NoConf
#NoID

EmoRec: Graph-based recommender systems with explicit negative feedback encoding

Anonymous NoConf submission

Paper ID NoID

Abstract

Graph neural networks have long been studied in the context of recommendation systems, as they effectively replicate social interactions between users and can accurately extract higher-order joint signals. These features are crucial in creating high-quality recommendations. Traditional recommendation approaches often rely on negative feedback through indirect means, such as regularization of loss functions or contrastive sampling. However, these methods neglect the structural representation of user interactions in the form of graphs or increase the complexity of the models to account for all available information about interactions. In this paper, we present the Emotional Recommendation System (EmoRec), which incorporates the concept of considering emotions (different types of user feedback) to reduce computational complexity by clustering feedback. ***The final model, described in the following, serves as an example.***

1. Introduction

Modern recommender systems struggle with a fundamental tension: users expect platforms to understand both their enthusiastic likes and categorical dislikes with surgical precision. While graph neural networks (GNNs) have transformed recommendation engines through their ability to model complex user-item interactions [9], most implementations treat negative feedback as a monolithic signal—if they consider it at all. This simplification persists despite empirical evidence that users express rejection through diverse behaviors ranging from passive skips (implicit negative) to active dislikes (explicit negative), each carrying distinct semantic meaning about preference boundaries [7, 25].

Consider the emotional granularity in a streaming platform scenario (Fig. **??**):

- *Explicit Positive*: **LATER**
- *Implicit Positive*: **LATER**
- *Explicit Negative*: **LATER**
- *Implicit Negative*: **LATER**

Traditional GNN approaches [32] collapse these signals into binary interactions, losing crucial contextual information about the intensity and nature of preferences. Recent attempts to model feedback polarity either increase model complexity through separate interaction graphs [21] or rely on contrastive sampling techniques that inadvertently amplify noise [23]. The result is what we term **EMPTY HERE**.

Our analysis identifies three main systemic limitations in current negative feedback handling:

- **Signal Homogenization**: Treating skips, dislikes, and low ratings as equivalent negative signals [11]
- **Contextual Blindness**: Ignoring the temporal relationship between explicit/implicit feedback within sessions [27]
- **Computational Inflation**: Complexity growth when modeling separate positive/negative interaction graphs [5]

We propose *EmoRec*, an emotion-clustered graph architecture that addresses these challenges through feedback disentanglement and topological sparsification. Our key innovation replaces monolithic user nodes with four emotion-specific proxies: Explicit Positive (EP), Implicit Positive (IP), Explicit Negative (EN), and Implicit Negative (IN). As illustrated in Fig. **?? WILL BE LATER**, each emotion cluster dynamically aggregates relevant item embeddings:

- **EP Clusters** capture strong preferences (e.g., likes, 5-star ratings)
- **IP Clusters** model passive consumption (e.g., watch time, replays)
- **EN Clusters** learn hard rejection patterns (e.g., dislikes, negative reviews)
- **IN Clusters** detect soft aversion signals (e.g., skips, fast-forwards)

The EmoRec framework introduces **N** key mechanisms to be detailed in subsequent sections: **(1) ; (2) ; and (3)** . Preliminary experiments show our clustered approach achieves **METRIC WILL BE LATER** NDCG@10 with **X×** faster training than baseline GNNs, proving that emotional granularity need not come at computational expense.

[Full implementation details, including emotion cluster initialization strategies and multi-channel convolution operators, will be comprehensively described in the final version alongside expanded ablation studies.]

The remainder of this paper is organized as follows: Section 2 describes in detail the research conducted in the field of graph recommendation systems, taking into account the negative feedback. Section 3 details the EmoRec architecture and emotion clustering methodology. Section 4 evaluates performance against 14 baseline models across three recommendation domains. Section **??** discusses broader implications for emotionally-aware recommender systems.

2. Related work

Recent studies explore encoding negative feedback directly in encoder-based models. In this paper, we mainly focused on researching works combining the use of graph neural networks and practices of encoding negative connections directly, which helped us analyze the current limits that the authors faced and which we ourselves were able to identify, and tried to cope with them by proposing a new method of encoding feedback from users.

2.1. Sequential Neural Network for Recommendation

The latest advances in sequential recommendation systems have significantly expanded our ability to model temporal user interactions. Initial studies were conducted by Kang and McAuley [14], who focused on predicting future user preferences based on behavioral trajectories, and Li et al. [15], who explored similar aspects. Further research delved into diverse behavioral patterns: Wang et al. [31] identified periodic and short-term trends, while Luo et al. [22] utilized spatiotemporal signals to enhance recommendation accuracy. To improve training, Sun et al. [30] incorporated masked item prediction, and Ma et al. [24] introduced intention disentanglement. More recently, contrastive learning has gained attention as a self-supervised approach, with Chen et al. [6] and Xie et al. [41] exploring its applications. Addressing this limitation, Jin et al. [13] integrated the Ebbinghaus Forgetting Curve to model recency effects in user memory and optimize trade-offs in multi-objective recommendations. Additionally, metric learning has emerged as a promising method for handling implicit feedback imbalance, with Wang et al. [12] introducing unbiased negative sampling and a push–pull loss function. Complementary advancements include AutoMLP, developed by Li et al. [17], which automates short-term interest modeling, and graph-based models like SURGE, proposed by Chang et al. [3], which capture relational dependencies in sequential behaviors. Collectively, these innovations drive the evolution of adaptive recommendation systems.

