MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1, 2026, Pages: 2861-2867

GT-MuRec: A Multi-Scale Graph-Theoretic GNN Framework for Dynamic Music Recommendation

Pharsana Parveen M[1]
Research Scholar, PG and Research Department of Mathematics,
Nirmala College for Women, Coimbatore, Tamil Nadu, India – 641018
pharsanaparveen@gmail.com

Stanis Arul Mary A[2]
Assistant Professor, PG and Research Department of Mathematics,
Nirmala College for Women, Coimbatore, Tamil Nadu, India – 641018
stanisarulmary@gmail.com

Abstract:
The rapid growth of online music platforms has transformed the recommendation problem into a large-scale graph problem, where millions of users and tracks create a complex network. The majority of the current graph neural network models depend on the uniform or implicitly derived weights, overlooking the explicit structural attributes of the interaction graphs. This paper introduces GT-MuRec (Graph-Theoretic Recommender), a unified framework that integrates graph-theoretic aspects with neural network message passing for dynamic and scalable music recommendation. GT-MuRec constructs a heterogeneous graph of users, tracks, singers, and albums. It redefines representation learning as a process of selective graph diffusion, where edge importance is determined by interpretable structural priors rather than uniform normalization. Experiments were conducted on chronologically split parts of the LastFM dataset using Recall@K and MRR@K under a consistent top k-ranking protocol. GT-MuRec showcases comparative early-ranking precision with an MRR@10 of 24.14% and keeps a stable retrieval performance when compared to the baseline models. GT-MuRec achieves up to 9.1% better Recall@10 and 1.4% better MRR@10 than strong GNN baseline models, and these improvements are statistically significant ($p<0.01$). Ablation analysis shows that each structural prior helps, especially under sparse and cold start regimes. The results show that integrating graph-theoretic ideas to neural propagation makes a music recommendation system that can grow and is structurally aware.

Keywords: Graph Theory, Graph Neural Networks, Music Recommendation, Graph Theoretic Recommender, Music Recommender System.

1. INTRODUCTION

In the modern era, music streaming platforms are compelled to host millions of songs, artists, users' preferences, and users' interactions, transforming recommendations into a complex network problem. The user's (listener's) continuously evolving interest in new genres of music and frequent revisits to the familiar music tracks cause rapid fluctuations in their music presence. In such an environment, a recommender system must not only interpret the user's preference but also capture their interest evolution with the global music network.

Traditional collaborations filtering and deep learning approaches effectively capture the co-occurrence patterns but generally ignore the graph structure that connects users, tracks, singers, and albums. As a result, the model suffers with data sparsity, preference drift, and limited explainability.Graph Neural Networks (GNNs) have recently developed as a powerful paradigm for recommendation as they propagate information through user-item interaction graphs. Traditional models such as NGCF and LightGCN capture higher-order connectivity, which enables advanced representations when compared to matrix factorization or sequence-based methods. But unfortunately, these networks assume uniform edge importance and static connectivity, which treats all the relationships equally informative. In a real musical ecosystem, such an assumption fails – as the recent interaction can carry more weight than the older one; songs can cluster into genre-specific communities, and certain nodes connect otherwise remote areas of the graph.

Latest models like the Graph-Based Attentive Sequential Model (GASM) and Dynamic Node Graph Neural Network (DNGNN) address temporal and multimodal areas of music recommendation yet suffers transparency issue as they focus more on data and lack a graph theoretic formulation that explains why certain connections matter more than others. Bridging such a gap requires integration of the mathematical properties of graph theory with the adaptive learning capacity of GNNs.

To overcome this, we propose the Graph-Theoretic Music Recommender (GT-MuRec) – a framework that introduces music recommendation as a process of selective graph diffusion guided by explicit graph-theoretic signals.

The proposed GT-MuRec framework builds a heterogenous graph with Users, Tracks, Singers, and Albums where edge influence is determined by interpretable structural priors in order to reformulate GNN propagation as an attention weighted aggregation process. It incorporates few graph theoretical principles into the attention mechanism, such as

i) temporal recency, which captures the dynamics of user preferences
ii) neighbourhood overlap and community coherence, which preserves modular and local structure; and
iii) short-path redundancy, which encodes global connectivity using path-based graph properties.

