Contrastive Graph Recommendation with Sequence-Item Views and Modality-Guided Fusion

Anonymous Author(s)

Abstract

In recommender systems, cold-start and sparsity are often addressed by incorporating sequential, multimodal, and contrastive signals. However, combining these sources of information can also introduce noise and weaken useful semantics, particularly when augmentations are imposed artificially or when multimodal features are not well aligned. We propose MuSICRec, a multi-view graph recommender that integrates collaborative, sequential, and multimodal information within a unified contrastive framework. Central to MuSICRec is a sequence-item (SI) view, where sequence nodes are formed by attention pooling over the items in each user history. Propagation over this SI graph provides an additional view naturally, avoiding the need for hand-crafted graph perturbations while simultaneously injecting sequential context into representation learning. To further reduce modality noise and better coordinate multimodal information, the influence of text and visual features is controlled through an ID-guided gating mechanism.

We evaluate MuSICRec under a strict leave-two-out protocol against a broad range of sequential, multimodal, and contrastive baselines. Across the Amazon Baby, Sports, and Elec datasets, MuSICRec consistently outperforms state-of-the-art methods from all three model families. The gains are particularly pronounced for users with short interaction histories, indicating that the model is especially effective at alleviating the sparsity and cold-start problems that commonly arise in recommendation. An anonymous implementation is available at https://anonymous.4open.science/r/MuSICRec-3CEE and will be released publicly.

CCS Concepts

• **Information systems** → **Recommender systems**; **Multimedia information systems**; • **Computing methodologies** → *Neural networks.*

Keywords

Sequence-Item View, Multimodal, Sequential, Recommender System, GNN

ACM Reference Format:

Anonymous Author(s). 2026. Contrastive Graph Recommendation with Sequence-Item Views and Modality-Guided Fusion. In *Proceedings of Proceedings of the ACM Conference on Recommender Systems 2026 (RecSys'26).* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

Recommendation datasets are typically very sparse, necessitating methods that enable sharing of training signal between samples, such as collaborative filtering over user-item graphs, contrastive learning via alternative augmented views and extracting multimodal item-item relationships. Sequential recommendation is particularly valuable at capturing temporal user behaviour, but suffers more so from the same data sparsity issues. Figure 1 presents a simplified illustration of collaborative filtering, sequential recommendation methods and how MuSICRec links the two approaches. Existing graph-based recommender systems suffer from two limitations. **Firstly**, where contrastive learning (CL) is utilised to address label sparsity with self-supervised learning, CL approaches synthesise alternative views via heuristic perturbation of the user-item graph (e.g., node/edge dropout, random walks, clustering) [17]. Such augmentations may distort semantics and drop useful information, which could exacerbate the sparsity issue of inactive users. The success of such contrastive learning schemes is highly dependent on the view generator, affecting these models' generality. LightGCL [1] argues this directly and utilises SVD to generate an alternative user-item view to preserve important user-item interaction semantics. Sequential CL models [14] similarly carry out sequence augmentations, which could lead to the same issues. **Secondly**, multimodal fusion is often not calibrated. While visual and textual data provide useful cues, there may be misalignment between the

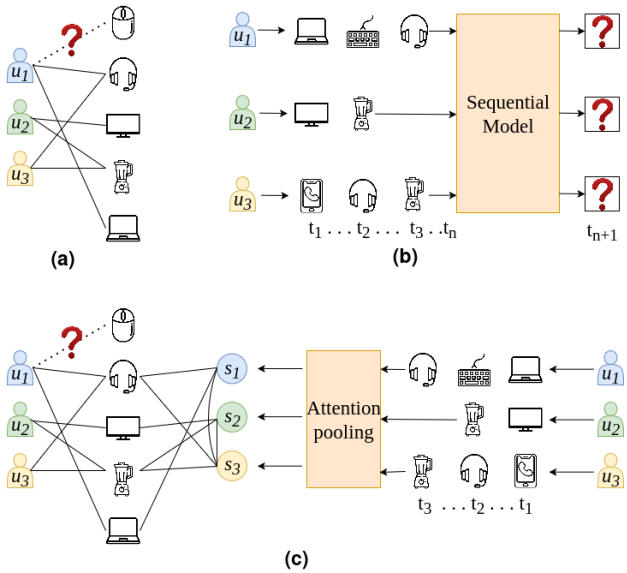


Figure 1: Simplified diagram illustrating the difference between (a) collaborative filtering, (b) sequential methods and (c) MuSICRec. Collaborative filtering predicts user's preference for an item based on other users with similar preferences. Sequential recommenders predict next item a user would be interested in given their history. MuSICRec extracts sequence representations based on user history and treats them as nodes/entities to complement collaborative filtering.

RecSys'26, Minneapolis, Minnesota, USA
2026. https://doi.org/10.1145/nnnnnnn.nnnnnnn

modalities and with the preference/behaviour signal. This could lead to preference-irrelevant noise propagating into learned user and item representations. Recent analyses explicitly report such modality noise diffusion and amplification[3, 22, 25]. Meanwhile, work that unifies sequential dynamics with multimodal information remains limited. Classic multimodal models and their denoising successors largely ignore sequences-as-structure, while sequential ones typically omit modalities.

Beyond treating sequences as lists to be encoded, we treat every user's interaction history as a new class of node within a sequence-item (SI) graph, introducing sequence-to-sequence and sequence-to-item connections. This choice is motivated by the evidence from sequential models showing sequential signals are highly informative for preference modelling [2, 8, 10, 13]. Next-item prediction is greatly boosted by focusing on the most relevant past items through self-attention and bidirectional encoders (e.g., SASRec[8], BERT4Rec[13]). Treating sequences as nodes can expose structural patterns hidden by flat encoders. By passing attention-pooled sequence embeddings into the SI graph and propagating along sequence-item edges, MuSICRec derives an alternative view organically from the data's contextual behavioural topology rather than via synthetic data augmentation. This enables clean entity-level contrast (user u <-> sequence s_u) as the SI view complements the user-item (UI) view. At the same time, the SI graph captures a different signal from the UI bipartite graph. The UI graph encodes a global collaborative signal from user-item interactions, while an SI graph exposes intra-sequence structure (short-range and higher-order item transitions, co-occurrence order, and sequence/context signals) by treating each sequence as a node and connecting it to its constituent items and other sequences. This is akin to what sequential models exploit, but here it is made structural (sequence-as-nodes), so message passing can reveal patterns that UI alone cannot. Practically, forming sequence embeddings via additive attention provides a stable, length-aware representation that emphasises salient items before message passing, while SI propagation exposes the inter-sequence co-ocurrence structure. This proves to be especially beneficial for users with short histories, effectively mitigating sparsity issues, as will later be empirically demonstrated.

Building on the insight that modality graphs can help with training sparsity, yet also inject noise if fused naively, MuSICRec anchors fusion in the ID (collaborative) space before any propagation. Instead of learning full modality graphs, we adopt a static multimodal item-item (II) graph, following FREEDOM [25] (which showed that freezing the multimodal II graph can lead to higher stability and performance) and seed each item with its ID embedding plus an ID-conditioned gate that decides the proportion of visual and textual signal to inject. This mitigates cross-modality misalignment, respects dataset-specific salience of different modalities and limits modality noise from dominating collaborative signals, while letting multimodal content help where it is preference relevant, avoiding the coupling issues in MMGCN [16] and the structure-learning overhead in LATTICE [22]. The MuSICRec model seeds the item state with the ID vector plus an ID-gated mixture of normalised visual/text projections, it then performs LightGCN-style propagation on the frozen II graph [5].