2.2. Graph Neural Network for Recommendation

Many researchers have explored approaches based on graph neural networks to improve recommendation systems by capturing intricate high-order signals from user–item interaction graphs. For example, Wang et al. [32] proposed a model that aggregates multi-hop relational information to reveal complex behavioral patterns using the knowledge graph and the attention mechanism, while He et al. [9] demonstrated that it is possible to effectively simplify the GCN architecture. Wu et al. [38] introduced novel self-supervised learning methods that improve representation quality, and Yu et al. [44] experimented with contrastive learning, which discards graph augmentation and instead adds uniform noise to the embedding space to create contrasting representations.

In parallel, some authors have addressed theoretical challenges in refining these graph models. Hao et al. [8] proposed a pre-training strategy that initializes GCNs more effectively, and Yu et al. [46] later developed low-pass collaborative filter networks to reduce noise in the graph signal. Zhao et al. [49] introduced a multi-view intent disentanglement mechanism that separates different user intents across various graph views. Moreover, contrastive augmentation techniques have been applied by Yu et al. [43, 45], Lin et al. [19], and Cai et al. [2] to enforce consistency across different graph representations.

Recent innovations further extend these ideas by integrating Transformer layers on top of GCNs—a method advanced by Xia et al. [40] and Wang et al. [34] — to mine richer representations from sequential data. Complementing these developments, Chen et al. [4] proposed a macro graph neural network framework for online billion-scale recommender systems, which groups similar micro nodes into macro nodes to dramatically reduce computational complexity. Oh et al. [26] introduced TempGNN, a temporal graph neural network framework that captures both structural and temporal dynamics in dynamic session-based recommendations. Additionally, Yuan et al. [47] developed an amplify graph learning approach via sparsity completion to effectively address data sparsity by integrating higher-order interaction features as latent perturbations. Together, these diverse strategies are paving the way for more robust, scalable, and adaptive recommendation systems.

2.3. Negative Feedback Learning for Recommendation

Research on negative feedback learning spans diverse approaches to model user preferences and optimize sampling strategies. Zhao et al. [50] and Qin et al. [28] leverage reinforcement learning with recurrent and graph-based architectures to capture item transitions and feedback dynamics. A common thread across studies is the nuanced treatment of implicit signals: Gong et al. [7] differentiate positive (time spent), negative (skipped items), and neutral (temporal context) feedback in news recommendations, while Mei et al. [25] demonstrate that explicit negative targets (e.g., song skips) reduce training time by 60% and improve accuracy in music personalization. These works highlight the importance of domain-specific feedback interpretation—skipping behavior, for instance, serves dual roles as both negative preference signals (news/music) and engagement metrics (short-video platforms).

Negative sampling optimization emerges as another key focus. Qin et al. [33] employ knowledge graphs to generate informative counterexamples, whereas Lyu et al. [23] propose AutoSample, a framework that adaptively selects optimal negative samplers by aligning them with dataset statistics and model capacity through differentiable training, addressing the model-dataset mismatch via loss-to-instance approximation and adaptive search.

Industrial solutions further unify these themes. Wang et al. [36] propose a 'not-to-recommend' loss function for sequential recommenders, integrating both explicit and implicit negative feedback during training. To address the challenge of measuring responsiveness, they develop a counterfactual simulation framework, validating their approach through live experiments on a large-scale industrial system. Similarly, Pan et al. [27] design a multi-objective framework for billion-scale short-video platforms, decoupling skip analysis from watch-time prediction. These works underscore the scalability of negative feedback integration, balancing precision with computational efficiency across domains.

2.4. Graph Neural Network with Negative Feedback Learning for Recommendation

Recent advances in graph-based recommendation systems underscore the critical role of explicitly modeling negative feedback to refine user preference representation. While traditional approaches predominantly focus on positive interactions, emerging methodologies directly integrate negative signals into graph structures and learning frameworks, addressing both structural and semantic challenges.

Direct coding in the graph model. Wang et al. [35] propose NFARec, combining hypergraph convolutions with a Transformer Hawkes Process to model temporal dependencies in feedback sequences, where the Hawkes mechanism captures dynamic user sentiment shifts for improved polarity prediction. This approach outperforms traditional GNNs, particularly in sparse scenarios.

Wu et al. [39] address high-frequency negative signals via DFGNN's dual-frequency graph filter, separating positive/negative interactions while mitigating representation collapse through signed regularization. Huang et al. [11] introduce SiGRec with dual encoders and a Sign Cosine loss, demonstrating that nuanced negative feedback interpretation improves ranking metrics.

Key principles include structural adaptation (e.g., hypergraphs), frequency-aware processing, and temporal modeling (e.g., Hawkes), transforming negative signals into semantically rich inputs for robust preference modeling.

Direct Coding in the Graph Model & Using Self-Attention Mechanism

Recent advances in graph-based recommendation models emphasize the integration of both positive and negative feedback through structured representations and attention mechanisms. PANE-GNN [21] and SIGformer [5] propose contrasting strategies for modeling signed user-item interactions: PANE-GNN separates feedback into dual bipartite graphs with interest/disinterest embeddings and contrastive denoising, while SIGformer leverages transformer architectures with spectral-aware positional encodings to holistically capture collaborative signals in signed graphs. These approaches demonstrate that explicit negative feedback encoding enhances user preference disentanglement compared to traditional graph-based methods.

In multi-behavior recommendation, Zhang et al. [48] (MB-EBIH) and Xu et al. [42] (MBSSL) address auxiliary behavior noise and sparsity through heterogeneous graph modeling. MB-EBIH introduces explicit behavior interaction weights via pre-trained GNNs, whereas MBSSL employs self-supervised discrimination at inter- and intra-behavior levels, combined with gradient-based optimization to balance auxiliary and target tasks.

