregation, and updating.

Firstly, in graph construction, treating each modality (such as images) as an individual node will overlook their semantic relatedness. Moreover, given the three channels, the number of nodes in the graph processing will triple, significantly increasing the graph’s sparsity. Secondly, graph nodes and relations are typed. During graph aggregation, simply viewing them as homogeneous nodes and relations results in oversimplification, resulting in poor representation and confusing fusion order (similarly, invasive feature fusion across channels also disrupts graph aggregation [28]). Thirdly, naïve graph updating is synchronous for all nodes, unable to support fusion order.

To tackle these issues, we propose solutions. Firstly, to construct graphs, we adopt a similar approach [51] to create compositional embeddings that represent nodes as compositions of smaller groups. Specifically, we cluster modality features and select the identifiers of the cluster centers as modality codes, which are then treated as new nodes in the graph. This approach offers a two-fold advantage: reducing overfitting by having fewer modality nodes during training, and establishing links between items by grouping highly similar modalities under the same node. Secondly, for graph aggregation, we employ a dual attention function that distinguishes between homogeneous and heterogeneous nodes’ correlations. This utilizes content-based attention and key-value attention for measurement, respectively. Expanding on this, we propose a non-invasive propagation method that allows homogeneous and heterogeneous neighbors to influence — but not invasively disrupt — each other. Thirdly, for graph updating, in MMSR, each node adaptively chooses the order of fusion through an update gate. This means each node can decide whether to fuse heterogeneous information first followed by homogeneous information, or vice versa.

We experiment across six diverse scenarios, incorporating both image and text modalities as feature sets. In ablation experiments, we found that the optimal order for modality fusion — whether early or late — varies per dataset. Our proposed method, which adaptively determines the fusion order for each node, strikes balance, consistently enhancing the efficacy of fusion. Our MMSR outperforms the state-of-the-art baselines by 8.6% in terms of HR@5 on average, while also exhibiting strong robustness to missing modalities in real-world scenarios. We show that this is because MMSR enables items to search for matching visual or linguistic features, even in the absence of certain text or image nodes, rather than simply replacing missing modalities with default values. MMSR can be scaled beyond two modalities, and thus is practical for diverse real-world multi-modal scenarios.

We summarise our contributions as follows: (i) We spotlight challenges in modality fusion for sequential recommendation, and propose a versatile solution — our MMSR framework. It accommodates both early and late fusion across modalities. (ii) We offer a graph-centric holistic fusion method as the engine in MMSR, enabling the adaptive selection of fusion order for each feature node. (iii) We conduct comprehensive experiments on six datasets, which show significant gains in both accuracy and robustness.

2 RELATED WORK

2.1 Multi-modal Recommendation

Multimodal recommendation systems leverage features from various modalities, including textual content [11, 21, 32, 38, 47, 55] and images [6, 7, 31, 33, 34, 44], to enhance item representations. Image feature extraction encompasses signal detectors for color [52] and texture [5]; local feature extractors for detecting objects in coherent regions [5, 12]; and latent feature encoders using pre-trained CNN models [10, 26]. Textual features range from concept-level features obtained via tools like NER [38] and TextRank [21], to semantic features from encoders like CNN [11, 55] or BERT [32, 47].

To obtain the final hidden representation, the fusion [56] of modality features can occur either before [15, 29] or after [43, 46] being sent into the feature interaction module. Approaches such as MGAT [43] directly sum the features to disentangle personal interests by modality and aggregate them into the final item representation. MMGCN [46] merges modality-specific graphs through
concatenation, but may not fully capture intermodal relations. In contrast, EgoGCN [3] introduces Ego fusion, extending information propagation beyond the unimodal graph to capture relationships between modalities. It aggregates informative intermodal messages from neighboring nodes, generating final representations by combining multimodal and ID embedding propagation results via concatenation. Despite these advancements, current multi-modal recommendation research predominantly targets collaborative tasks, still leaving the use of multi-modality in sequential recommendation largely unexplored.

2.2 Feature Fusion in Sequential Recommenders

Sequential recommenders (such as GRU- [18], Transformer- [22], or BERT-based [41] models) capture user interests using item ID sequences. To incorporate additional item features (primarily attribute features), fusion methods are used to integrate them into the overall item representation. These fusion methods can be categorized as late, early, or holistic fusion, depending on when feature representations are merged.