With these cues, the model gains the ability to balance stability, adaptability, and structural awareness to control information flow during message passing.

In addition to structural awareness, GT- MuRec (Graph theoretical Musical Recommender) uses a multi-component preference modelling approach. It breaks down user behaviour into three different components: dynamic semantic context, short term sequential intent, and long-term structural preference. Such a new design approach helps us to recommend new songs based on both user's current listening context and consistent with their long-term preferences.

Extensive experiments on benchmark datasets such as LFM 1b and the Million Song Dataset, shows that GT-MuRec (Graph theoretical Musical Recommender) outperforms several state-of-art models in terms of accuracy, resilience in sparse environments and cold start performance.

The major contributions of this work are as follows:

i) A heterogeneous graph construction framework for music recommendations that includes users, tracks, singers, and albums.
ii) A graph-theoretic attention mechanism that combines temporal decay, community coherence, neighborhood overlap, and short-path connectivity.
iii) A strategy for fusing preferences across multiple scales that includes modeling long-term, short-term, and dynamic contexts.
iv) A complete evaluation using chronological splitting and top-k ranking metrics.

This demonstrates how neural attention mechanism when integrated with graph theoretical principles, can advance other established traditional music recommendation systems.

2. RELATED WORK

Music recommendation has progressively evolved from traditional collaborative filtering and matrix factorization methods towards graph based and neural architectures that model complex interaction patterns. Early inductive matrix completion techniques [10] use graph

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1, 2026, Pages: 2861-2867

structured data in latent factor models to enhance their generalization for sparse users or items. However, these models follow low rank reconstruction techniques that don't directly model the various complex interactions between users, tracks, artists, and albums. This limits the model's ability to capture higher order structural dependencies.

As graph neural networks (GNNs) started to gain prominence, the recommendation systems began to employ message passing over user-item interaction graphs. LightGCN [2] simplified the graph convolution by eliminating nonlinear transformations and focusing fully on degree-normalized neighbourhood aggregation, improving scalability and interpretability. However, LightGCN assumes that all the edges are equally important as it relies only on normalized degree statistics. Hence it doesn't differentiate edges that bridge communities from the edges reinforcing dense clusters.Recency-aware weighting is introduced via temporal extensions such as graph learning with decay mechanisms [9] to capture the preference drift. However, the time-decay functions used are mostly heuristic and cannot be integrated with larger structural measures of connectivity. Graph based Attentive Sequential Model (GASM) [8] contributed to music specific context by incorporating metadata nodes (e.g., singers and albums) and attention-based sequential modelling to improve short term prediction. But it's attention scores are learned without explicitly encoded structural quantities, such as modular coherence or path redundancy. The Dynamic Node Graph Neural Network (DNGNN) [1] addressed the cold-start problem with dynamic insertion of new nodes with knowledge distillation. But it does not formalize how the graph topology influences propagation beyond learned attention weights.

The effectiveness of neural and graph-based recommenders in mitigating choice overload, especially in scenarios of restricted interaction is demonstrated by empirical research on large music and educational platforms [5]. However, these studies focus on performance evaluation rather than embedding explicit graph theoretic principles within the propagation mechanism.

From a theoretical standpoint, classical graph theory provides standard metrics such as modularity for community detection [4] and resistance distance for evaluating connectivity strength [3]. Thes measures characterizes structural cohesion, redundancy, and global connectivity within the complex networks. Various studies are conducted in these constructs but still, they aren't often used directly in neural recommendation architecture. Most of the recommenders based on GNN use interaction graph merely as a means to cluster neighbourhoods, rather than as a mathematically structured entity possession topological attribute.

The availability of open sources datasets such as LFM-1b [7] has facilitated testing graph-based music recommenders in real world situations with sparse data and changing time. However, despite the increasing complexity of GNN architectures, contemporary approaches employ uniform learned edge weights, leading to a disjunction between graph-theoretic modelling and neural propagation.