Further innovations focus on refining feedback quality and scope. DRPN [10] denoises implicit news feedback by mutually reinforcing positive (e.g., reading time) and negative (e.g., skips) signals, proving that bidirectional noise suppression outperforms single-feedback baselines. Extending beyond item-level interactions, RELIFE [37] reranks recommendations using list-level hybrid feedback, disentangling user interests and disinterests through contrastive alignment of historical and candidate list patterns. These methods underscore the importance of holistic feedback integration across granularities (item vs. list) and modalities (explicit vs. implicit).

2.5. Contrastive Representation Learning for Recommendations

Recent works have adapted contrastive learning paradigms to address recommendation challenges through improved representation learning. Graph-based approaches demonstrate particular promise: Liao et al. [18] mitigates false negatives in academic reviewer recommendations via pseudo-negative labeling in graph contrastive learning (GCL), while Li et al., [16] (GCL-MO) enhances e-commerce personalization by optimizing multi-objective contrastive loss for long-tail item representations.

For sequential recommendation, Seshadri et al. [29] integrates negative feedback (e.g., music skips) into contrastive objectives to distance skipped tracks from session contexts, whereas Zhou et al. [51] (ECL-SR) introduces equivariant contrastive learning, distinguishing between invasive (e.g., item substitution) and mild augmentations to preserve user intent.

General frameworks further advance contrastive learning robustness. Liu et al. [20] proposes a debiased contrastive loss to address sampling bias in collaborative filtering, replacing traditional BPR with bias-corrected negative sampling. This complements GNN-based methods by reducing message dropout randomness through self-supervised embedding alignment.

2.6. Marked limits

Based on the extensive research conducted, we have summarized the following limitations that we aim to address in the development of our system:

1. High Computational Complexity
   (a) Scalability Challenges for Real-World Applications: Models such as PANE-GNN [21] and NFARec [35] use computationally intensive pairwise ranking and graph convolution mechanisms that struggle to scale efficiently for millions of users and items in industrial settings.
   (b) Trade-Offs in Resource Allocation: Approaches like SIGformer [5] heavily rely on dense embeddings and complex attention mechanisms, which may achieve marginal gains in accuracy at the expense of prohibitive hardware requirements.
2. Prediction Smoothing: GNN models, particularly deeper ones, are more likely to ”smooth out” predictions where the node representations become indistinguishable.
3. Accounting for All Negative Interactions
   (a) Imbalance-Induced Overfitting: Models like PANE-GNN [21] and NFARec [35] consider all negative interactions, which creates a skewed loss landscape. This leads to the model disproportionately focusing on predicting negatives rather than learning nuanced patterns for positive feedback.
   (b) Noise in Negative Interactions
   (c) Lack of Hierarchical Handling of Negatives: Few approaches implement strategies to categorize negative interactions based on severity or context, leading to simplistic modeling of negative signals. For instance, SIGformer [5] treat all negative edges uniformly without contextual granularity.
4. Static Graph Assumption
   (a) Inability to Handle Temporal Dynamics: Most models assume a static graph representation. This ignores dynamic user preferences and item popularity trends, making recommendations less accurate over time.
   (b) Over-Reliance on Offline Training: While approaches like KGUF [1] and NFARec [35] excel in modeling relationships with rich offline data, they lack mechanisms to adapt in near-real-time scenarios, leading to stale and less relevant recommendations.
5. In most of the studies, the researchers use data splits based on user activity shares rather than timestamps, which is mathematically inaccurate and can lead to data leakage.

3. Methodology

3.1. Heterogeneous Graph Convolution via `HeteroConv`

In real-world recommendation scenarios, user–item interactions come in multiple flavors (e.g. explicit likes, implicit views, negative signals). To capture the distinct semantics of each feedback type while allowing them to influence one another, we model the data as a heterogeneous graph and apply a two-stage, relation-specific message-passing scheme.

3.1.1 Graph Structure and Node Embeddings

We define three families of nodes:

$$\mathcal{V} = \{\underbrace{\texttt{user}}_{U}, \underbrace{\texttt{item}}_{I}, \underbrace{\texttt{feedback}_r}_{F_r:\, r\in R}\},$$

where each feedback type $r \in R$ (e.g. “implicit_positive”, “explicit_negative”) has its own copy of user–item interactions. Initial node features are learned embeddings:

$$\mathbf{h}_u^{(0)} = \mathbf{e}_{\text{user}}(u), \quad \mathbf{h}_i^{(0)} = \mathbf{e}_{\text{item}}(i), \quad \mathbf{h}_{f_r}^{(0)} = \mathbf{e}_{\text{fb}}^{(r)}(u),$$

where $u \in U$, $i \in I$, and each feedback node f_r is indexed by the user who generated it.

3.1.2 Relation-Specific Message Passing

For each directed relation $(s \xrightarrow{r} t)$, we associate a distinct graph convolution operator $\mathcal{M}^r_{s\to t}$ (e.g. GAT or SAGE). In a single layer, each target node $v \in \mathcal{V}_t$ receives messages from its neighbors of type s:

$$\mathbf{m}_v^{(r)} = \mathcal{M}^r_{s\to t}\big(\{\mathbf{h}_u : u \xrightarrow{r} v\}\big).$$

For example, items send embeddings to their feedback nodes via $\mathcal{M}^r_{I\to F_r}$, feedback nodes send back to items via $\mathcal{M}^r_{F_r\to I}$, and feedback nodes link to users via $\mathcal{M}^r_{F_r\to U}$.