In late fusion, sequential relationships within each feature channel are modeled before merging them in a final stage. For example, FDSA [54] separately encodes item and side features using self-attention before fusion. Conversely, early fusion integrates feature representations prior to exploring sequential interactions. Early fusion can be invasive or non-invasive. Invasive methods irreversibly merge item IDs with side features through techniques like concatenation [39, 42], addition [19, 41], or gating [25]. As an example, DETAIN [27] uses a 2D approach to handle sequential items’ features, merging feature channels with vertical attention and items with horizontal attention. However, these methods alter the original representations and have documented drawbacks in terms of compound embedding space [28]. Non-invasive approaches do not directly mix item representation with features. For example, NOVA [28] fuses features while maintaining consistency in item representation. DIF-SR [49] introduces an attribute-based attention function for fusing items. In contrast, Holistic fusion posits that modality fusion and sequential modeling can proceed without rigid ordering. Trans2D [40] employs 4D attention matrices to gauge item attribute correlations but overlooks the ordering of heterogeneous and homogeneous relations. Our work introduces an adaptive method that determines relation application order per node during propagation, providing a more versatile solution.

3 PRELIMINARIES

In our problem, the core task is sequential recommendation: Given a user’s historical interaction data $H_u$, the aim is to find a function $f : H_u \rightarrow v$ that predicts the next item $v$ that the user is most likely to consume. In a typical sequential recommendation task, the historical interaction data includes only item ID information; i.e., $H_u = \{v_1, v_2, \ldots, v_m\}$. Based on this foundation, modality-enhanced sequential recommendation considers the modality of items in the sequence as well, represented by $H_u = \{s_1, s_2, \ldots, s_m\}$, where each $x$ is the combination of different feature channels of the item (including item identifier and item modalities). In this work, we only consider image and text modalities (although extensible to other modalities), and one instance is represented as $s_i : \{a_i, b_i\}$. Here, $a_i$ and $b_i$ indicate the image and text of item $s_i$, respectively. To simplify our discussion, we will refer item ID, image feature, and text feature as three feature channels of modalities; i.e., $v \in V$, $a \in A$, and $b \in B$.

3.1 Base Model

We now discuss the base sequential recommendation model, which we characterize as a 3-tuple of (an Embedding, Representation learning, Prediction).

Initial embedding. The item ID features are represented as integer index values and can be converted into low-dimensional, dense real-value vectors by performing table lookups from an embedding table. For modality embeddings, the commonly-used approach is to directly utilize its extracted features and represent them as a feature vector, through a third-party model [13, 35]. In order to obtain a comprehensive embedding tensor $E \in \mathbb{R}^{3 \times m \times d}$ of the input features of user history, the feature channels are organized in columns and the sequences are organized in rows.

$$ E = \begin{bmatrix} e_{v_1}, & e_{v_2}, & \cdots, & e_{v_m} \\ e_{a_1}, & e_{a_2}, & \cdots, & e_{a_m} \\ e_{b_1}, & e_{b_2}, & \cdots, & e_{b_m} \end{bmatrix} $$

(1)

Representation learning. Numerous existing works have concentrated on designing network architectures for the purpose of modeling feature interactions, outputting the user representation $P$. This can be expressed as:

$$ P = f(E) $$

(2)

For early fusion, the vertical feature channels are fused first, followed by the fusion of the horizontal sequence relationships. For simplicity, we use a linear combination for fusion through channels.

$$ E_{i,:} = \sigma(W(cat[E_{i,1}; E_{i,2}; E_{i,3}])) $$

(3)

$$ P = M([E_{1,:), E_{2,:), \ldots, E_{m,:}]) $$

(4)

For late fusion, the order is reversed and can be formulated as:

$$ E_{j,:} = M([E_{1,j}; E_{2,j}; \ldots, E_{m,j}]) $$

(5)

$$ P = \sigma(W(cat[E_{1,:}; E_{2,:}; E_{3,:}])) $$

(6)

where $cat[\cdot]$ is the concatenation operation, $W$ is the linear weight parameter, $\sigma$ is the activation function, and $M$ is the models for sequence modeling. In contrast, for holistic fusion, the $f$ will process $E$ as a whole. Trans2D [40] directly applies 2D-attention on $E$, and our method considers $E$ as node representations in a graph structure.

Prediction. By scoring candidates items $e_v$ against the learned user representation $P$ using a dot product, we generate the predicted probability scores:

$$ \hat{y} = \langle P, e_v^T \rangle $$

(7)

During training, the model measures and minimizes the differences between the ground-truth $y$ and the prediction $\hat{y}$ through cross-entropy loss [22].

4 APPROACH

As stated earlier, the fusion order during the representation learning stage is crucial. Current methods fail to balance the extremes of the two orders. To address this, we propose the MMSR framework,
which extends the base model and incorporates a graph-based fusion neural network in the representation learning stage to fuse features. After constructing Multi-modal Sequence Graphs for each user, we utilize a dual attention mechanism to independently aggregate heterogeneous and homogeneous node information, enabling an adaptive merging order that facilitates simultaneous consideration of both sequential and cross-modal aspects.