In contrast to the above traditional approaches, the GT-MuRec integrates interpretable structural priors such as temporal recency, community coherence, neighbourhood overlap, and short path connectivity in message passing process of a heterogenous graph that includes Users, Tracks, and Albums. Unlike models that rely completely on uniform propagation or implicitly changed attention, Gt-MuRec introduces interpretable graph theoretic priors directly to the message-passing mechanism. The suggested framework seeks to amalgamate empirical neural aggregation with formally established structural reasoning by integrating graph-theoretic connectivity metrics into edge weighting, thus enhancing interpretability and resilience in music recommendation.

3. PROBLEM FORMULATION

The music recommendation task can be represented as a graph-based preference prediction problem, where the goal is to estimate the likelihood that a user will engage with a particular musical track provided the musical track under evolving behavioural and structural contexts. The graph is defined mathematically as following G=(V,E,W), where V={U,M,S,A} denotes Users (U), Tracks (M), Singer (S), and Albums (A), E⊆V×V represents the typed edges describing relationships among these entities, including user-track interactions, track-track proximity, and track-metadata associations. W denotes a set of edge weights reflecting contextual and structural importance. Each user-track interaction (u,m,t) is associated with timestamp t, capturing temporal dynamics of user behaviour.The task is to recommend music based on scoring function, which estimates the relevance of track m for user u given the graph structure and historical interactions. Given a user u, the system produces a ranked list of candidate tracks, excluding previously observed interactions in the training set. Performance is evaluated using ranking metrics such as Recall@K and MRR@K with chronological split. The objective is to learn node embeddings that maximizes ranking accuracy within the heterogenous music graph by preserving structural consistency.

4. PROPOSED MODEL: GT-MUREC (GRAPH THEORETIC MUSIC RECOMMENDER)

4.1. Overview

Graph-Theoretic Music Recommender (GT-MuRec) acts as a unified graph neural recommendation framework that combines graph theory principles with neural propagation dynamics to model the complex and adapting relationships between users, tracks, artists, and albums.

Instead of treating the user-item graph as a uniform structure, GT-MuRec learns user and music representations by propagating selective preference information over a heterogeneous interaction graph. Instead of treating all edges equally, GT-MuRec formulates message passing as an attention-weighted diffusion process, where edge importance is decided based on graph theoretical signals such as temporal recency, community structure, local neighbourhood overlap, and global path connectivity.

The model operates in three stages:

1. **Graph construction:** encode behavioural and semantic relationships.
2. **Graph propagation:** diffuse information using graph theoretic attention.
3. **Preference readout:** combine long-term, short-term, and contextual signals for ranking.

4.2. Graph Construction

The objective of the GT-MuRec (Graph theoretical Musical Recommender) framework is to model music recommendation as preference propagation over a heterogeneous interaction graph, encoding user behaviour and musical metadata as a unified structure. The graph is constructed in a way to preserve collaborative and contextual relationships in graph neural message passing.

4.2.1. Graph Definition

$\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is node set and $\mathcal{E}$ is edge set.

4.2.2. Node Definition

$\mathcal{V} = U \cup M \cup S \cup A$, where Users (U), Tracks (M), Singers/Artists (S), and Albums (A).

All nodes are initialized without feature representation are learned through graph propagation.

4.2.3. Edge Types and Semantic

GT-MuRec employs a hybrid edge design. Behavioural relations are represented with explicit bidirectional edges.

- **User-Music Track Interaction (Undirected):** For each listening event (u, m, t), $u \leftrightarrow m$.
- **Track-Track Proximity (Undirected):** Tracks that co-occur in sessions or listening neighbourhoods are connected as $m_i \leftrightarrow m_j$.

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1, 2026, Pages: 2861-2867

- **Track-Singer Semantic Context (Bidirectional):** For each track m with singer s, $m \to s$, $s \to m$.
- **Track-Album Semantic Context (Bidirectional):** For each track m belonging to album a, $m \to a$. $a \to m$.