In summary, this paper makes four main contributions:

- Because collaborative filtering methods miss sequential/context signals, we introduce sequences as nodes, built by attention pooling over user history, within a SI graph.
- As artificially augmented views could disrupt behaviour semantics by dropping useful information, we propose using the SI view as an organic alternative view and carry out entity-level cross-view contrast between each user and their own sequence.
- ID-guided cross-modal calibration is proposed to mitigate cross-modal misalignment, by gating the proportion of text and image features that are injected before propagation over the item-item graph.
- We propose a unified benchmarking procedure and perform an extremely broad evaluation, with baselines covering collaborative, sequential, multimodal and contrastive learning recommenders. We report comparative results on several Amazon datasets, as well as ablations, sensitivity to key hyperparameters, and an evaluation of different history lengths.

2 Background

2.1 Multimodal Recommender Systems

Multimodal recommender systems typically adopt graph-based or other deep learning methods to enrich collaborative filtering with visual and textual information. Earlier works, like VBPR [4], expand on classical BPR [11] by concatenating image embeddings to item-ID embeddings, while later models leverage attention mechanisms to discern user-specific modality preferences.

The ability of GNNs to capture higher-order relationships led to the introduction of several key approaches. MMGCN [16] builds user-item graphs per modality, and the learned representations from each graph are aggregated to produce the final user and item embeddings. GRCN [19] improves on this by using a graph refining layer to down-weight noisy edges prior to message passing and aggregation. DualGNN [15] takes a slightly different approach, building user-user co-occurrence graphs from modality-specific graphs via spectral clustering. This implicitly captures users' shifting preferences for different modalities and item-item relations.

Mixing modality features into the user-item behaviour encoding process can lead to the amplification of modality noise and can harm the robustness of learned user and item embeddings. To this end, LATTICE [22] and FREEDOM [25] construct item-item graphs and separate modality propagation and aggregation from the main user-item graph. LATTICE [22] builds item-item graphs per modality and jointly learns item embeddings and adjacency matrices. FREEDOM [25] observed that freezing the LATTICE item-item graph leads to gains and is more lightweight, and therefore uses fixed item-item graphs, precomputed from raw multimodal features. Thus, GCN propagation depends only on structure. It also includes BPR losses to align user and modality embeddings and carries out degree-sensitive edge pruning on the user-item graph to denoise it. LGMRec [3] similarly separates collaborative and modality signals, utilising a Local Graph Embedding module to learn separate interaction and multimodal user/item representations. It complements that with a Global Hypergraph Embedding module to capture global dependencies, also tackling sparsity.

Earlier non-multimodal contrastive learning models construct alternative views of the user-item graph with stochastic graph augmentations. SGL [17] uses edge/node dropout and random walks as a self-supervised signal learning. Later work demonstrates the unreliability of random perturbations with SimGCL [20] dropping graph augmentation in favour of random noise injection into the embeddings to form the contrastive view. LightGCL [1] similarly avoids heuristic graph perturbations, generating views instead with SVD spectral refinement, which simultaneously mitigates data sparsity and popularity bias as aligning with the global low-rank SVD view lowers reliance on high-degree (popular) nodes. BM3 [26] introduces contrastive learning to multimodal recommender systems via simple dropouts and three contrastive losses to optimise representations.

2.2 Sequential Recommender Systems

Sequential methods optimise next-item prediction based on the user interaction history. With the popularisation of Deep learning, RNNs (particularly gated recurrent units) became popular in GRU4Rec [6], but these were limited by sequential processing bottlenecks and vanishing gradient issues. Following the success of transformers in other fields through parallel processing and capturing long-range dependencies, SASRec [8] exploited unidirectional self-attention to attend to the most relevant items in a user's history, while BERT4Rec [13] implemented a bidirectional Cloze objective to recover masked items from both left and right context.

Transformers face two limitations. Firstly, they lack strong inductive biases and therefore require more data to generalise well. The second is the 'over-smoothing' issue, with self-attention acting like a low-pass filter that smoothes out high-frequency details in user behaviour [12]. To counter these issues, recent methods utilise inductive bias or frequency-aware components. FEARec [2] performs self-attention on time and frequency domains with contrastive regularisation to capture low and high frequency signals. BSARec [12] introduces a Fourier-based inductive bias that rebalances low and high frequencies to mitigate oversmoothing. On the other hand, FMLPRec [24] reframes the problem as learnable signal filtering and removes self-attention, replacing it with stacked filter-enhanced blocks.

3 Methodology

3.1 Problem Formulation and Notation

This section will formalise the problem setup and the overall architecture of the proposed MuSICRec framework, as illustrated in Figure 2.

Entities and data. $\mathcal{U} = \{1, \dots, M\}$ and $\mathcal{I} = \{1, \dots, N\}$ denotes the set of users and items, respectively. A binary interaction matrix is defined, $\mathbf{R} \in \{0,1\}^{M \times N}$, such that $R_{ui} = 1$ indicates that user $u \in \mathcal{U}$ interacted with item $i \in \mathcal{I}$ (implicit feedback). A time-ordered interaction history is observed for every user u:

$$\mathbf{s}_u = \left[(i_{u,1}, t_{u,1}), \dots, (i_{u,T_u}, t_{u,T_u})\right], \quad i_{u,k} \in \mathcal{I},\ t_{u,1} < \cdots < t_{u,T_u}, \tag{1}$$

where T_u is the sequence length.

Base embeddings. We store trainable ID-embedding tables for users and items, $\mathbf{U} \in \mathbb{R}^{M \times d}$ and $\mathbf{I} \in \mathbb{R}^{N \times d}$, and denote by $\mathbf{e}^u \in \mathbb{R}^d$ and $\mathbf{e}^i \in \mathbb{R}^d$ the u-th and i-th rows of $\mathbf{U}$ and $\mathbf{I}$, respectively, and by d the embedding dimension.

A set of sequence nodes $\mathcal{S}$ is introduced, where each user is associated with one sequence node $s_u \in \mathcal{S}$. Similar to user and item embeddings, we maintain a trainable sequence embeddings table:

$$\mathbf{S} \in \mathbb{R}^{|\mathcal{S}| \times d}. \tag{2}$$

Multimodal item content. As well as its learned embedding, each item i also has pre-extracted visual and textual features, $\mathbf{f}_i^{(v)} \in \mathbb{R}^{d_v}$ and $\mathbf{f}_i^{(t)} \in \mathbb{R}^{d_t}$. Linear projection matrices $\mathbf{W}_v \in \mathbb{R}^{d_v \times d}$ and $\mathbf{W}_t \in \mathbb{R}^{d_t \times d}$ map these features into the shared space.

Graphs. MuSICRec is designed to exploit collaborative filtering to improve sample efficiency. To that end, three graph structures are constructed to enable information sharing between the above entities. (i) a user–item (UI) bipartite graph with adjacency $\mathbf{A}_{\text{ui}}$, (ii) a sequence–item (SI) bipartite graph with adjacency $\mathbf{A}_{\text{si}}$ and (iii) a multimodal item–item (MM) graph with adjacency $\mathbf{A}_{\text{mm}}$. The exact construction of these adjacency matrices will be discussed in more detail in the following sections.

3.2 User-Item Graph

The simplest graph in MuSICRec is designed to allow information sharing between users who interacted with similar items, and items which were selected by similar users. It is formed directly from the interaction matrix $\mathbf{R}$ as a bipartite graph over the users and items.

$$\hat{\mathbf{A}}_{\text{ui}} = \begin{bmatrix} \mathbf{0} & \mathbf{R} \\ \mathbf{R}^\top & \mathbf{0} \end{bmatrix}, \qquad \mathbf{A}_{\text{ui}} = \mathbf{D}^{-\frac{1}{2}} \hat{\mathbf{A}}_{\text{ui}} \mathbf{D}^{-\frac{1}{2}}, \tag{3}$$

where $\mathbf{D}$ is the diagonal degree matrix of $\hat{\mathbf{A}}_{\text{ui}}$.

To allow longer range connections, information is repeatedly propagated through the graph via L_{ui} sequential layers using simple neighbourhood averaging

$$\mathbf{Z}^{(l+1)} = \mathbf{A}_{\text{ui}}\, \mathbf{Z}^{(l)}, \qquad l = 0, \dots, L_{\text{ui}} - 1. \tag{4}$$

As in LightGCN [5], no transforms, biases or nonlinearities are applied between layers.