4.1 Multimodal Sequence Graph Construction

For each user $u$, we represent his/her history as a graph — a Modality-enriched Sequence Graph (MSGraph). $G_u = (N_u, \mathcal{R}, E_u)$. Note that each user’s graph $N_u$ and $E_u$ can differ. For simplicity, we’ll refer to a single user’s graph, and just represent them as $N$ and $E$, in the discussion that follows. Figure 3 depicts the construction pipeline. The right side illustrates node construction from modalities, while the left details the edge construction within the MSGraph.

**Nodes and their initialization.** Each MSGraph should consist of $m \times 3$ nodes (where $m$ is the sequence length), forming the node set $N$. $N$ encompasses the three types of nodes, representing three distinct features of channels: $\{e_1, ..., e_m\}$, $\{a_1, ..., a_m\}$, and $\{b_1, ..., b_m\}$. Their representations are associated with the first row (item ID feature), second row (image feature), and third row (text feature) of matrix representation tensor $E$, respectively.

During node representation initialization, $e_0$ is randomly initialized. For $e_0$ and $e_b$, we extract semantic features from the corresponding modality. Our method is not limited to image and text modalities, and for better extension ability, we use separate models instead of large visio-linguistic models for feature extraction. Visual features $e_0$ are obtained from a ResNet-50 [13] model pre-trained on ImageNet [8], while textual features $e_b$ are extracted using a pretrained T-5 model [35]. This scheme can be represented as “modality $a \Rightarrow$ representation $e_a$” (the same applies to $b$).

**Node transformation and compositions.** According to Hou et al. [20], closely binding text encodings with item representations can be detrimental. Thus, instead of using each modality as an individual node, we introduce “modality codes” [20, 36] as alternative nodes. These nodes correspond to discrete indices obtained by mapping the original modality features. This approach helps alleviate the tight binding between item modality and item representations. The node representations utilize these indices to look up the code embedding table, resulting in a scheme denoted as “modality $a \Rightarrow$ code ID$A \Rightarrow$ representation $e_a \approx e_{ID_A}$”. To achieve this, we use a linear autoencoder [2] to condense image/text feature vectors. We then use a K-means [30] to cluster the modality feature vectors by modality type. The indices of cluster centers are used as modality codes $ID_A$. Initialized representations $e_{ID_A}$ are derived from these cluster center representations.

We go beyond treating each item modality as an independent index, employing a composition technique. It enables mapping of multiple modalities to a single code, and a single modality to multiple codes. For example, both $a_1$ and $a_2$ share a common modality node in the graph, and $a_1$ can correspond to multiple codes represented by $ID_{A_1}$, as a set of codes. By doing so, we significantly enhance the connectivity of features within each MSGraph. To achieve this, we cluster each channel of modality into $c$ clusters and select the top $k$ nearest centers as the corresponding code set for each individual modality. The selection process is based on cosine similarity between the modality feature vectors and cluster center vectors. Here, $k$ represents a hyperparameter that determines the number of codes each modality is connected to. For brevity, we will refer to the modality codes as modalities.

**Edges and Relation Types.** In the MSGraphs, we specify the edges as relations $E$ between nodes, including homogeneous relations $E_{homo}$ and heterogeneous relations $E_{hete}$. Both can be formulated as $E = (n_x, r, n_y)$, indicating the relation $r$ between subject node $n_x$ and object node $n_y$ (where both $n \in N$). In $E_{homo}$, $n_x$ and $n_y$ should be in the same type, such as encompassing items (a, r, a) or modalities (a, r, a). And the $r \in \mathcal{R}$ encompases 3 types of sequential relations (intra-item or intra-modality): transition-in, transition-out, and bi-directional transitions. The term “transition” refers to the direct adjacent relationship in a sequence. For instance, if item A is selected immediately before item B, A to B is a transition-out relation, while B to A is a transition-in. For modalities, we also establish direct connections between adjacent nodes in the sequence order. In case there is a back-and-forth relationship between the two modalities, we label it as a bi-directional relation. In $E_{hete}$, $n_x$ and $n_y$ belong to different node types, such as (a, r, b) or (a, r, a). There just exists one type of relation $r$, which signifies the correspondence matching between different feature channels of the same item. Additionally, in both types of relations, we introduce self-loop relations for each node to preserve its original information.

4.2 Node Representation

In MSGraph, each node is assigned an independent representation. However, graphs pose a challenge when modeling sequential tasks as they undermine the inherent sequential nature [4]. This issue is evident when graphs fail to reconstruct sequences due to repeated nodes, particularly as modality codes intensify this repetition. Additionally, the impact of different node types on user preferences within a sequence may vary. For example, images may have a more pronounced short-term influence on user preferences than text.