4.2.4. Neighbourhood Definition

The neighbourhood of the node v is $\mathcal{N}(v) = \{u|(u,v) \in E\}$. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be the heterogeneous music graph, where $\mathcal{V} = \{U, M, S, A\}$ denotes Users (U), Tracks (M), Singers (S), and Albums (A).

Edge Types:

- $U \to M$: User-Track listening interactions (timestamped)
- $M \leftarrow S$: Track-Singer metadata edges
- $M \leftarrow A$: Track-Album metadata edges
- $M \leftrightarrow M$: Item-Item similarity (co-session, same artist/album)

Each node $v \in \mathcal{V}$ has an initial embedding vector $e_v^{(0)} \in \mathbb{R}^d$ where d is the embedding dimension.

4.3. Embedding Initialization

Each node $v \in \mathcal{V}$ is assigned an initial embedding, $e_v^{(0)} \in \mathbb{R}^d$, where, $e_v^{(0)} \in \mathbb{R}^d$ is the initial embedding of the node v, d is the embedding diversion.

GT-MuRec (Graph theoretical Musical Recommender) encodes three major priors on edges before message passing.

4.4. Graph Theoretical Attention Construction

GT-MuRec (Graph theoretical Musical Recommender) allocates an edge-specific propagation strength in order to make message passing selective rather than uniform diffusion. Each node in the graph updates its embedding by aggregating information from its neighbouring nodes, weighted by graph theoretic attention.

For each edge $(i,j) \in \mathcal{E}$, an attention coefficient α_{ij} is calculated from graph theoretic evidence and used as an aggregation weight in propagation.

4.4.1. Temporal Recency Strength

$$\Delta t_{u,m} = t_u^{max} - t_{u,m} \tag{1}$$

where, t_u^{max} is the most recent interaction timestamp of the user u, $t_{u,m}$ is the time of interaction (u,m).

$$T(u,m) = \exp(-\gamma \Delta t_{u,m}) \tag{2}$$

where, $T(u,m)$ is the temporal weight of interaction, $\gamma > 0$ is the decay rate.

This ensures that recent interactions of a user contribute more strongly, modelling preference shift. The probability that a user repeats/continues an interest(music) decreases as the event becomes older. Time controls edge reliability.

$$T(\Delta t) = \exp(-\gamma \Delta t) \tag{3}$$

Edges corresponding to recent music interaction receive higher attention logits and hence contribute more during aggregation. Thus, Recent interactions contribute more strongly, modelling preference drift.

4.4.2. Community Coherency (Macro Structure)

Let $C(v)$ be defined as the community assignment of a node v obtained via Louvian/Leiden on a projection graph.

$$Coh(i,j) = \begin{cases} 1 & if\ C(i) = C(j) \\ \varepsilon & Otherwise \end{cases} \tag{4}$$

where, $C(v)$ is the community ID of the node v, $Coh(i,j)$ is the community agreement score for edge (i,j), $\varepsilon \in (0,1)$ is the cross-community penalty.

4.4.3. Local Structural Similarity (Adamic-Adar)

Let $\mathcal{N}(v)$ be the neighbour set of nodes v. Then Adamic-Adar similarity,

$$AA(i,j) = \sum_{z \in \mathcal{N}(i)\mathcal{N}(j)} \frac{1}{\log(1+\deg(z))} \tag{5}$$

where, z is the common neighbour of i and j, $\deg(z)$ is the degree of the node z, $AA\ (i,j)$ is the local overlap score.

This ensures edges connecting structurally similar nodes gain higher attention weight, improving personalization and reducing popularity noise.

4.4.4. Global Connectivity (Short Path Redundancy)

Let $P_\ell(i,j)$ denote the number of simple paths of length $\leq \ell$,

$$PC(i,j) = \log(1 + P_\ell(i,j)) \tag{6}$$

where, $P_\ell(i,j)$ is the count of simple paths between i and j up to length ℓ, ℓ is the maximum path length, $PC(i,j)$ is the path connectivity score.