NoConf #NoID

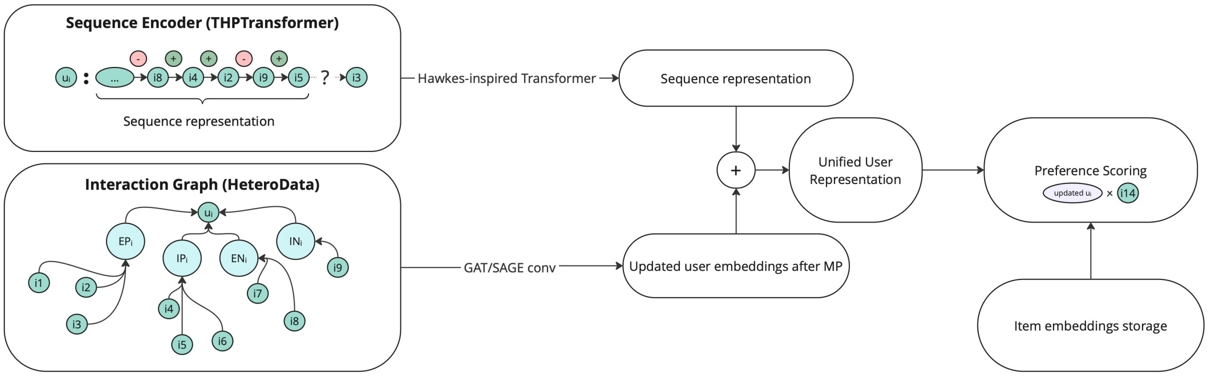

Figure 1. Proposed model

3.1.3 Aggregation Across Relations

After computing relation-specific messages, we aggregate them for each node type t. Let $R_{\text{in}}(t)$ be the set of incoming relations:

$$\mathbf{h}'_v = \text{AGG}\Big(\big\{\mathbf{m}_v^{(r)} : r \in R_{\text{in}}(t)\big\}\Big),$$

where AGG is a permutation-invariant function such as element-wise mean. In practice, we follow each aggregation with a nonlinearity, layer normalization, and dropout:

$$\mathbf{h}_v^{(1)} = \text{Dropout}\big(\text{LayerNorm}(\sigma(\mathbf{h}'_v))\big).$$

3.1.4 Multi-Hop Propagation

To allow information to flow multiple hops (e.g. user→item→feedback→user), we stack two heterogeneous convolution layers. Denoting the first-layer output by $\mathbf{h}^{(1)}$ and the second-layer output by $\mathbf{h}^{(2)}$, the final user embedding is

$$\mathbf{h}_u^{(\text{final})} = \mathbf{h}_u^{(2)}.$$

This two-stage design balances expressivity (capturing higher-order connectivity) with computational cost.

3.1.5 Rationale

- **Relation specificity:** Distinct $\mathcal{M}^r$ operators let the model learn separate transformations for e.g. positive vs. negative feedback.
- **Heterogeneous aggregation:** By aggregating messages from all relevant relations, each node's representation integrates complementary signals (item popularity, user sentiment, feedback type).
- **Multi-hop reasoning:** Two layers enable paths such as user→item→user via shared feedback nodes, enhancing collaborative filtering effects.

This `HeteroConv` architecture thus unifies diverse interaction types into a single embedding space, ready to be combined with sequence-based encoders for final recommendation and prediction tasks.“‘

3.2. Sequence Encoding via Hawkes-inspired Transformer

In many recommendation and user-behavior modeling tasks, a user's dynamic interaction history contains both the identity of items they consumed and the time at which each event occurred. To capture the evolving preferences and recency effects, we embed each event in both an item-space and a time-space, then feed the resulting sequence into a transformer augmented with Hawkes-like decay kernels. In this section, we describe this process in detail, explain the intuition behind each component, and present the final mathematical formulation.

3.2.1 Input Construction

Each user u is represented by a fixed-length sequence of past interactions, preprocessed as follows:

1. Sort all events of u by timestamp.
2. Truncate or left-pad the sequence to length $L+1$, reserving the first position for a special `[CLS]` token.
3. Record three tensors per user:
   - **Item IDs** $i_0, i_1, \ldots, i_L$, where $i_0 =$ `[CLS]` and subsequent i_k are actual item indices or padding.
   - **Event types** $e_0, e_1, \ldots, e_L$, similarly padded or set to a `[CLS]` code.
   - **Timestamps** $t_0, t_1, \ldots, t_L$ in seconds since epoch (with t_0 arbitrary).
   - **Mask** $m_k = 1$ if position k corresponds to a real interaction, else 0 for padding.

These arrays have shape of $[B, L+1]$ for batch size B. The mask ensures we never attend to padding positions.

3.2.2 Embedding Layer

We lift each discrete input into a D-dimensional space by three parallel embeddings:

- **Item embedding:** $\mathbf{v}_k = \text{Embitem}(i_k) \in \mathbb{R}^D$.
- **Positional embedding:** $\mathbf{p}_k = \text{Embpos}(k) \in \mathbb{R}^D$, to encode ordinal position in the window.
- **Temporal embedding:** $\boldsymbol{\tau}_k = W \cdot \text{time}(t_k) \in \mathbb{R}^D$, a linear projection of the scalar timestamp.

The three vectors are summed to produce the input to the transformer:

$$x_k = \mathbf{v}_k + \mathbf{p}_k + \boldsymbol{\tau}_k, k = 0, 1, \cdots, L.$$

This summation allows the model to treat content, position, and recency on an equal footing, and to learn how to combine them.