We propose a solution by integrating positional embeddings and node type embeddings into the original initialized representation $e_n$ for each node. These embeddings map integer indices to low-dimensional dense vectors using separate embedding tables. Specifically, for position embedding of node $n$, its node type is embedded, yielding vector $e_n^p$. Furthermore, the node’s positions within the sequence are captured by a set of position indices, as modality nodes would take multiple positions. Each position index corresponds to an individual embedding, and the position embedding $e_n^{po}$ is obtained by averaging these embeddings. This average vector indicates the position bias of the node towards the beginning or end of the sequence. Finally, the node representation is combined as $e_n = W[e_n, e_n^p, e_n^{po}]$, where $W$ is the weight parameter used for merging the concatenated embeddings.

4.3 Representation Propagation Layers

Given user graph $G_u$, the next step involves aggregating the neighbor information for each node. This process can also be interpreted as modal fusion, where the sequential order and interdependencies between modalities are simultaneously taken into account.
4.3.1 Synchronous Graph Neural Networks. The most intuitive idea is to use graph neural networks to synchronously fuse the node information together [16, 37, 43, 46]. Here, synchronously refers to all nodes being updated simultaneously from the previous layer to the next layer, without any specific order. Here, we denote the central node as $n_i$ and its corresponding neighbor set in the graph as $N_i$. The aggregator updates the representation of each node iteratively from the previous layer $h_i^{(l)}$ to the next layer $h_i^{(l+1)}$. Here, $h_i^{(0)}$ is initialized by $e_i$. We give examples of the following state-of-the-art graph aggregators as potential candidates to facilitate synchronous information propagation.

- **GCN Aggregator** [24] takes into account the neighborhood information of a central node and aggregates it using a convolution operation. Its formulation is represented as follows:

$$h_i^{(l+1)} = \sigma \left( \sum_{j \in N_i} d(i, j) W^{(l)} h_j^{(l)} \right)$$

where $\sigma$ and $W^{(l)}$ are the activation function and the transformation matrix of layer $l$, $d(i, j) = 1/\sqrt{|N_i||N_j|}$ is the normalization factor. We give an illustration in Figure 2 (upper right).

- **GAT Aggregator** [45] further considers that each neighbor has a different impact on the central node, incorporating the attention mechanism to assign varying weights to neighbors:

$$h_i^{(l+1)} = \sum_{j \in N_i} a_{ij}^{(l)} h_j^{(l)}$$

where $a_{ij}^{(l)}$ represents the attention score between node $i$ and node $j$. It is calculated by applying softmax to the dot product of a learnable weight vector $a$ and the concatenated representations of nodes $i$ and $j$ after linear transformations $W^{(l)}$:

$$e_{ij}^{(l)} = a^T [W^{(l)} h_i^{(l)}; W^{(l)} h_j^{(l)}]$$

$$a_{ij}^{(l)} = \text{softmax}(e_{ij}^{(l)} | N_i)$$

where, $a$ is the parameter for calculating the attention score $e$. For simplicity, we denote the softmax operation as $\text{softmax}(e_{ij}^{(l)} | N_i)$.

As aggregators are very important for our method’s performance, acting as the modality fusion module, we study the effectiveness of the above aggregators as well as other aggregators in the experiment section (§ 5.2).

4.3.2 Our Graph Neural Network. There are some drawbacks to using the above graph neural networks: Firstly, concerning the 3 types of nodes and 5 types of relations, the heterogeneity of both should be taken into account. Secondly, as stated earlier, the order of fusion matters, thus synchronous updating is not optimal. Some prior modal information may be more beneficial to the corresponding item representation, thus should be merged first. Thirdly, the representations of the item, image, and text nodes are not in the same space, and thus are inappropriate to fuse them directly. The inclusion of different modalities can interfere with each other’s representations, resulting in invasive fusion problems [28]. This issue also persists during aggregation. Based on these considerations, we propose an Heterogeneity-aware, Asynchronous, and Non-invasive graph neural network (or HAN-GNN for short).