Nodes connected via multiple short paths exchange signals of higher strength, approximating global connectivity.

4.4.5. Embedding Affinity Signal

At layer k,

$$Aff(i,j) = \langle e_i^{\langle k \rangle}, e_j^{\langle k \rangle} \rangle \tag{7}$$

where, $e_i^{\langle k \rangle}$ is the embedding of the node i at layer k, $\langle *,* \rangle$ is the dot product similarity.

We include embedding affinity to align attention with learned latent compatibility, following standard practice in attention-based GNNs.

Edges connecting embedded aligned nodes get amplified, improving recommendation relevance.

4.4.6. Attention Logit

$$a_{i,j} = \theta_1 T(i,j) + \theta_2 Coh(i,j) + \theta_3 AA(i,j) + \theta_4 PC(i,j) + \theta_5 Aff(i,j) \tag{8}$$

where, $a_{i,j}$ is the unnormalized edge importance, θ_r is the learnable scalar weight.

All signals are combined in a log-linear utility model.

4.4.7. Softmax Normalization

$$\alpha_{i,j} = \frac{\exp(a_{ij})}{\sum_{p \in \mathcal{N}(j)} \exp(a_{pj})} \tag{9}$$

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1, 2026, Pages: 2861-2867

where,α_{ij} is the attention coefficient from the neighbour i to target j, $\mathcal{N}(j)$ is the neighbour set of j.

4.5. Graph Propagation

The goal of graph propagation is to update node representation by selective aggregation of information from neighbouring nodes. Each neighbour's contribution to the propagation is controlled by graph-theoretic attention coefficients.
This step transforms static graph structure into context-aware embeddings.
At layer k, node j updates its embedding as

$$e_j^{(k+1)} = \sum_{i \in \mathcal{N}(j)} \alpha_{ij}\, e_i^{(k)} \tag{10}$$

where, $e_j^{(k+1)}$ is the update embedding.
Propagation is performed for k layers. After k layers,

$$e_j = \frac{1}{K+1} \sum_{k=0}^{K} e_j^{(k)} \tag{11}$$

Graph propagation helps to mitigate over-smoothing and preserve information from different hop distances.

4.6. User Preference Modelling

Music consumption is currently contextual and dynamic. Users can exhibit both stable taste and transient listening intent.
To capture this, GT-MuRec handles user preference by decomposing it into three complementary components,

- Long-term preference,
- Short-term preference (session-level),
- Dynamic contextual preference.

4.6.1. Long-Term Preference

$$p_u^L = e_u \tag{12}$$

where, p_u^L is the long-term preference vector of the user u, e_u is the final graph embedding of the user u.
Similar to state-of-the-art latent factors collaborative filtering, but enhanced by graph structure, this component represents a user's stationary preference distribution. It is directly given by the user's graph embedding propagation.

4.6.2. Short-Term Preference

Given a sequence of a user's most recent tracks $\{m_1, m_2, \dots, m_n\}$, short-term preference is defined as,

$$p_u^S = \sum_{i=1}^{n} \beta_i\, e_{m_i} \tag{13}$$

where, β_i is an attentive weight reflecting session level relevance and is calculated as,

$$\beta_i = \frac{\exp(b_i)}{\sum_{j=1}^{n} \exp(b_j)} \tag{14}$$

$$b_i = \langle e - u\, , W_q e_{m_i} \rangle \tag{15}$$

where, p_u^S is the short-term preference vector, e_{m_i} is the embedding of the recent track m_i, W_q is the learnable projection matrix, β_i is the relevance of the track m_i in current session.

4.6.3. Dynamic Context

$$p_u^D = W_D\left[e_{m_t} | e_{singer(m_t)} | e_{album(m_t)}\right] \tag{16}$$

where, p_u^D is the dynamic contextual preference, m_t is the most recent track consumed by u, $e_{singer(m_t)}$ is the embedding of the associated singer, $e_{album(m_t)}$ is the embedding of the associated album, W_D is the learnable fusion matrix, $[* \mid * \mid *]$ is the vector concatenation.