3.2.3 Hawkes-inspired Self-Attention

Standard self-attention computes, for each query position i and key position j, a compatibility score based on the inner product of normalized queries and keys. Here, we extend that mechanism with two multiplicative decay kernels:

1. Layer Normalization and Projections. We first apply layer normalization and split into queries, keys, and values:

$$\tilde{X} = \text{LayerNorm}([x_0, \cdots, x_L]),$$

$$\mathbf{Q} = \frac{\tilde{X}}{\sqrt{D}},$$

$$\mathbf{K} = \tilde{X},$$

$$V = \text{ELU}(X\tilde{W}_V)$$

This scaling of queries by $1/\sqrt{D}$ stabilizes gradients.

2. Raw Attention Scores. The basic dot-product attention score is

$$S_{i,j} =< Q_i K_j >$$

3. Temporal Decay Kernel. To model the intuition that more recent interactions should have greater influence, we compute the non-negative time difference

$$\Delta t_{i,j} = \max(0, t_i - t_j)$$

and multiply the raw score by an exponential decay:

$$S_{i,j} \leftarrow S_{i,j} e^{-\lambda_{decay} \Delta t_{i,j}}$$

where λ_{decay} is a learned or fixed hyperparameter controlling how quickly past events fade.

4. Positional Window Decay. We further enforce a locality bias by penalizing attention to distant positions in the sequence. Let $d_{i,j} = |i - j|$; then

$$\Psi_{i,j} = e^{-\lambda_{pos} d_{i,j}}, \lambda_{pos} = \frac{1}{W}$$

where W is the window size. We multiply again:

$$S_{i,j} \leftarrow S_{i,j} \Psi_{i,j}$$

This encourages the model to focus on nearby events unless there is a very strong signal.

5. Masking and Normalization. We set $S_{i,j} = -\infty$ whenever position j is padding (i.e. $m_j = 0$), ensuring that

$$A_{i,j} = \frac{e^{S_{i,j}}}{\sum_{k=0}^{L} e^{S_{i,j}}} \Rightarrow A_{i,j} = 0 \quad \text{if } S_{i,j} = -\infty.$$

6. Head Output. Finally, each head yields a weighted sum of the values:

$$O_{i,j} = \sum_{j=0}^{L} A_{i,j} V_j$$

and H parallel heads are summed to produce the multi-head output.

3.2.4 Residual and Feed-Forward Layers

The multi-head result is added back to the input via a residual connection,

$$X' = [x_0, \cdots, x_L] + O$$

followed by a two-layer position-wise feed-forward network, another residual addition, and a final layer normalization. The output is a contextual representation of each position that integrates item identity, position, and time-dependent recency.

3.2.5 Integration with Graph-based Static Embeddings

Before sequence encoding, users and items are also embedded by a heterogeneous GNN to capture long-term and social relations. The final recommendation score is computed by combining the static GNN-derived embedding of a user with the last position of the THPEncoder output, thereby blending global network structure with fine-grained temporal dynamics.

4. Experimental Results

4.1. Research Questions

To investigate whether negative feedback affects model training, we formulated the following questions:

1. **Q1:** If we augment the base interaction graph with special nodes that group feedback by category (explicit/implicit, positive/negative), will this improve performance metrics compared to a model without these special nodes?
2. **Q2:** Can we address the over-smoothing issue in GNNs by incorporating various mechanisms into the message-passing process?
3. **Q3:** Will adding a user-sequence processing component boost metrics and help determine the polarity of users' reactions to items?

4.2. Experimental Setup

We evaluated all models on three datasets: MovieLens[1] [19], Beauty[2], and Books[3]. We binarized ratings as follows:

1. **Negative feedback:** ratings < 4
2. **Positive feedback:** ratings ≥ 4

Moreover, we distinguished between explicit and implicit feedback:

1. **Explicit positive:** ratings $= 5$
2. **Explicit negative:** ratings ≤ 2
3. **Implicit positive:** ratings $= 4$
4. **Implicit negative:** ratings $= 3$

The datasets are described in Table 3.

4.3. Models

In our experimental study, we compare the following variants of heterogeneous GNN and sequence-aware models. Each method is trained with a Bayesian Personalized Ranking (BPR) loss unless stated otherwise.

- **BaseGAT**: A baseline graph-only model in which users and items share the same embedding space. Edges represent user–item interactions. Trained with BPR, using observed interactions as positives and randomly sampled non-interactions as negatives.
- **EmoGAT**: The heterogeneous feedback graph (implicit/explicit, positive/negative) encoded via two-layer `HeteroConv` with GATConv for each relation. Trained with BPR on explicit positive interactions; negatives are any items not interacted by the user.
- **diffGAT**: Same feedback-graph architecture as EmoGAT, but applies *signed aggregation*: positive feedback relations "attract" and negative relations "repel" user embeddings in each convolution layer before nonlinearity. Training protocol identical to EmoGAT.
- **EmoSAGE-THP**: Combines the feedback-graph encoder of EmoGAT (with SAGEConv instead of GATConv) and the Hawkes-inspired Transformer (THP) sequence encoder. No auxiliary polarity task; trained end-to-end with BPR loss.
- **EmoSAGE-THP+use_negatives**: Builds on EmoSAGE-THP, but replaces random negative sampling in BPR with true user–specific negatives (both implicit and explicit negative feedback).
- **EmoGAT-THP+extra_task**: Extends EmoGAT by adding the THP sequence encoder and an auxiliary head to predict the polarity (positive vs. negative) of the next interaction. Optimizes joint BPR and cross-entropy losses.
- **diffGAT-THP+extra_task**: Mirrors EmoGAT-THP+extra_task but uses the signed-aggregation GNN (diffGAT) in place of EmoGAT for the graph encoder. All other components and training losses remain the same.