**Heterogeneity-aware.** To aggregate homogeneous and heterogeneous neighbor nodes, we employ a divide-and-conquer strategy. For homogeneous nodes $n_i \in N_i$, they represent neighbors that are connected to the central node $n_i$ through edges $(n_i, r, n_j) \in E_{\text{homo}}$, with their identical types $t_j = t_i$ (where $t$ belongs to the indices of three channels). For heterogeneous nodes $n_i \in N_i$ with $t_j \neq t_i$, they represent neighbors that differ in type from the central node. These nodes are connected through edges $(n_i, r, n_j) \in E_{\text{hete}}$. Unlike GCN/GAT that transform the original hidden vector $h$ into a value vector using $W$ for layer-wise aggregation, our method establishes $W_Q, W_K,$ and $W_V$ to transform the original vector into query, key, and value vectors, respectively. Considering three distinct node type $t$, three sets of parameters $(W_{Qi}, W_{Kt}, W_{Vt})$ are designated.
For attention regarding homogeneous nodes, their shared space allows direct comparison. We employ content-based attention for this. Formally, for \((n_i, r, n_j) \in \mathcal{E}_{\text{homo}}\), their attention scores are:

\[
e_{ij}^{(l), \text{ho}} = a_r(W_{V,t}^{(l)} h_j^{(l)} \odot W_{V,t}^{(l)} h_i^{(l)})
\]

where \(\odot\) is the element-wise production. \(t_i\) and \(t_j\) indicate the types of \(n_i\) and \(n_j\) respectively, which are identical in this case. It results in \(W_{V,t_i} = W_{V,t_j}\). Concerning types of sequential relations in \(\mathcal{E}_{\text{homo}}\) for each relation \(r\) between \(n_i\) and \(n_j\), we define an individual parameter \(a_r\) for each relation type.

For the attention calculation with respect to heterogeneous nodes, as they are located in different spaces, so we employ type-specific transformation matrices \((t_j \neq t_i)\) to bring them into a common space for comparison. More specifically, we utilize key-value attention to evaluate the correlations between nodes after their individual transformations. For \((n_i, r, n_j) \in \mathcal{E}_{\text{heter}}\), the attention scores are:

\[
e_{ij}^{(l), \text{he}} = (W_{Q,t}^{(l)} h_j^{(l)}) (W_{K,t}^{(l)} h_i^{(l)})^\top
\]

Building on the previously mentioned attention scores, we can independently gather the updated representations that each individually aggregate the two distinct sources of information. These can be succinctly expressed as follows:

\[
h_i^{(l+1), \text{ho}} = \sum_{j \in N_i} \text{softmax}(e_{ij}^{(l), \text{ho}}) W_{V,t_j}^{(l)} h_j^{(l)}
\]

\[
h_i^{(l+1), \text{he}} = \sum_{j \in N_i} \text{softmax}(e_{ij}^{(l), \text{he}}) W_{V,t_j}^{(l)} h_j^{(l)}
\]

The symbol \(\ast\) denotes either ho or he, indicating the individual aggregation of homogeneous or heterogeneous information, respectively. This is represented as homogeneous aggregation and heterogeneous aggregation in Figure 2 (right).

Providing information from both sources of neighbors, a direct approach would be to concatenate them and concurrently update this combined representation in the next layer:

\[
h_i^{(l+1)} = \text{Linear}(h_i^{(l+1), \text{ho}}; h_i^{(l+1), \text{he}})
\]

where \(\text{Linear}\) indicates a linear layer. We refer to this approach as a “synchronous updating” version of our implementation, to be evaluated in our ablation experiments (§ 5.3).

**Asynchronous updating.** Synchronous updating overlooks the effect of the fusion order. Therefore, we propose an asynchronous updating strategy with two defined updating orders. In every layer \(l\), one could either first aggregate homogeneous information for node updates and then use these updated representations for heterogeneous information aggregation, or vice versa. We term these two distinct updating orders as “homogeneous-first, heterogeneous-second” (or hohe) and “heterogeneous-first, homogeneous-second” (heho). Taking hohe as an example, this two-phase paradigm can be denoted as \(h_i^{(l)} \rightarrow h_i^{(l), \text{ho}} \rightarrow h_i^{(l), \text{he}}\). The first phase \(h_i^{(l)} \rightarrow h_i^{(l), \text{ho}}\) uses homogeneous aggregation depicted in Equation 14, but holding the layer number \((l)\) unchanged. The second phase \(h_i^{(l), \text{ho}} \rightarrow h_i^{(l), \text{he}}\) follows the heterogeneous aggregation in the Equation 14 but substitutes the input \(h_i^{(l)}\) with the output \(h_i^{(l), \text{ho}}\) from the first phase. Similarly, the heho updating order follows: \(h_i^{(l)} \rightarrow h_i^{(l), \text{he}} \rightarrow h_i^{(l), \text{ho}}\).