For the first layer input, the user and item embedding are stacked

$$\mathbf{Z}^{(0)} = \begin{bmatrix} \mathbf{U} \\ \mathbf{I} \end{bmatrix} \in \mathbb{R}^{(M+N) \times d}, \tag{5}$$

while the final L_{ui}+1 layer outputs are aggregated by a uniform mean:

$$\bar{\mathbf{Z}} = \frac{1}{L_{\text{ui}} + 1} \sum_{l=0}^{L_{\text{ui}}} \mathbf{Z}^{(l)}, \tag{6}$$

before finally splitting back out into user and item blocks,

$$\bar{\mathbf{Z}} = \begin{bmatrix} \mathbf{U}_{\mathbf{ui}} \\ \mathbf{I}_{\mathbf{ui}} \end{bmatrix}, \qquad \mathbf{U}_{\mathbf{ui}} \in \mathbb{R}^{M \times d},\ \mathbf{I}_{\mathbf{ui}} \in \mathbb{R}^{N \times d}. \tag{7}$$

We denote the embeddings produced by the UI-graph as $\mathbf{e}_{\text{ui}}^u$ (the u-th row of $\mathbf{U}_{\mathbf{ui}}$) and $\mathbf{e}_{\text{ui}}^i$ (the i-th row of $\mathbf{I}_{\mathbf{ui}}$).

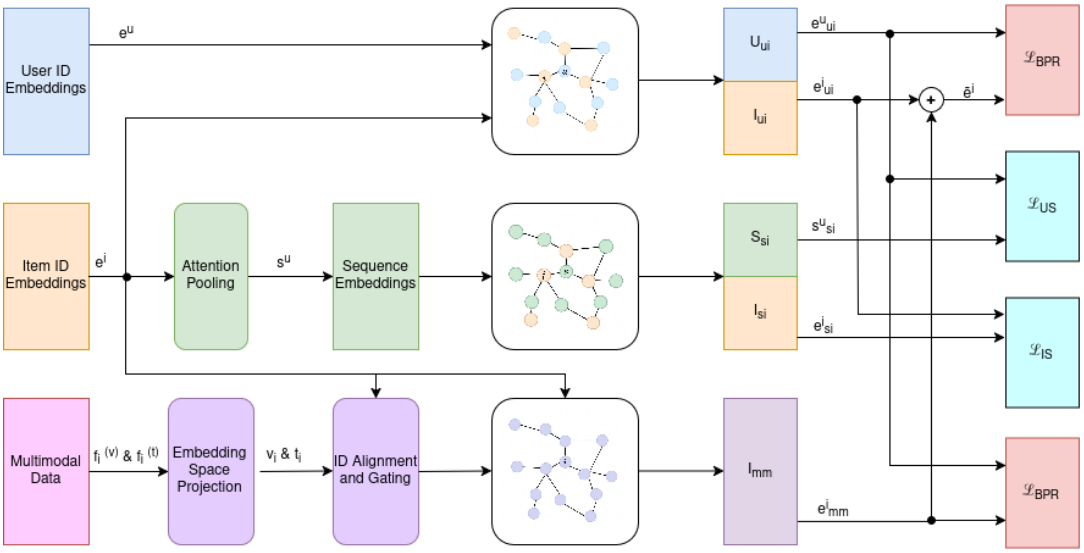


Figure 2: MuSICRec Architecture

3.3 Sequence-Item Graph

To allow us to consider sequential relationships without our graphical model, each interaction sequence s_u in the dataset is aggregated to a single embedding vector using additive attention pooling. To achieve this for sequence s with length ℓ_s, let $\mathbf{H}_s = [\mathbf{e}^{i_1}, \ldots, \mathbf{e}^{i_{\ell_s}}] \in \mathbb{R}^{\ell_s \times d}$ be the concatenation of the item embeddings. This is passed through a learned attention function to obtain the attention weights

$$\mathbf{a}_s = \text{softmax}(\mathbf{e}_s), \quad \text{where} \quad \mathbf{e}_s = \mathbf{v}_a^\top \tanh(\mathbf{W}_a \mathbf{H}_s^\top) \in \mathbb{R}^{\ell_s}, \tag{8}$$

and $\mathbf{W}_a \in \mathbb{R}^{h \times d}$ and $\mathbf{v}_a \in \mathbb{R}^h$ are learned weights.

Finally, the item embedding sequence is combined with learned positional embeddings $\mathbf{P}_{1:\ell_s} \in \mathbb{R}^{\ell_s \times d}$ to form

$$\mathbf{X}_s = \mathbf{H}_s + \mathbf{P}_{1:\ell_s}, \tag{9}$$

which is modulated with the attention weights through a weighted sum aggregation[1]

$$\mathbf{s}_s^{(0)} = \sum_{t=1}^{\ell_s} a_{s,t} \mathbf{X}_{s,t}. \tag{10}$$

For efficiency, we cache the matrix $\mathbf{S}$ containing these attention pooled sequence embeddings. For every batch during training, we employ a refresh strategy where only the embeddings of the sequences contained in the batch are recomputed from the latest item-ID embedding table. Then at the start of each epoch, we refresh *all* subsequence vectors in the cache.

SS and SI edges. To integrate these sequence embeddings into the overall MuSICRec graphical structure, we construct a *weighted* adjacency over $\mathcal{S} \cup \mathcal{I}$. This combines both sequence–sequence (SS) and sequence–item (SI) links, enabling information sharing between the training signal of similar sequences, and between sequences and their constituent items.

[1] other interchangeable attention variants such as sinusoidal pooling are possible and will be examined in Section 4.6

SS edges are constructed and weighted using the Jaccard similarity between *sets* of items to determine their similarity:

$$\text{jac}(s, s') = \frac{|\text{set}(\mathcal{I}_s) \cap \text{set}(\mathcal{I}_{s'})|}{|\text{set}(\mathcal{I}_s) \cup \text{set}(\mathcal{I}_{s'})|}. \tag{11}$$

We construct the graph by adding an undirected SS edge (s, s') if $\text{jac}(s, s') \geq \tau$, with the edge weight stored as the Jaccard similarity, $w_{ss} = \text{jac}(s, s')$. We also add SI edges to connect s to each constituent item $i \in \mathcal{I}_s$ with unit weight.

The resulting weighted adjacency is $\hat{\mathbf{A}}_{\text{si}} \in \mathbb{R}^{(n_s+N)\times(n_s+N)}$, where $n_s = |\mathcal{S}|$. We apply symmetric GCN normalization

$$\mathbf{A}_{\text{si}} = \mathbf{D}^{-1/2} \hat{\mathbf{A}}_{\text{si}} \mathbf{D}^{-1/2}, \quad \mathbf{D} = \text{diag}\left(\hat{\mathbf{A}}_{\text{si}} \mathbf{1}\right), \tag{12}$$

and represent $\mathbf{A}_{\text{si}}$ as a sparse tensor for efficient computation.

Propagation and outputs. As for the User–Item graph, we perform L_{si} steps of linear propagation on the Sequence–Item graph:

$$\mathbf{Z}_{\text{si}}^{(\ell+1)} = \mathbf{A}_{\text{si}} \mathbf{Z}_{\text{si}}^{(\ell)}, \qquad \ell = 0, \ldots, L_{\text{si}} - 1. \tag{13}$$

The input at layer 0 is the stacked sequence and item embeddings:

$$\mathbf{Z}_{\text{si}}^{(0)} = \begin{bmatrix} \mathbf{S} \\ \mathbf{I} \end{bmatrix} \in \mathbb{R}^{(N+n_s)\times d}, \tag{14}$$

and the final output is split back out into sequences and items

$$\mathbf{Z}_{\text{si}}^{(L_{\text{si}})} = \begin{bmatrix} \mathbf{S}_{\text{si}} \\ \mathbf{I}_{\text{si}} \end{bmatrix}, \tag{15}$$

where $\mathbf{I}_{\text{si}} \in \mathbb{R}^{N \times d}$ are the SI-updated item representations and $\mathbf{S}_{\text{si}} \in \mathbb{R}^{n_s \times d}$ are the updated sequence representations. These feed into the fusion stage together with the user–item branch and the multimodal branch.