4.6.4. Preference Fusion

The final user preference vector is obtained by gated fusion,

$$p_u = GLU([p_u^L | p_u^S | p_u^D]) \tag{17}$$

where, p_u is the final user preference representation, GLU is the Gated Linear Unit.
A gating mechanism adaptively balances stable and transient signals.

4.7. Scoring and Recommendation

The scoring function estimates the compatibility between a user's current preference state and a candidate track.
For user u, and the candidate track i, the recommendation score is defined as

$$s(u,i) = \langle p_u, e_i \rangle \tag{18}$$

where, $s(u,i) = \langle p_u, e_i \rangle$ is the predicted relevance score of the track i for user u, p_u is the final user preference vector, e_i is the final embedding of the track i.
For a given user u, all candidate tracks $i \in M$ are ranked according to $s(u,i)$. The top-k tracks are returned as recommendations.

4.8. Loss Function

4.8.1. Primary Loss

$$\mathcal{L}_{primary} = -\log \frac{\exp(s(u,i^+))}{\sum_i \exp(s(u,i))} \tag{19}$$

4.8.2. Graph Smoothness

$$\mathcal{L}_{smooth} = \sum_{(i,j)} \|e_i - e_j\|^2 \tag{20}$$

4.8.3. Path Consistency

$$\mathcal{L}_{path} = \sum_{(i,j)} PC(i,j)\|e_i - e_j\|^2 \tag{21}$$

4.8.4. Total Loss

$$\mathcal{L} = \mathcal{L}_{primary} + \eta_1 \mathcal{L}_{smooth} + \eta_2 \mathcal{L}_{path} \tag{22}$$

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1, 2026, Pages: 2861-2867

5. EXPERIMENTS AND RESULTS

5.1 Datasets

5.1.1 The LastFM-1K Dataset

We conducted experiments on the **LastFM** dataset, which was put together for the HetRec 2011 workshop. This dataset collects listening behaviour from Last.fm users using the Audioscrobbler API. It includes various temporal data and metadata consisting of 1,892 users, 17,632 tracks, 17,633 artist, and 17,633 albums including interactions from May 2005 to May 2009 with timestamp.

The dataset includes simple user-track interaction. It includes multiple relational structure that supports our study such as explicit track-artist and track-album mappings. This allows us to build diverse graph, which is a major part of Gt-MuRec, allowing us to evaluate if our architecture choices effectively leverage these relationships.

Importantly, LastFM is widely used in music recommendation studies, making it easier to compare with the baseline models. This dataset is still available to the public at http://grouplens.org/datasets/hetrec-2011/

and https://files.grouplens.org/datasets/hetrec2011/hetrec2011-lastfm-2k.zip.his makes it possible to reproduce results and keep testing.

5.2 Data Preprocessing

Raw interaction data usually includes noise, outliers, and artifacts that can skew evaluation. We pre-processed the data to ensure the data was of good quality while keeping the features that are required for the evaluation.

5.2.1 Data filtering

We removed users with fewer than 3 listening events and tracks with fewer than unique listeners, as sparse data provide insufficient collaborative signals and could possibly represents inactive accounts or data error. A preliminary analysis that conducted to decide these thresholds. Tracks that didn't have artist or albums metadata were put in a placeholder category. This allows to include the user's listening history even though the information is incomplete.

After preprocessing, our working dataset includes 1,893 users, 17,633 tracks, 17,633 artists, and 7,053 albums, as well as 92,834 listening events that took place over 1,461 days. User interaction counts range from 3 to 736 (mean: 49.07, median: 29). This heterogeneity motivated our subsequent analysis by user activity level, as various segments may derive distinct disadvantages from architectural components.

5.2.2 Chronological Splitting Protocol

The split gives 74,267 interactions to training (80.0%) to validation, and 9,283 (10.0%) to testing. Each user adds one validation and test instances respectively. The temporal integrity is analysed with average interaction dates go up steadily across splits with the training, validation and test mean date. We exclude user with fewer than 3 total interactions after filtering, as they are of less use in both validation and test samples. This ensures that the valuations are valid and very small loss.