As depicted in Figure 2 (left), in HAN-GNN, each node has two potential aggregation routes from its representation \(h_i^{(l)}\) to \(h_i^{(l+1)}\) via either the ‘heho’ or ‘hohe’ path. We introduce an update gate to adaptively select the optimal path for each node using the following gate selection mechanism:

\[
\beta = \text{MLP}([h_i^{(l+1), \text{hohe}}, h_i^{(l+1), \text{heho}}]),
\]

\[
h_i^{(l+1)} = \beta_0 \times h_i^{(l+1), \text{hohe}} + \beta_1 \times h_i^{(l+1), \text{heho}},
\]

where MLP is a multi-layer perceptron, and \(\beta \in \mathbb{R}^2\) contains the scores for gate selection.

**Non-invasive fusion.** Drawing inspiration from NOVA [28], we employed a non-invasive technique to limit interference among different node types during feature updates. For example, although the image features are fused with the item node in Phase 1, they do not actually update it but only use the updated representation for calculating the attention scores in Phase 2.

Let us take the example of \(h_i^{(l)} \rightarrow h_i^{(l), \text{he}} \rightarrow h_i^{(l), \text{hohe}}\) to make this concept concrete. In Phase 1, the graph aggregates neighbors’ information from the transformed value vector of \(h_i^{(l)}\) while in Phase 2, the aggregation uses the transformed value vector of \(h_i^{(l), \text{he}}\). This implies that the value vector used in the second phase has already undergone a substantial update. Considering the degradation of representations caused by excessive fusion when modeling sequences with heterogeneous information, we introduce a non-invasive approach during graph updating. Specifically, in Phase 2, though we calculate the attention based on the intermediate state \(h_i^{(l), \text{he}}\), we continue to use the value vector of \(h_i^{(l)}\) (instead of \(h_i^{(l), \text{he}}\)) for aggregation. We posit that non-invasive technique also applies to the converse order for updating (i.e., hohe). Taking into consideration that each item is an independent entity, the question remains whether to permit the invasive integration of homogeneous contextual information into the fusion of heterogeneous information. We examine this question in our ablation experiments (§ 5.3).

### 4.4 User Interest Representation and Prediction

Following \(L\) layers of aggregation, we obtain the final \(h_v^{(L)}\) and set the collection of hidden states of item nodes \(\{h_v^{(L)}\}_v \in \mathcal{V}_c\) forms the output \(Z \in \mathbb{R}^{|\mathcal{V}_c| \times d}\), where \(\mathcal{V}_c\) indicate item ID node set. The resulting \(Z\) can be considered a representation that has undergone modal fusion. Hence, the key lies in the mapping function \(P : \mathbb{R}^{|\mathcal{V}_c| \times d} \rightarrow \mathbb{R}^d\) of outputting user representation \(P = \mathcal{P}(Z)\), facilitating the next-item prediction. From our observation, using graphs presents a challenge as it tends to diminish the impact of individual items, making it challenging to differentiate similar sequences. For instance, the graph model may produce similar representations for sequences such as \((v_1, v_2, v_3)\) and \((v_1, v_2, v_3, v_4)\). To address this issue, instead of employing average pooling, we adopt last pooling, where we select the last item from the sequence as the pooled representation. Specifically, we denote it as \(P = Z_{|\mathcal{V}_c|}\).

### 4.5 Model Comparison & Complexity Analysis

When sequential relationships between modalities are strong, the selection gate prioritizes updates among homogeneous nodes first. Conversely, when interdependent relationships are strong, the gate prioritizes updates among heterogeneous nodes. Thus, our framework can set a fusion order that spans from early to late modality.
fused as a spectrum of possibilities, with *hohe* representing late fusion and *heho* early fusion.

For complexity comparison, the fused representation from HAN-GNN (in the representation learning stage) can be used directly for online inference, matching the base model’s time complexity. The main time cost for model training comes from layer-wise graph networks. Compared to GCN’s complexity of $O(L|\mathcal{U}||\mathcal{E}_u|)$ and Graphormer’s [50] $O(L|\mathcal{U}||\mathcal{N}_u|^3)$, HAN-GNN takes $O(2L|\mathcal{U}||\mathcal{E}_u|)$ as there are two phases in each layer propagation. Here, $|\mathcal{U}|$ indicates the number of users, and $|\mathcal{E}_u|$ and $|\mathcal{N}_u|$ indicate the average number of edges and nodes in each user graph, respectively. While our approach is more complex than simpler networks like GCN, it offers lower complexity compared to yet more complex networks, like Graphormer, while still delivering superior performance. In the user graph, each node connects to its preceding and following nodes in the sequence, and at least 2 other modality nodes. With 4 edges per node, $|\mathcal{E}_u| = O(4|\mathcal{N}_u|) = O(|\mathcal{N}_u|)$, where $H_u$ signifies the user interaction sequence length. Therefore, our method is more efficient than Graphormer when the user history exceeds $2 \times \frac{4}{3} \approx 2.66$. In typical cases where the average user history length varies between 7 and 9, our method is considerably more efficient.