3.4 Multimodal Item-Item Graph

The third graph module within the MuSICRec framework is designed to allow the sharing of information between items that do not co-occur, but which are similar according to their supplementary multimodal data. To this end, we build item–item graphs from

the raw image and text data, then project these modality features into the shared ID embedding space. Each item is then *seeded* with an ID-anchored, per-item *gated* mix of modality embeddings before linear propagation. As suggested by [25] we keep the connectivity of the II graph frozen during training to improve stability. Learning item–item structures online [22] is powerful but expensive and can drift; freezing the kNN content graph keeps training lean and reproducible.

Feature projections. Each item i carries raw visual and textual features, $\mathbf{f}_i^{(v)} \in \mathbb{R}^{d_v}$ and $\mathbf{f}_i^{(t)} \in \mathbb{R}^{d_t}$ which we linearly project to d and ℓ_2-normalize:

$$\tilde{\mathbf{v}}_i = \mathrm{norm}\big(\mathbf{f}_i^{(v)}\mathbf{W}_v\big), \qquad \tilde{\mathbf{t}}_i = \mathrm{norm}\big(\mathbf{f}_i^{(t)}\mathbf{W}_t\big), \tag{16}$$

using learnable weights $\mathbf{W}_v \in \mathbb{R}^{d_v \times d}$ and $\mathbf{W}_t \in \mathbb{R}^{d_t \times d}$.

Per-modality kNN graphs and frozen fusion. We build a cosine-kNN graph among items per modality $m \in \{v, t\}$. $\hat{\mathbf{A}}_{ii}^{(m)}$ is the adjacency matrix with ones on kNN edges. Using standard GCN normalisation:

$$\mathbf{A}_{ii}^{(m)} \;=\; \mathbf{D}^{-1/2}\hat{\mathbf{A}}_{ii}^{(m)}\mathbf{D}^{-1/2}, \qquad \mathbf{D} = \mathrm{diag}\big(\hat{\mathbf{A}}_{ii}^{(m)}\mathbf{1}\big). \tag{17}$$

We then fuse the two modality graphs with a *static* weighting α_v:

$$\mathbf{A}_{ii} \;=\; \alpha_v \mathbf{A}_{ii}^{(v)} + (1-\alpha_v)\,\mathbf{A}_{ii}^{(t)}, \qquad \alpha \in [0,1]. \tag{18}$$

ID-conditioned gating. In contrast to previous works, we introduce a learnable gate that modulates *how much* text and video features to mix, on a per-item basis. Importantly, conditioning this gate on the ID space indirectly mitigates the issue of cross-modal misalignment where one modality suggests two items are similar but other modalities disagree. Modality signals that agree with ID signals are amplified, while ones that do not are suppressed.

Let $\mathbf{E} \in \mathbb{R}^{N \times d}$ be the item ID table, and let $\mathbf{e}^i$ denote row i. We compute a *per-item gate* $\beta_i \in [0,1]$ from $\mathbf{e}^i$:

$$\beta_i \;=\; \sigma\big(\mathbf{w}_\beta^\top \mathbf{e}^i\big), \qquad \mathbf{w}_\beta \in \mathbb{R}^d. \tag{19}$$

The gate blends normalized text and image projections to form an ID-anchored content vector:

$$\mathrm{mix}_i \;=\; (1-\beta_i)\,\tilde{\mathbf{t}}_i + \beta_i\,\tilde{\mathbf{v}}_i, \tag{20}$$

This makes it possible for certain items to depend more or less heavily on text or visual features. For example, it might be reasonable to expect that the similarity of clothing items is determined more by their visual appearance than their textual description. Meanwhile, the similarity of power tools may be determined more by the description of their function than by their visual appearance.

Finally, we use these gated multimodal embeddings to *seed* the multimodal branch together with $\mathbf{e}^i$ at item i by

$$\mathbf{z}_i^{(0)} \;=\; \mathbf{e}^i \;+\; \alpha_{\mathrm{seed}}\,\mathrm{mix}_i, \tag{21}$$

where $\alpha_{\mathrm{seed}} \geq 0$ sets the overall content strength. This makes multimodal content a residual, ID-anchored cue rather than a replacement for IDs with α_{seed} constraining its influence. Empirically, small α_{seed} (0.1 or 0.001 dataset-dependent) works best, which is consistent with treating multimodal content as a calibration aid.

Propagation on the frozen graph. We propagate $\mathbf{Z}_{ii}$over the fused, static adjacency $\mathbf{A}_{ii}$ for L_{mm} steps with the same neighborhood averaging rule used elsewhere:

$$\mathbf{Z}_{ii}^{(\ell+1)} \;=\; \mathbf{A}_{ii}\,\mathbf{Z}_{ii}^{(\ell)}, \qquad \ell = 0, \ldots, L_{\mathrm{mm}} - 1. \tag{22}$$

Output and fusion. We denote the multimodal branch output by $\mathbf{e}_{\mathrm{mm}}^i = \mathbf{z}_i^{(L_{\mathrm{mm}})}$. Later, the model fuses $\mathbf{e}_{\mathrm{mm}}^i$ with $\mathbf{e}_{\mathrm{ui}}^i$ using a fixed scalar weight α_{mm}.

$$\tilde{\mathbf{e}}^i \;=\; \mathbf{e}_{\mathrm{ui}}^i + \alpha_{\mathrm{mm}}\,\mathbf{e}_{\mathrm{mm}}^i, \quad \lambda_{\mathrm{mm}} \in [0,1], \tag{23}$$

3.5 Training and Optimisation

Fused scoring. We score candidate items for user u by the inner product of $\tilde{\mathbf{e}}^i$ the fused item vector and $\mathbf{e}_{\mathrm{ui}}^i$ the UI-branch item vector

$$\hat{y}_{u,i} \;=\; \langle \mathbf{e}_{\mathrm{ui}}^u,\, \tilde{\mathbf{e}}^i \rangle. \tag{24}$$

Pairwise ranking objective (BPR). Training is driven by a pairwise ranking loss over triples $\mathcal{T} = \{(u, i^+, i^-)\}$, where i^+ is observed for u and i^- is an unobserved negative sample (drawn uniformly, or randomised within the batch):

$$\mathcal{L}_{\mathrm{BPR}} \;=\; -\frac{1}{|\mathcal{T}|} \sum_{(u,i^+,i^-)\in\mathcal{T}} \log \sigma\Big(\hat{y}_{u,i^+} - \hat{y}_{u,i^-}\Big), \tag{25}$$

with $\sigma(\cdot)$ the logistic sigmoid. Eq. (25) directly encourages $\hat{y}_{u,i^+} > \hat{y}_{u,i^-}$ under implicit feedback.