5. Results Analysis

Table 1 reports our key findings:

- **Best performance by signed-aggregation + THP.** The **diffGAT-THP+extra_task** model achieves the highest *Recall@10*, *NDCG@10*, and *MAP@10* across all datasets, indicating that explicitly modeling both feedback polarity and temporal sequence yields the strongest recommendations.
- **Limited gains from explicit vs. implicit separation alone.** Contrary to expectations, **EmoGAT-THP+extra_task** does not improve over **EmoSAGE-THP** on any metric. This suggests that simply distinguishing explicit positive interactions without a signed-aggregation mechanism is insufficient to boost ranking quality.
- **Evidence for negative-aware modeling.** The superior performance of **diffGAT** (graph only) and **diffGAT-THP+extra_task** (graph + sequence) over their unsigned counterparts demonstrates that directly encoding negative feedback as "repulsive" signals materially improves user representation and downstream recommendation.
- **Need for polarity-sensitive evaluation.** Although our primary metrics focus on ranking accuracy (Recall, NDCG, MAP), they do not capture whether recommended items align positively or negatively with user sentiment. Future work should introduce evaluation measures that explicitly assess the *polarity* of recommendations—i.e. how well they match inferred user preferences

[1] https://grouplens.org/datasets/movielens/
[2] https://snap.stanford.edu/data/amazon/productGraph/categoryFiles/reviews_Beauty_5.json.gz
[3] https://cseweb.ucsd.edu/~jmcauley/datasets/goodreads.html

NoConf
#NoID

Table 1. Performance comparison of recommender models for Leave-10-Last-out dataset with seed = 42.

Dataset	Metric	**Ours**							Improv.	p-value
		BaseGAT	EmoGAT	diffGAT	EmoSAGE-THP	EmoSAGE-THP+use_negatives	EmoGAT-THP+extra_task	diffGAT-THP+extra_task		
MovieLens	Recall@10	0.02427	0.03328	0.03587	0.03955	0.008414	0.03664	0.04001	-% *	-
	NDCG@10	0.02269	0.03476	0.03597	0.04175	0.008563	0.03865	0.04	-% *	-
	MAP@10	0.006801	0.01238	0.01254	0.01555	0.002689	0.01386	0.01477	-% *	-
Beauty	Recall@10	0.001108	0.001155	0.0006838	0.01749	0.001985	0.01715	0.01593	-% *	-
	NDCG@10	0.001008	0.001158	0.0006371	0.01863	0.001954	0.01817	0.01638	-% *	-
	MAP@10	0.0002644	0.0003405	0.0001757	0.007262	0.0005849	0.007466	0.006174	-% *	-
Books	Recall@10	0.0005425	0.007282	0.00772	0.01122	0.0005366	0.009887	0.01157	-% *	-
	NDCG@10	0.0004682	0.007337	0.007872	0.01152	0.0005239	0.01039	0.0118	-% *	-
	MAP@10	0.000117	0.002269	0.002422	0.003625	0.0001532	0.003297	0.003719	-% *	-

Table 2. Statistics for the datasets showing the number of users, items, interactions, positive and negative interactions, as well as the ratio of positives to negatives.

Dataset	Users	Items	Inter.	Pos.	Neg.	P/N
MovieLens	6040	3706	1000209	575281	424928	1.3538
Beauty	22363	12101	198502	154272	44230	3.488
Books	18892	25475	1378033	883573	494460	1.7869

Table 3. Types of splits and their sizes for the datasets. The numbers are reported before any dataset preprocessing. For the time-based split, the 0.9 quantile was used.

Type of Split	Dataset	Train	Test
Leave-N-last-out	MovieLens	939809	60400
	Beauty	156082	42420
	Books	1226673	151360
Time	MovieLens	900188	100021
	Beauty	178674	19828
	Books	1240229	137804

beyond mere click-through likelihood.

References

[1] Salvatore Bufi, Alberto Carlo Maria Mancino, Antonio Ferrara, Daniele Malitesta, Tommaso Di Noia, and Eugenio Di Sciascio. Kguf: Simple knowledge-aware graph-based recommender with user-based semantic features filtering. In *International Workshop on Graph-Based Approaches in Information Retrieval*, pages 41–59. Springer, 2024. 4

[2] Xuheng Cai, Chao Huang, Lianghao Xia, and Xubin Ren. Lightgcl: Simple yet effective graph contrastive learning for recommendation. *arXiv preprint arXiv:2302.08191*, 2023. 2

[3] Jianxin Chang, Chen Gao, Yu Zheng, Yiqun Hui, Yanan Niu, Yang Song, Depeng Jin, and Yong Li. Sequential recommendation with graph neural networks. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, pages 378–387, 2021. 2

[4] Hao Chen, Yuanchen Bei, Qijie Shen, Yue Xu, Sheng Zhou, Wenbing Huang, Feiran Huang, Senzhang Wang, and Xiao Huang. Macro graph neural networks for online billion-scale recommender systems. In *Proceedings of the ACM web conference 2024*, pages 3598–3608, 2024. 2

[5] Sirui Chen, Jiawei Chen, Sheng Zhou, Bohao Wang, Shen Han, Chanfei Su, Yuqing Yuan, and Can Wang. Sigformer: Sign-aware graph transformer for recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1274–1284, 2024. 1, 3, 4

[6] Yongjun Chen, Zhiwei Liu, Jia Li, Julian McAuley, and Caiming Xiong. Intent contrastive learning for sequential recommendation. In *Proceedings of the ACM web conference 2022*, pages 2172–2182, 2022. 2