5 EXPERIMENT

Datasets. In line with previous studies [15, 53], we utilized the Amazon review dataset [14] for evaluation. This dataset provides both product descriptions and images, with varying sizes across product categories. To showcase our approach’s versatility, we selected six datasets from diverse categories: Beauty, Clothing, Sport, Toys, Kitchen, and Phone. In these datasets, each review rating signifies a positive user–item interaction. Following the standard practice in prior research [15, 16, 53] and to facilitate fair comparison with existing methods, we applied core-5 filtering, which refines the dataset ensuring each user and item has a minimum of five interactions. Dataset details are presented in Table 1.

<table>
<thead>
<tr>
<th>Table 1: Dataset Statistics after preprocessing.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>User #</strong></td>
</tr>
<tr>
<td>---------</td>
</tr>
<tr>
<td>User #</td>
</tr>
<tr>
<td>Item #</td>
</tr>
<tr>
<td>Inter. #</td>
</tr>
<tr>
<td>Avg. Len. #</td>
</tr>
<tr>
<td>Sparsity</td>
</tr>
</tbody>
</table>

Baselines. We compare against three groups of models. (A) Basic SR models include GRU4Rec [18] using Gated Recurrent Units (GRU) to model the sequential dependencies between items; SASRec [51] adopting a self-attention mechanism to capture long-term dependencies more effectively; and SR-GNN [48], a graph-based approach, incorporating both user–item interactions and item–item relationships to capture higher-order dependencies in sequential data. (B) Multi-modal collaborative models include MGAT [43] focusing on disentangling personal interests by modality. It employs a graph attention network to integrate information from different modalities; MIMGCN [46] integrating multimodal features into a graph-based framework. It utilizes a message-passing scheme to learn the representations of users and items; BM3 [57] bootstrapping latent contrastive views of user/item representations, optimizing multimodal objectives for learning. (C) Feature-enriched SR models include NOVA [28] and DIF-SR [49] as state-of-the-art non-invasive fusion methods; Trans2D [40] as holistic fusion methods. We also used modified versions known as GRU4Rec$^F$ (late fusion) and SASRec$^F$ (early fusion), based on the GRU4Rec and SASRec models. These determine the best fusion choice for each, as seen in the intro case study.

5.1 Overall Performance

MSMR consistently outperforms other models (Table 2). It shows a significant improvement in HR (8.6% for Top-5, 2.8% for Top-20) and MRR (17.2% for Top-5, 7.6% for Top-20), on average. Our approach of fusing modal features enhances recommendation precision, ranking preferred items higher.

Comparing the basic sequential recommendation baseline with our baseline that includes modalities as side features, the latter is stronger overall. SASRec stands out among the baseline models, demonstrating the excellent performance of attention in sequential recommendation. In contrast, SR-GNN, the existing graph-based baseline, performs poorly, highlighting the superiority of our method in utilizing the graph. Among the sequence recommendation baselines enhanced with modal features, DIF-SR and SASRec$^F$ perform best, demonstrating that attention effectively enhances early fusion (both invasive and non-invasive). SASRec$^F$ adopts an invasive early fusion approach, directly fusing modal representation into item representation. In contrast, DIF-SR uses a non-invasive approach, where modal features are not fully integrated into the item representation vector. However, contrary to previous findings [28], our analysis shows that the invasive approach can be comparatively effective. This can be attributed to our modality codes (from the autoencoder), which introduce a more generalized modality representation for items, instead of too specific representation.

Existing multi-modal recommendation baselines focusing on inter-modality modeling with collaborative signals (MGCN, MGNN, BM3) do not incorporate sequential relationships, resulting in poor performance. It reveals that, for the SR task, besides inter-modality relationships, considering the intra-modality sequential relationships remains vital. Our proposed method fills this gap and is necessary for improving sequence recommendation tasks.
<table>
<thead>
<tr>
<th>Model</th>
<th>Beauty</th>
<th>Clothing</th>
<th>Sport</th>
</tr>
</thead>
<tbody>
<tr>
<td>GCN</td>
<td>5.6348</td>
<td>3.163</td>
<td>2.134</td>
</tr>
<tr>
<td>GraphSAGE</td>
<td>5.5773</td>
<td>3.128</td>
<td>2.397</td>
</tr>
<tr>
<td>GAT</td>
<td>5.7116</td>
<td>3.194</td>
<td>2.777</td>
</tr>
<tr>
<td>Graphormer</td>
<td>5.9267</td>
<td>3.302</td>
<td>3.777</td>
</tr>
<tr>
<td>RGAT</td>
<td>6.8157</td>
<td>3.978</td>
<td>2.982</td>
</tr>
<tr>
<td>HGNN</td>
<td>6.9701</td>
<td>4.127</td>
<td>3.063</td>
</tr>
<tr>
<td>HGAT</td>
<td>7.0671</td>
<td>4.249</td>
<td>3.048</td>
</tr>
<tr>
<td>HAN-GNN</td>
<td>7.1386</td>
<td>4.624</td>
<td>2.0402</td>
</tr>
</tbody>
</table>

Table 3: Graph Aggregator Comparison.