Cross-view alignment (user-item ↔ sequence-item). We utilise an InfoNCE-style contrastive loss to align matching entities across views, between each user vector and their sequence node (from the SI branch) and between item representations from the two views. $\mathbf{s}_{si}^u$ denotes the final sequence representation for user u, and define a cosine similarity $\mathrm{sim}(\mathbf{a}, \mathbf{b}) = \big\langle \frac{\mathbf{a}}{\|\mathbf{a}\|}, \frac{\mathbf{b}}{\|\mathbf{b}\|} \big\rangle$. With temperature $\tau > 0$ and a negative set $\mathcal{N}(u)$ of other users' sequences, the loss is defined as

$$\mathcal{L}_{\mathrm{US}} = -\frac{1}{|\mathcal{U}|} \sum_{u\in\mathcal{U}} \log \frac{\exp\big(\mathrm{sim}(\mathbf{e}_{\mathrm{ui}}^u,\, \mathbf{s}_{\mathrm{si}}^u)/\tau\big)}{\sum_{u'\in\mathcal{U}} \exp\big(\mathrm{sim}(\mathbf{e}_{\mathrm{ui}}^u,\, \mathbf{s}_{\mathrm{si}}^{u'})/\tau\big)}. \tag{26}$$

For the positive items in the batch, with $\tilde{\mathbf{e}}^i$ the fused item vector and $\mathbf{e}_{\mathrm{si}}^i$ the SI-branch item vector,

$$\mathcal{L}_{\mathrm{IS}} = -\frac{1}{|\mathcal{B}|} \sum_{i\in\mathcal{B}} \log \frac{\exp\big(\mathrm{sim}(\tilde{\mathbf{e}}^i,\, \mathbf{e}_{\mathrm{si}}^i)/\tau\big)}{\sum_{i'\in\mathcal{B}} \exp\big(\mathrm{sim}(\tilde{\mathbf{e}}^i,\, \mathbf{e}_{\mathrm{si}}^{i'})/\tau\big)}. \tag{27}$$

Multimodal alignment loss. We define a *modality-aware* pairwise loss that aligns the UI-branch user vector $\mathbf{e}_{\mathrm{ui}}^u$ with the corresponding content spaces:

$$\mathcal{L}_{\mathrm{MM}} = -\frac{1}{|\mathcal{T}|} \sum_{(u,i^+,i^-)\in\mathcal{T}} \sum_{m\in\{t,v\}} \log \sigma\big(\langle \mathbf{e}_{\mathrm{ui}}^u,\, \tilde{\mathbf{m}}_{i^+} - \tilde{\mathbf{m}}_{i^-} \rangle\big), \tag{28}$$

Regularisation and total objective. The complete loss combines supervised ranking, optional cross-view alignment, and weight regularisation:

$$\mathcal{L} \;=\; \mathcal{L}_{\mathrm{BPR}} \;+\; \lambda_u\,\mathcal{L}_{\mathrm{US}} \;+\; \lambda_i\,\mathcal{L}_{\mathrm{IS}} \;+\; \lambda_{\mathrm{mm}}\,\mathcal{L}_{\mathrm{MM}}, \tag{29}$$

where λ_u, λ_i and λ_{mm} are non-negative scalar weights.

Table 1: Overall performance comparison of the different recommendation models. The best result on each dataset for each metric is boldened, and the second best is underlined.

Datasets	Baby				Sports				Elec			
Metrics	**R@10**	**R@20**	**N@10**	**N@20**	**R@10**	**R@20**	**N@10**	**N@20**	**R@10**	**R@20**	**N@10**	**N@20**
LightGCN	0.0336	0.0549	0.0178	0.0231	0.0402	0.0631	0.0213	0.0270	0.0288	0.0427	0.0154	0.0189
SGL	0.0361	0.0586	0.0188	0.0244	0.0425	0.0688	0.0227	0.0293	0.0292	0.0439	0.0158	0.0195
SASRec	0.0298	0.0478	0.0150	0.0195	0.0274	0.0421	0.014	0.0177	0.0241	0.0368	0.0128	0.0160
BERT4Rec	0.0207	0.0355	0.0102	0.0139	0.0176	0.0273	0.0094	0.0118	0.023	0.0358	0.0117	0.0149
FEARec	0.0285	0.048	0.0146	0.0195	0.0281	0.0456	0.0143	0.0187	0.0255	0.039	0.0136	0.0170
SRGNN	0.0260	0.0418	0.0132	0.0172	0.0196	0.0314	0.0102	0.0132	0.0242	0.0367	0.0128	0.0159
MMGCN	0.0253	0.0436	0.0124	0.0169	0.0275	0.0447	0.014	0.0183	0.0173	0.0285	0.0086	0.0114
GRCN	0.0359	0.0574	0.0191	0.0245	0.0371	0.0606	0.0189	0.0249	0.0232	0.0367	0.0121	0.0155
VBPR	0.0313	0.0517	0.0163	0.0214	0.0406	0.0638	0.0204	0.0263	0.0228	0.0348	0.0113	0.0143
BM3	0.0371	0.0612	0.0189	0.0250	0.0435	0.0698	0.0224	0.0290	0.0302	0.0450	0.0164	0.0201
MGCN	0.0420	0.0666	0.0222	0.0282	0.0476	0.0750	0.0251	0.0320	0.0330	0.0495	0.0179	0.0221
FREEDOM	0.0411	0.0655	0.0210	0.0272	0.0479	0.0752	0.0246	0.0315	0.0315	0.0482	0.0169	0.0211
LGMRec	0.0412	0.0649	0.0212	0.0271	0.0455	0.0715	0.0238	0.0303	0.0328	0.0496	0.0176	0.0219
SMORE	0.0439	0.0684	0.0229	0.0291	0.0480	0.0746	0.0254	0.0321	0.0333	0.0499	0.0181	0.0222
MuSICRec*	**0.0455**	**0.0718**	**0.0235**	**0.0300**	**0.0508**	**0.0805**	**0.0267**	**0.0341**	**0.0350**	**0.0519**	**0.0189**	**0.0232**
Improvement	3.64%	4.97%	2.62%	3.09%	5.83%	7.05%	5.12%	6.23%	5.11%	4.01%	4.42%	4.50%

4 Experiments

We perform multiple experiments to evaluate our proposed model's performance and benchmark it against a broad range of existing state-of-the-art recommendation systems, including generic, multimodal, and sequential methods in our analysis.

A key challenge in this breadth arises from the different evaluation protocols used in sequential and multimodal recommendation research, for example, variations in train-validation-test data split and metric definition. Consequently, although we are comparing on standard datasets, we were forced to develop a more unified evaluation protocol to enable fair and meaningful comparisons across the broad range of distinct approaches.

In addition to benchmarking against prior work, we conduct an ablation study to determine the contribution of individual system components. We also analyse the sensitivity of MuSICRec with respect to λ_u and λ_i, which control the alignment of the user-item and sequence-item views.

4.1 Experimental Setup

Multimodal recommender systems are atemporal and typically rely on random train/test splits. This does not mesh well with sequential recommendation, where the prediction target must lie in the future relative to the training data. We therefore define a unified evaluation protocol where we use all interactions except the last two for training, keeping the penultimate and final interactions for validation and testing, respectively. We deviate from the typical leave-one-out split usually used to evaluate sequential recommendation by not feeding the validation item back into the test prefix. This means that we provide the same exact prefix for validation and testing, which only includes the training interactions. This is done because in multimodal and generic recommendation systems, the validation interaction is kept out and not used to build or update the user-item graph before testing. In other words, there is no message passing or representation updates that use the penultimate validation item. Similarly, in MuSICRec, we neither update the user-item adjacency matrix nor the sequence-item adjacency matrix or initial sequence nodes (formed by attention pooling over item ID embeddings forming user history) with the validation item. This avoids leakage by ensuring information from held-out interactions cannot slip into test item representation through graph updates and ensures an equitable comparison between all baselines under the same strict leave-two-out split. Our choice is consistent with the practice of building multimodal recommenders on fixed interaction graphs and recent cautions about leakage in offline recommender evaluation [7]. All baselines were rerun with this leave-two-out split, and thus, numbers may differ slightly from those reported in their respective publications.

4.2 Datasets

Table 2: Datasets Statistics

Dataset	Users	Items	Reviews	Sparsity (%)
Baby	19,445	7,333	160,792	99.89
Sports	35,598	18,357	296,337	99.95
Elec	192,403	63,001	1,689,188	99.99

The publicly available Amazon datasets are widely used to evaluate sequential and multimodal recommender systems. These contain product reviews extracted by Amazon.com and are among the few datasets that provide textual and visual data that could be utilised for our task. We focus on the Baby, "Sports and Outdoors" and Electronics datasets in our evaluation, and refer to them as

Baby, **Sports** and **Elec**. The text and image URLs are provided with the metadata for Amazon datasets. We utilise the published 384-dimensional text features and 4096-dimensional visual features. 5-core filtering of users and items is carried out on all three datasets and user ratings are treated as positive interactions.