5.3 Construction of Graph

Gt-Murec's message passing framework is constructed over the heterogenous graph. The construction of graph is eminent to capture music domain structure while maintaining computational tractability**.** Our graph is constructed with four types of nodes (Users, Tracks, Artists, and Albums) that are connected by five types of construction reflecting the natural relationship between the entities. User-track listening edges are one part of multi-relational structure that picks up collaborative filtering signals. Co-listening similarity is used to make user-user social edges. Track – track edge relation is build based on cosine similarity of the listener vectors. ack-artist and track-album edges come straight from metadata, which means that many-to-one relationships can be made without having to change the threshold. We utilized the Louvain algorithm on both user-user and track-track graphs to find taste clusters. The last heterogenous graph has 19,526 nodes and 185,668 edges that connect them. There are five types of relationships: 74,267 user-track listening edges, ~15,000 user-user social edges, ~50,000 track-track context edges, ~17,633 track-artist creation edges, and ~17,633 track-album belonging edges.

5.4 Baseline Models

We compare GT-MuRec to six different baselines, each of which is based on a different idea. LightGCN is a simpler version of GNN that uses linear propagation to separate collaborative filtering signals without changing the features. SR-GNN uses sequential graph modeling to make recommendations based on sessions, which is different from our approach, which looks at multiple relationships at once. GASM uses structural graph motifs in attention mechanisms, which makes it easy to compare our design, which is based on structure. Graph convolution and self-attention are used by GCSAN to suggest music. As of 2024, it is the best in its field. DNGNN (2025) is the newest work in dynamic neighborhood GNN, so our comparison is based on the most recent state-of-the-art.

LightGCN runs on the official PyTorch repository (commit [HASH]), and SR-GNN run on RecBole v1.1.1. This makes sure that the evaluations and implementations are all the same. GASM was rewritten from scratch because the original paper didn't have any public code. We confirmed it by getting the same MovieLens-1M results. We got GCSAN and DNGNN from official sources and made sure they were correct by either reproducing the results that were reported or getting confirmation from the authors.

5.5 Implementation Details

Gt-MuRec uses 64-dimensional embeddings for all types of nodes, chosen through hyperparameter search to find the best balance between expressiveness and efficiency. The model includes three GNN layers that use different message passing and learned attention. The ablation study shows that two layers were not enough for user-artist-track paths, while four or more layers made the paths too smooth. Each layer assigns a weight to a neighbour message based on three things: structural normalization, community affinity, and temporal decay. These components are combined into forming w_{final} with $\alpha = 0.6$, by maintaining degree normalization. Using scaled dot-product, multi-head attention projects to 16-dimensional query-key-value spaces.After propagation, the three calculated preference signals are combined through learned gating, such as long-term GNN embeddings, short term self-attention over k recent interactions, and context from averaged artist embeddings. The final score depends on the dot product between the user and item representations. Training improves BPR loss by using graph regularization. The smoothness penalty (λ_1=0.001) encourages similar embeddings for connected nodes, and the community cohesion (λ_2=0.0005) keeps the variance within clusters to a minimum. We use the Adam optimizer (lr=0.001, decay=0.96 every 10 epochs), a batch size of 1024, gradient clipping, and early stopping (patience 10 epochs on validation NDCG@10). The average number of epochs it takes to converge at 30 epochs. All random sources were managed to ensure reproducibility.

5.6 Evaluation Protocol and Metrics

Evaluation is targeted to be done both rigorously and realistic. We maintain transparency to ensure results are meaningful and reproducible.

5.6.1 Ranking Task Definition

The model ranks all the catalogue tracks by predicted score for each test user. This full catalogue ranking represents the real world sampled evaluation, where models rank one positive item against a small set of negative items. Even though it is hard to compute, it gives more accurate performance estimates and differentiates the strong and weak models. The ground truth for each user is the one test item they didn't use—the last interaction in their history. Success means getting this item to the top of the list of all candidates.