### 5.3 Ablation Study

To better understand the superiority of our approach, we conducted an ablation study on HAN-GNN. In Table 4, ho and he signify HAN-GNN propagation solely through homogeneous or heterogeneous relations, respectively. hohe signifies the use of Homo–Hetero Ordering Fusion, while heho represents Hetero–Homo Ordering Fusion. "NI" signifies the non-invasive fusion ordering for each of them. "Synchronous" refers to Equation 15, which simply concatenates and linearly transforms homogeneous and heterogeneous information.

Examine the fusion of ho and he only, we found that the Sport dataset performs better when considering homogeneous relationships, while the Beauty and Clothing datasets benefit more from considering only heterogeneous information. This suggests that users in the latter scenarios rely more on either visual or textual information for ordering decisions, while this is not the case in the Sport dataset. Regarding fusion order, for invasive fusion, fusing heterogeneous information for ordering decisions, while this is not the case in the Sport dataset. This suggests that the heterogeneous information from item node information. Thus, our non-invasive approach is more effective in handling heterogeneous information.

### 5.2 Graph Aggregator Study

In our paper, we designed a graph neural network specifically for integrating multi-modal features. To demonstrate its superiority over other graph neural networks, we compared it against several popular models, including GCN, GraphSAGE, GAT, and Graphormer, which do not consider heterogeneity; as well as RGAT, which considers heterogeneity in edge types; and HGNN and HGAT, which consider heterogeneity in modality-enriched graphs. In MSGraph, incorporating heterogeneity in modality-enriched graphs leads to significant performance improvements compared to models that do not consider heterogeneity.

Further comparing HGNN and RGAT, we find that the heterogeneity in modality-enriched graphs leads to significant performance improvements compared to models that do not consider heterogeneity. Further comparing HGNN and RGAT, we find that the heterogeneity of nodes is more important, particularly in distinguishing modality information from item node information. Thus, our non-invasive approach is more effective in handling heterogeneous information.
Table 4: Ablation analysis, evaluated with (HR, MRR)@5. The relation ablation is based on a GCN aggregator.

Table 5: Robustness to missing modalities.

Figure 4: MMSR robustness against missing modalities.

Figure 5: The performance comparison with different MS-Graph construction parameters on the Beauty dataset.

5.5 Modality-enriched Graph Construction

Constructing a graph from a user’s historical sequence can be challenging, as having too many modality nodes can result in an overly sparse graph. We thus compare different settings within our graph construction method (using a modality code set and soft links between original modalities and modality codes to improve the graph density). The x-axis represents the cluster number (i.e., the number of modality codes), while the y-axis represents the number of codes corresponding to an original modality (i.e., k).

We see that using modality codes achieves better performance over not using modality codes (compare HR@5: 7.4263, MRR@5: 4.7469 to results in Figure 5). Secondly, we observed that a larger value of c does not necessarily lead to better performance, as the optimal point is typically between 20 and 30. As c increases, the optimal value of k increases accordingly. Finally, the utilization of modality codes is consistent with the findings of previous studies [20, 36], which demonstrated their positive impact on performance.

6 CONCLUSION AND FUTURE WORK

We introduce a Multi-Modality enriched Sequential Recommendation framework while optimally fuses modality features in sequential recommendation. Our approach tackles the complexity of fusing multi-modalities in sequential tasks, where fusion order notably influences the recommendation model performance. To drive MMSR, we develop a novel graph aggregation mechanism (HAN-GNN) that employs a dual graph attention network and asynchronous updating strategy. HAN-GNN flexibly integrates modality information while preserving sequential relationships. MMSR consistently outperforms state-of-the-art baselines, even under challenging missing-modality scenarios. This makes it a flexible and robust solution for real-world applications.

MMSR is easily extensible, allowing for expansion to additional modalities. We are optimistic about its utility in industrial contexts. Furthermore, exploring the interpretability of complex modal relationships in modality-enriched SR opens up new horizons for future research. Unraveling how and when sequentiality or independent relationships become pivotal could lead to more nuanced and efficient recommendation.

ACKNOWLEDGMENTS

We thank the new deep learning computing framework MindSpore [1] for the partial support of this work.