Recall (R@k) and Normalised Discounted Cumulative Gain (N@k) are the two metrics we report in our evaluation. Under our leave-two-out protocol, the validation and test sets comprise one item per user. R@k indicates whether the ground-truth item appears in the top-k list, and N@k is rank-sensitive, applying a logarithmic discount to the position of the correct item, thus rewarding higher placement.

4.3 Baselines

We compare against state-of-the-art multimodal and sequential baselines. These baselines fall under these categories:

- ID-only collaborative methods: LightGCN[5] and SGL[10]
- Multimodal models: VBPR[4], MMGCN[16], MGCN[21], GRCN [19], FREEDOM[25], LGMRec[3], BM3[26] and SMORE[9]
- sequential baselines: SASRec[8], BERT4Rec[13], FEARec[2], and SR-GNN[18].

4.4 Implementation Details

MuSICRec and all baselines were implemented in PyTorch, trained with an Adam optimiser, Xavier initialisation, and fixed user item and sequence dimensionality to 64. For MuSICRec, we set the number of GCN layers in the item-item graph at $L_{ii} = 1$, the user-item graph at $L_{ui} = 3$, and the sequence-item graph at $L_{si} = 2$. We set $\alpha_v = 0.1$, vary α_{seed} in $\{0.001, 0.01, 0.1\}$ and cap the maximum sequence length at $L_{\max} = 50$. We train up to 1000 epochs with early stopping (patience) set to 20, using validation R@20 as the training stopping indicator. MuSICRec, LightGCN, and all multimodal baselines are implemented within the unified MMRec framework. All sequential baselines and SGL are trained with the RecBole library [23]. MMRec is based on and integrates with RecBole. Both evaluation pipelines were made to align with each other, setting training and valid/test batch size to 512 and 4096, respectively, and using the leave-two-out split described in Section 4.1. Hyperparameter sweeps for baselines follow their paper's recommendation, and the best test results, based on valid R@20, are reported for all models. All baselines and MuSICRec use BPR loss as the primary objective for optimisation.

4.5 Performance Comparison

Table 1 reports R@10/R@20 and N@10/N@20 on **Baby**, **Sports** and **Elec**. MuSICRec achieves the best results across all datasets and metrics. On **Baby**, MuSICRec surpasses the best baseline by an average across all metrics of 3.58%. Similarly, average improvements of 6.05% and 4.51% were obtained on **Sports** and **Elec**, respectively. The best baselines are consistently multimodal, with larger gains obtained over sequential models and LightGCN and SGL. Averaged over all datasets and metrics, MuSICRec improves by 4.72% relative to the next best result.

These results show that MuSICRec, regardless of dataset size and sparsity, consistently outperforms state-of-the-art multimodal and sequential recommender systems. These gains indicate the utility of building and propagating signals from the sequence-item graph and passing ID-conditioned mix of multimodal information into the frozen item-item graph, leading to consistent gains. These remain beneficial even at a larger scale as seen on **Elec**.

4.6 Ablation Study

To isolate the contribution of MuSICRec's individual components, we conduct an ablation study evaluating the following variations of the model: **(i) w/o UI graph** removes user-item graph contribution
(ii) w/o SI graph removes sequence-item graph
(iii) w/o MM removes multimodal item-item graph
(iv) w/o MM-ID seed disables ID-guided multimodal seeding
(v) w/ Sinusoidal Pooling replaces learned sequence attention pooling with a parameter-free pooling that adds fixed sinusoidal positional encodings to item embeddings before aggregation.
As shown in Figures 3 and Figures 7a, and 7b in Appendix A, dropping UI graph propagation and aggregation yields the largest degradation on all datasets. Multimodal and sequential signals rely on and complement collaborative behaviour relationships; thus, the user-item graph cannot be fully substituted. Removing SI consistently hurts performance, showing that pooling sequence representations into the graph adds orthogonal information that complements the UI graph. **w/o MM** reduces accuracy as would be expected from removing item multimodal contribution entirely. **w/o MM-ID seed** performs below the full model but above **w/o MM**, indicating that seeding and gating the multimodal information with item IDs prior to item-item graph propagation improves alignment and denoising. Sinusoidal pooling underperforms learned attention pooling, demonstrating that data-dependent, learnable fusion is needed to decide how much each interaction (recent vs. early, bursty vs. steady) should influence the sequence representation prior to graph propagation.

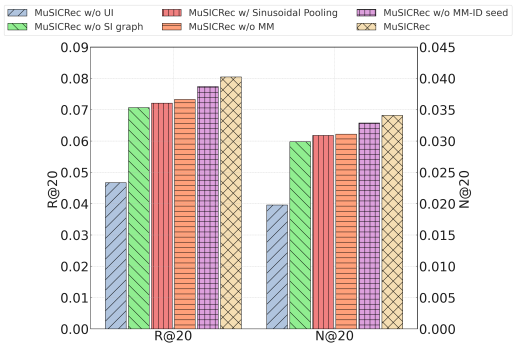


Figure 3: *Sports* Ablation

4.7 Sensitivity Study

We analyse λ_u and λ_i, which control the contribution of the contrastive losses to the overall objective, balancing the input of the sequence-item view, affecting recommendation performance. Figures 4 and 5 and Figures 7c, 7d, 7e and 7f in Appendix A show the sweeps conducted and the results obtained. MuSICRec performs best when setting λ_u to 0.1 on **Sports** and **Elec** and to 0.001 on

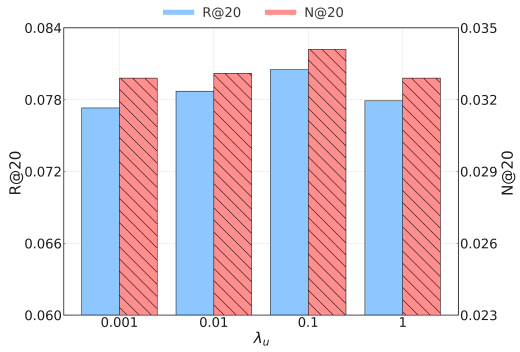


Figure 4: λ_u *Sports* Sensitivity

Baby. λ_i is best when set to 0.01 for all datasets. We also note that parameter sensitivity is low, and performance is generally stable with only mild accuracy drift when setting λ_u and λ_i too high or too low.

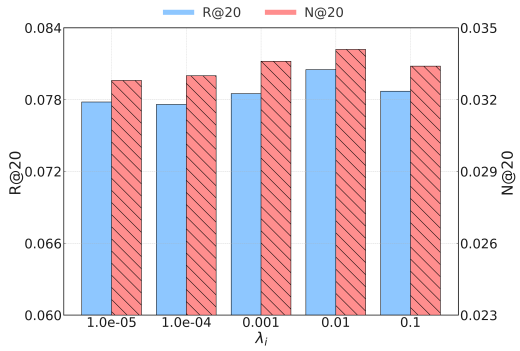


Figure 5: λ_i *Sports* Sensitivity

4.8 User History Depth Study

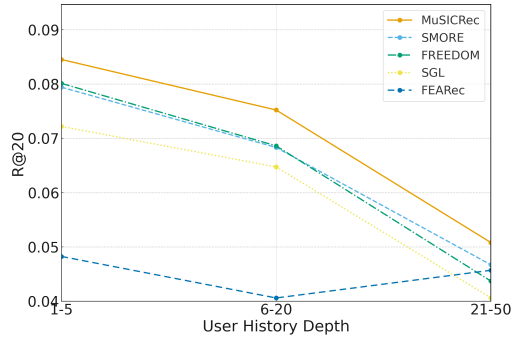


Figure 6: *Sports* User History Depth Analysis

We filter users by history length, with near-cold users having 1-5 interactions, warm users with 6-20 interactions, and long-term users with 21-50 interactions. This bucket-wise evaluation reveals behaviour and cold/near-cold conditions while checking that gains still persist for users with denser histories. As can be seen in Figure 6, MuSICRec delivers the largest gains for the 1-5 bucket, while improving accuracy to a lesser extent for 6-20 and 21-50 users. This shows that the extensive collaborative filtering and information sharing in MuSICRec helps to tackle common cold start and data sparsity issues. Similarly, cross and intra-modal denoised multi-modal representations also allow MuSICRec to leverage relevant side information when histories are short. However, it is pleasing to see that as histories lengthen, performance remains strong and the technique does not merely overfit to short histories, to the detriment of other users.