[7] Shansan Gong and Kenny Q Zhu. Positive, negative and neutral: Modeling implicit feedback in session-based news recommendation. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*, pages 1185–1195, 2022. 1, 3

[8] Bowen Hao, Jing Zhang, Hongzhi Yin, Cuiping Li, and Hong Chen. Pre-training graph neural networks for cold-start users and items representation. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 265–273, 2021. 2

[9] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 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*, pages 639–648, 2020. 1, 2

[10] Yunfan Hu, Zhaopeng Qiu, and Xian Wu. Denoising neural network for news recommendation with positive and negative implicit feedback. *arXiv preprint arXiv:2204.04397*, 2022. 3

[11] Junjie Huang, Ruobing Xie, Qi Cao, Huawei Shen, Shaoliang Zhang, Feng Xia, and Xueqi Cheng. Negative can be positive: Signed graph neural networks for recommendation. *Information processing & management*, 60(4):103403, 2023. 1, 3

[12] Weiming Huang, Baisong Liu, and Zhaoliang Wang. A metric learning perspective on the implicit feedback-based recommendation data imbalance problem. *Electronics*, 13(2): 419, 2024. 2

[13] Jipeng Jin, Zhaoxiang Zhang, Zhiheng Li, Xiaofeng Gao,

Xiongwen Yang, Lei Xiao, and Jie Jiang. Pareto-based multi-objective recommender system with forgetting curve. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pages 4603–4611, 2024. 2

[14] Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*, pages 197–206. IEEE, 2018. 2

[15] Jiacheng Li, Yujie Wang, and Julian McAuley. Time interval aware self-attention for sequential recommendation. In *Proceedings of the 13th international conference on web search and data mining*, pages 322–330, 2020. 2

[16] Longbin Li, Chao Zhang, Sen Li, Yun Zhong, Qingwen Liu, and Xiaoyi Zeng. Graph contrastive learning with multi-objective for personalized product retrieval in taobao search. *arXiv preprint arXiv:2307.04322*, 2023. 3

[17] Muyang Li, Zijian Zhang, Xiangyu Zhao, Wanyu Wang, Minghao Zhao, Runze Wu, and Ruocheng Guo. Automlp: Automated mlp for sequential recommendations. In *Proceedings of the ACM web conference 2023*, pages 1190–1198, 2023. 2

[18] Weibin Liao, Yifan Zhu, Yanyan Li, Qi Zhang, Zhonghong Ou, and Xuesong Li. Revgnn: Negative sampling enhanced contrastive graph learning for academic reviewer recommendation. *ACM Transactions on Information Systems*, 43(1): 1–26, 2024. 3

[19] Zihan Lin, Changxin Tian, Yupeng Hou, and Wayne Xin Zhao. Improving graph collaborative filtering with neighborhood-enriched contrastive learning. In *Proceedings of the ACM web conference 2022*, pages 2320–2329, 2022. 2, 7

[20] Zhuang Liu, Yunpu Ma, Yuanxin Ouyang, and Zhang Xiong. Contrastive learning for recommender system. *arXiv preprint arXiv:2101.01317*, 2021. 4

[21] Ziyang Liu, Chaokun Wang, Jingcao Xu, Cheng Wu, Kai Zheng, Yang Song, Na Mou, and Kun Gai. Pane-gnn: Unifying positive and negative edges in graph neural networks for recommendation. *arXiv preprint arXiv:2306.04095*, 2023. 1, 3, 4

[22] Yingtao Luo, Qiang Liu, and Zhaocheng Liu. Stan: Spatio-temporal attention network for next location recommendation. In *Proceedings of the web conference 2021*, pages 2177–2185, 2021. 2

[23] Fuyuan Lyu, Yaochen Hu, Xing Tang, Yingxue Zhang, Ruiming Tang, and Xue Liu. Towards automated negative sampling in implicit recommendation. *arXiv preprint arXiv:2311.03526*, 2023. 1, 3

[24] Jianxin Ma, Chang Zhou, Hongxia Yang, Peng Cui, Xin Wang, and Wenwu Zhu. Disentangled self-supervision in sequential recommenders. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 483–491, 2020. 2

[25] M Jeffrey Mei, Oliver Bembom, and Andreas F Ehmann. Negative feedback for music personalization. In *Proceedings of the 32nd ACM Conference on User Modeling, Adaptation and Personalization*, pages 195–200, 2024. 1, 3

[26] Eunkyu Oh and Taehun Kim. Tempgnn: Temporal graph neural networks for dynamic session-based recommendations. *arXiv preprint arXiv:2310.13249*, 2023. 2

[27] Yunzhu Pan, Nian Li, Chen Gao, Jianxin Chang, Yanan Niu, Yang Song, Depeng Jin, and Yong Li. Learning and optimization of implicit negative feedback for industrial short-video recommender system. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 4787–4793, 2023. 1, 3

[28] Shuo Qin, Feng Lin, Lingxiao Xu, Bowen Deng, Siwen Li, and Fangcheng Yang. Multi-behavior session-based recommendation via graph reinforcement learning. In *Asian Conference on Machine Learning*, pages 1119–1134. PMLR, 2024. 2

[29] Pavan Seshadri, Shahrzad Shashaani, and Peter Knees. Enhancing sequential music recommendation with negative feedback-informed contrastive learning. In *Proceedings of the 18th ACM Conference on Recommender Systems*, pages 1028–1032, 2024. 4