4.9 Computation Cost

From a complexity standpoint, **MuSICRec** with $L_{ii} = 1$ $L_{ui} = 3$ $L_{si} = 2$ executes six sparse propagations per epoch and one global sequence-cache refresh. Each sparse propagation costs $\Theta(\text{nnz}(\mathbf{A}) \cdot d)$ for adjacency $\mathbf{A}$ (SpMM on $[N \times N] \times [N \times d]$), giving a leading term $\Theta((3|L_{\text{ui}}| + 2|L_{\text{si}} + 1|L_{\text{mm}}|) \cdot d)$. The sequence node refresh applies a single-head additive attention over at most $L_{\max} = 50$ tokens per sequence in every batch, adding $\Theta(Batch\_Size \cdot L \cdot d)$. No transfroms, biases or nonlinearities are applied between layers.

We benchmark computation cost using train epoch time and peak allocated GPU memory over full train→val→test for each model. Table 3 shows the computation cost results on Sports dataset. Compared to the strongest baseline from Table 1 (SMORE), MuSICRec trains faster per epoch while using lower peak VRAM, despite achieving better metrics. We train at faster epoch time and have lower GPU utilisation compared to most baseline models compared against.

Table 3: Computation cost on Sports with train batch size of 512. This shows pure training epoch time and peak GPU memory allocated over train→val→test.

Model	Train Epoch Time (s)	Peak GPU Memory (MiB)
LightGCN	6.90	681.32
SGL	16.31	355.21
SASRec	16.53	1531.80
BERT4Rec	23.01	1427.60
FEARec	40.39	1774.61
SRGNN	59.22	1302.96
MMGCN	60.05	2033.10
GRCN	18.96	2820.14
VBPR	2.60	1871.70
BM3	8.84	2700.88
MGCN	13.16	3620.33
FREEDOM	8.96	3638.33
LGMRec	19.54	2711.77
SMORE	18.52	2712.22
MuSICRec	13.31	2043.97

5 Conclusion

In this paper, we proposed a novel model MuSICRec that leverages sequences of interactions as a class of nodes within a new sequence-item graph. It unifies sequential signals from this graph user-item and multimodal item-item signals in a single contastive learning framework. In our extensive experiments, it delivered consistent gains over recent strong baselines. Ablations verified the contribution of each component, and sensitivity analysis showed robustness to λ_u and λ_i. In future work, we aim to improve sequence representation learning to further better performance.

References

[1] Xuheng Cai, Chao Huang, Lianghao Xia, and Xubin Ren. 2023. LightGCL: Simple Yet Effective Graph Contrastive Learning for Recommendation. arXiv:2302.08191 [cs.IR] https://arxiv.org/abs/2302.08191

[2] Xinyu Du, Huanhuan Yuan, Pengpeng Zhao, Jianfeng Qu, Fuzhen Zhuang, Guanfeng Liu, and Victor S. Sheng. 2023. Frequency Enhanced Hybrid Attention Network for Sequential Recommendation. arXiv:2304.09184 [cs.IR] https://arxiv.org/abs/2304.09184

[3] Zhiqiang Guo, Jianjun Li, Guohui Li, Chaoyang Wang, Si Shi, and Bin Ruan. 2024. LGMRec: Local and Global Graph Learning for Multimodal Recommendation. arXiv:2312.16400 [cs.IR] https://arxiv.org/abs/2312.16400

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

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

[6] Balázs Hidasi, Massimo Quadrana, Alexandros Karatzoglou, and Domonkos Tikk. 2016. Parallel recurrent neural network architectures for feature-rich session-based recommendations. In *Proceedings of the 10th ACM conference on recommender systems*. 241–248.

[7] Yitong Ji, Aixin Sun, Jie Zhang, and Chenliang Li. 2023. A Critical Study on Data Leakage in Recommender System Offline Evaluation. *ACM Transactions on Information Systems (TOIS)* (2023). doi:10.1145/3569930

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

[9] Rongqing Kenneth Ong and Andy W. H. Khong. 2025. Spectrum-based Modality Representation Fusion Graph Convolutional Network for Multimodal Recommendation. In *Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining* (Hannover, Germany) *(WSDM '25)*. Association for Computing Machinery, New York, NY, USA, 773–781. doi:10.1145/3701551.3703561

[10] Ruihong Qiu, Zi Huang, Hongzhi Yin, and Zijian Wang. 2022. Contrastive Learning for Representation Degeneration Problem in Sequential Recommendation. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining (WSDM '22)*. ACM, 813–823. doi:10.1145/3488560.3498433

[11] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618* (2012).

[12] Yehjin Shin, Jeongwhan Choi, Hyowon Wi, and Noseong Park. 2024. An attentive inductive bias for sequential recommendation beyond the self-attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 8984–8992.

[13] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management* (Beijing, China) *(CIKM '19)*. Association for Computing Machinery, New York, NY, USA, 1441–1450. doi:10.1145/3357384.3357895

[14] Chenyang Wang, Weizhi Ma, Chong Chen, Min Zhang, Yiqun Liu, and Shaoping Ma. 2023. Sequential Recommendation with Multiple Contrast Signals. *ACM Trans. Inf. Syst.* 41, 1, Article 11 (Jan. 2023), 27 pages. doi:10.1145/3522673

[15] Qifan Wang, Yinwei Wei, Jianhua Yin, Jianlong Wu, Xuemeng Song, and Liqiang Nie. 2021. Dualgnn: Dual graph neural network for multimedia recommendation. *IEEE Transactions on Multimedia* 25 (2021), 1074–1084.

[16] Yinwei Wei, Xiang Wang, Liqiang Nie, Xiangnan He, Richang Hong, and Tat-Seng Chua. 2019. MMGCN: Multi-modal graph convolution network for personalized recommendation of micro-video. In *Proceedings of the 27th ACM international conference on multimedia*. 1437–1445.

[17] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. 2021. Self-supervised Graph Learning for Recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21)*. ACM, 726–735. doi:10.1145/3404835.3462862

[18] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. 2019. Session-Based Recommendation with Graph Neural Networks. *Proceedings of the AAAI Conference on Artificial Intelligence* 33, 01 (July 2019), 346–353. doi:10.1609/aaai.v33i01.3301346

[19] Donghan Yu, Ruohong Zhang, Zhengbao Jiang, Yuexin Wu, and Yiming Yang. 2021. Graph-revised convolutional network. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2020, Ghent, Belgium, September 14–18, 2020, Proceedings, Part III*. Springer, 378–393.

[20] Junliang Yu, Xin Xia, Tong Chen, Lizhen Cui, Nguyen Quoc Viet Hung, and Hongzhi Yin. 2024. XSimGCL: Towards Extremely Simple Graph Contrastive Learning for Recommendation. *IEEE Trans. on Knowl. and Data Eng.* 36, 2 (Feb. 2024), 913–926. doi:10.1109/TKDE.2023.3288135

[21] Penghang Yu, Zhiyi Tan, Guanming Lu, and Bing-Kun Bao. 2023. Multi-View Graph Convolutional Network for Multimedia Recommendation. In *Proceedings of the 31st ACM International Conference on Multimedia (MM '23)*. ACM, 6576–6585. doi:10.1145/3581783.3613915

[22] Jinghao Zhang, Yanqiao Zhu, Qiang Liu, Shu Wu, Shuhui Wang, and Liang Wang. 2021. Mining latent structures for multimedia recommendation. In *Proceedings of the 29th ACM international conference on multimedia*. 3872–3880.

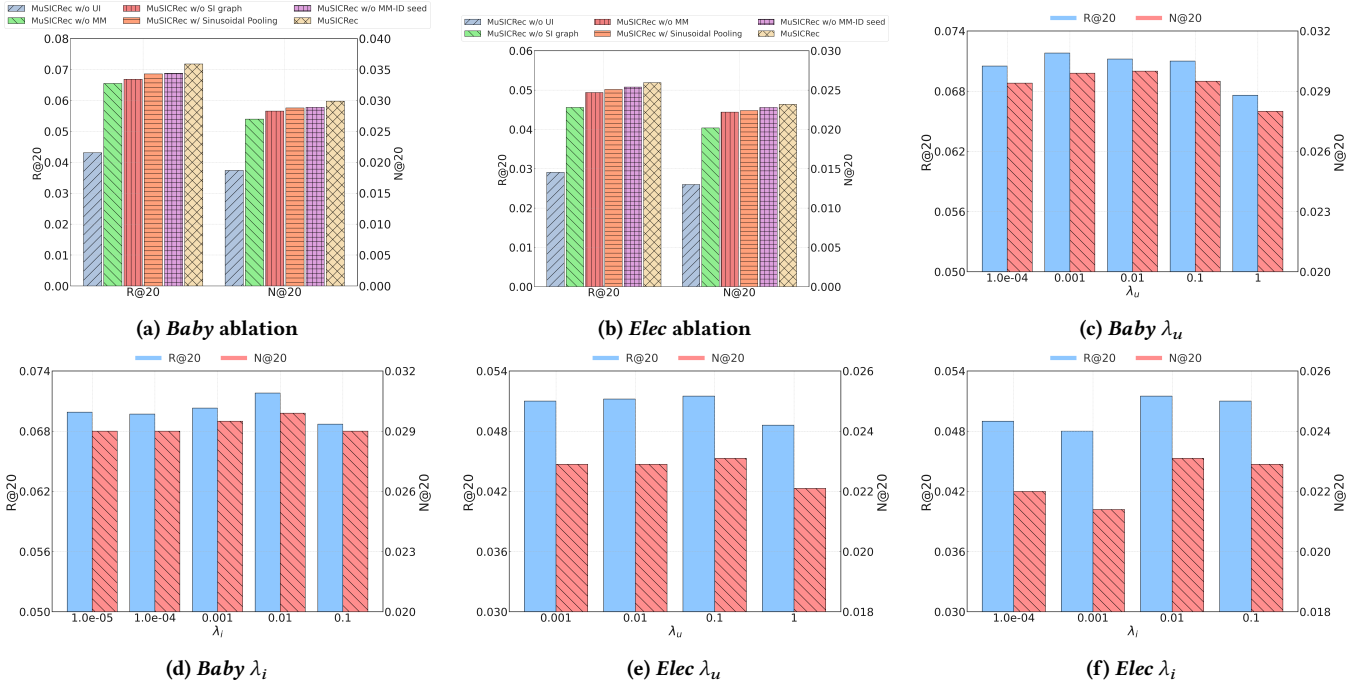
[23] Wayne Xin Zhao, Shanlei Mu, Yupeng Hou, Zihan Lin, Yushuo Chen, Xingyu Pan, Kaiyuan Li, Yujie Lu, Hui Wang, Changxin Tian, Yingqian Min, Zhichao Feng, Xinyan Fan, Xu Chen, Pengfei Wang, Wendi Ji, Yaliang Li, Xiaoling Wang, and Ji-Rong Wen. 2021. RecBole: Towards a Unified, Comprehensive and Efficient Framework for Recommendation Algorithms. arXiv:2011.01731 [cs.IR] https://arxiv.org/abs/2011.01731

[24] Kun Zhou, Hui Yu, Wayne Xin Zhao, and Ji-Rong Wen. 2022. Filter-enhanced MLP is All You Need for Sequential Recommendation. In *Proceedings of the ACM Web Conference 2022* (Virtual Event, Lyon, France) *(WWW '22)*. Association for Computing Machinery, New York, NY, USA, 2388–2399. doi:10.1145/3485447.3512111

[25] Xin Zhou and Zhiqi Shen. 2023. A tale of two graphs: Freezing and denoising graph structures for multimodal recommendation. In *Proceedings of the 31st ACM International Conference on Multimedia*. 935–943.

[26] Xin Zhou, Hongyu Zhou, Yong Liu, Zhiwei Zeng, Chunyan Miao, Pengwei Wang, Yuan You, and Feijun Jiang. 2023. Bootstrap latent representations for multi-modal recommendation. In *Proceedings of the ACM Web Conference 2023*. 845–854.

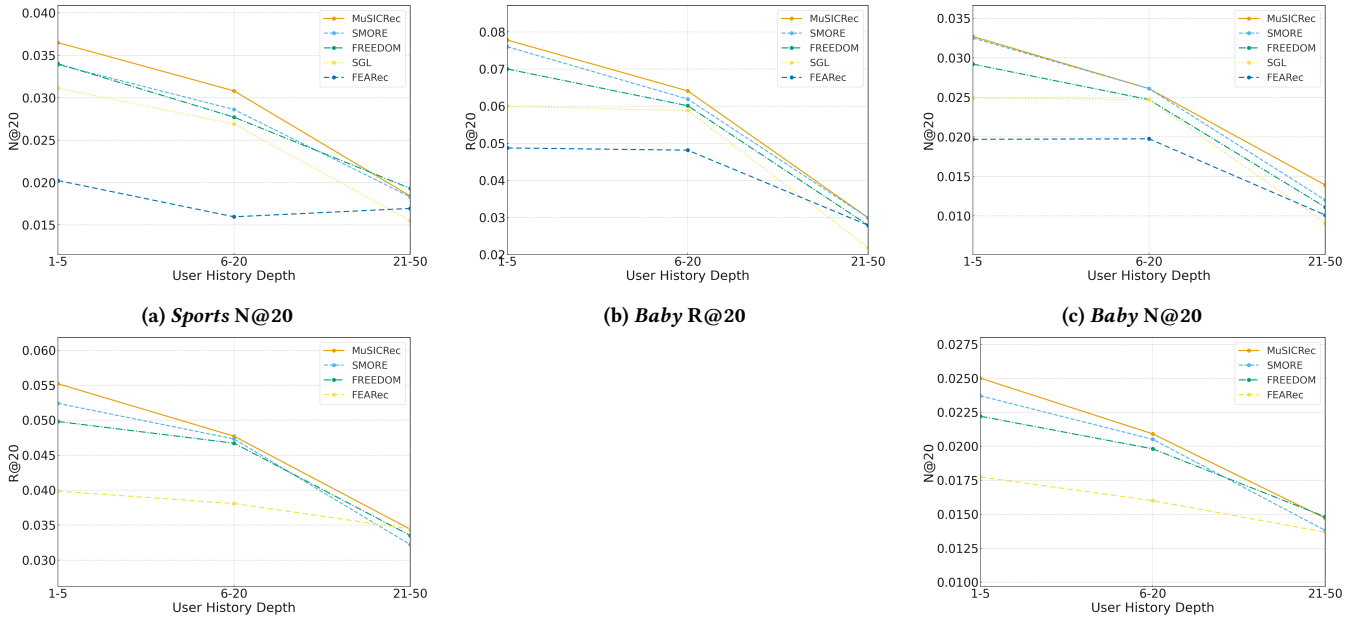
A Additional Figures



(a) *Baby* ablation (b) *Elec* ablation (c) *Baby* λ_u

(d) *Baby* λ_i (e) *Elec* λ_u (f) *Elec* λ_i

Figure 8: Ablation and sensitivity analyses (additional results).



(a) *Sports* N@20 (b) *Baby* R@20 (c) *Baby* N@20

(d) *Elec* R@20 (e) *Elec* N@20

Figure 9: User history depth analyses (additional results).