[30] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 1441–1450, 2019. 2

[31] Dongjing Wang, Xin Zhang, Zhengzhe Xiang, Dongjin Yu, Guandong Xu, and Shuiguang Deng. Sequential recommendation based on multivariate hawkes process embedding with attention. *IEEE transactions on cybernetics*, 52(11):11893–11905, 2021. 2

[32] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. Kgat: Knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 950–958, 2019. 1, 2

[33] Xiang Wang, Yaokun Xu, Xiangnan He, Yixin Cao, Meng Wang, and Tat-Seng Chua. Reinforced negative sampling over knowledge graph for recommendation. In *Proceedings of the web conference 2020*, pages 99–109, 2020. 3

[34] Xinfeng Wang, Fumiyo Fukumoto, Jin Cui, Yoshimi Suzuki, Jiyi Li, and Dongjin Yu. Eedn: Enhanced encoder-decoder network with local and global context learning for poi recommendation. In *Proceedings of the 46th international ACM SIGIR conference on research and development in information retrieval*, pages 383–392, 2023. 2

[35] Xinfeng Wang, Fumiyo Fukumoto, Jin Cui, Yoshimi Suzuki, and Dongjin Yu. Nfarec: A negative feedback-aware recommender model. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 935–945, 2024. 3, 4

[36] Yueqi Wang, Yoni Halpern, Shuo Chang, Jingchen Feng, Elaine Ya Le, Longfei Li, Xujian Liang, Min-Cheng Huang, Shane Li, Alex Beutel, et al. Learning from negative user feedback and measuring responsiveness for sequential recommenders. In *Proceedings of the 17th ACM Conference on Recommender Systems*, pages 1049–1053, 2023. 3

[37] Muyan Weng, Yunjia Xi, Weiwen Liu, Bo Chen, Jianghao Lin, Ruiming Tang, Weinan Zhang, and Yong Yu. Beyond

positive history: Re-ranking with list-level hybrid feedback. *arXiv preprint arXiv:2410.20778*, 2024. 3

[38] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. Self-supervised graph learning for recommendation. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, pages 726–735, 2021. 2

[39] Yiqing Wu, Ruobing Xie, Zhao Zhang, Xu Zhang, Fuzhen Zhuang, Leyu Lin, Zhanhui Kang, and Yongjun Xu. Dfgnn: Dual-frequency graph neural network for sign-aware feedback. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3437–3447, 2024. 3

[40] Lianghao Xia, Chao Huang, Yong Xu, Jiashu Zhao, Dawei Yin, and Jimmy Huang. Hypergraph contrastive collaborative filtering. In *Proceedings of the 45th International ACM SIGIR conference on research and development in information retrieval*, pages 70–79, 2022. 2

[41] Xu Xie, Fei Sun, Zhaoyang Liu, Shiwen Wu, Jinyang Gao, Jiandong Zhang, Bolin Ding, and Bin Cui. Contrastive learning for sequential recommendation. In *2022 IEEE 38th international conference on data engineering (ICDE)*, pages 1259–1273. IEEE, 2022. 2

[42] Jingcao Xu, Chaokun Wang, Cheng Wu, Yang Song, Kai Zheng, Xiaowei Wang, Changping Wang, Guorui Zhou, and Kun Gai. Multi-behavior self-supervised learning for recommendation. In *Proceedings of the 46th international ACM SIGIR conference on research and development in information retrieval*, pages 496–505, 2023. 3

[43] Junliang Yu, Hongzhi Yin, Min Gao, Xin Xia, Xiangliang Zhang, and Nguyen Quoc Viet Hung. Socially-aware self-supervised tri-training for recommendation. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pages 2084–2092, 2021. 2

[44] Junliang Yu, Hongzhi Yin, Xin Xia, Tong Chen, Lizhen Cui, and Quoc Viet Hung Nguyen. Are graph augmentations necessary? simple graph contrastive learning for recommendation. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*, pages 1294–1303, 2022. 2

[45] Junliang Yu, Xin Xia, Tong Chen, Lizhen Cui, Nguyen Quoc Viet Hung, and Hongzhi Yin. Xsimgcl: Towards extremely simple graph contrastive learning for recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 36(2):913–926, 2023. 2

[46] Wenhui Yu, Zixin Zhang, and Zheng Qin. Low-pass graph convolutional network for recommendation. In *Proceedings of the AAAI conference on artificial intelligence*, pages 8954–8961, 2022. 2

[47] Peng Yuan, Haojie Li, Minying Fang, Xu Yu, Yongjing Hao, and Junwei Du. Amplify graph learning for recommendation via sparsity completion. *arXiv preprint arXiv:2406.18984*, 2024. 2

[48] Zhongping Zhang, Yin Jia, Yuehan Hou, and Xinlu Yu. Explicit behavior interaction with heterogeneous graph for multi-behavior recommendation. *Data Science and Engineering*, 9(2):133–151, 2024. 3

[49] Sen Zhao, Wei Wei, Ding Zou, and Xianling Mao. Multi-view intent disentangle graph networks for bundle recommendation. In *Proceedings of the AAAI conference on artificial intelligence*, pages 4379–4387, 2022. 2

[50] Xiangyu Zhao, Liang Zhang, Zhuoye Ding, Long Xia, Jiliang Tang, and Dawei Yin. Recommendations with negative feedback via pairwise deep reinforcement learning. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1040–1048, 2018. 2

[51] Peilin Zhou, Jingqi Gao, Yueqi Xie, Qichen Ye, Yining Hua, Jaeboum Kim, Shoujin Wang, and Sunghun Kim. Equivariant contrastive learning for sequential recommendation. In *Proceedings of the 17th ACM conference on recommender systems*, pages 129–140, 2023. 4