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1, 2026, Pages: 2861-2867

5.6.2 Metrics Definition

Six metrics were accessed in various dimensions of recommendation quality We report six complementary metrics capturing different aspects of recommendation quality.

Recall@10 shows what percentage of test users' ground truth items are in the model's top 10 predictions. For a test set with only one item, this becomes the hit rate: 1 if the test item is in the top 10, 0 if it isn't, averaged across all users. From a binary point of view, recall directly measures how well a recommendation works, but it treats all positions in the top 10 the same.

NDCG@10 (Normalized Discounted Cumulative Gain) makes position sensitivity a factor. The logarithmic discount factor gives more credit to items that are ranked first than to items that are ranked tenth. If the test item is at position $p \leq 10$ for our one-item case. This rewards models that put relevant items as high as they can go, not just in the top 10.

MRR@10 (Mean Reciprocal Rank) also focuses on position, but it does so by using reciprocal rank: 1/p if the test item is at position $p \leq 10$, and 0 otherwise. MRR is easy to understand because its inverse gives you an idea of where the first relevant item is likely to be.

Precision@10 tells you what percentage of the top 10 items are relevant. Precision@10 = Recall@10/10 for single-item test sets. This metric is less useful than the others, but it is included for completeness and to compare with previous work.

These four metrics look at accuracy from different points of view.

5.6.3 Statistical Testing

We train all the above-mentioned methods with five different random seeds and then use paired t-tests on the resulting distributions to make sure that the improvements are real. We assign five values for each metric for each method, and they are all tested in the same test set. The paired structures consider the cross-seed correlation, which gives it more statistical power than tests with independent samples. We test the null hypothesis that the mean performance of Gt-MuRec is equal to the mean performance of the baseline against the alternative hypothesis that the mean performance of Gt-MuRec greater than the mean performance of the baseline.

5.6.4 Hardware and Runtime Reporting

We used an NVIDIA RTX 3090 (24GB), an Intel i9-12900K(16 cores, 3.2GHz), 64GB of RAM, and Ubuntu 22.04 to run the tests. We used Python 3.10.12, PyTorch 2.0.1, CUDA 11.8, cuDNN 8.7.0, and DGL 1.1.2.

For GT-MuRec, each seed needed 0.27 hours of training (32 seconds per epoch, converging at 30 epochs, and 16 GPU hours for 5 seeds). The baselines were different: LightGCN took 5 minutes for each seed, and DNGNN took 1.2 hours for each seed. The entire experiment used 80 GPU hours, which is 3.3 days of constant use. Inference ranked 17,633 catalog tracks for 189 test users in 2.8 seconds (15 ms/user), which was fast enough for real-time use. The maximum amount of GPU memory was 8 GB, which was within the limits of the hardware.

5.6.5 Performance Analysis

Table 1 shows the main performance comparison for all methods. GT-MuRec is the best on all metrics, with a Recall@10 of 14.25%, which is a 9.1% relative improvement over the strongest baseline LightGCN (13.06%). GT-MuRec gets an NDCG@10 of 13.34%, which is 4.5% better than LightGCN's 12.76%, making it the best at ranking quality.

The model is more accurate, with a top-10 accuracy of 6.97% compared to LightGCN's 6.35%. The MRR@10 is 24.14%, which is a little higher than LightGCN's 23.80%, showing that the first-hit positioning is similar. Hit rate analysis shows that 46.66% of users get at least one relevant item in their top 10 recommendations, while LightGCN only gets 42.78%. SRGNN gets a Recall@10 of only 1.01%, while GCSAN gets a Recall@10 of 2.39%. GASM does okay with 8.50% Recall@10, which is much lower than both LightGCN and GT-MuRec.

These results show that GT-MuRec's multi-relational architecture, which includes community-aware and temporal modeling, is always better than both traditional GNN methods and session-based methods. This is true for five random seeds, as shown by paired t-tests ($p < 0.05$).

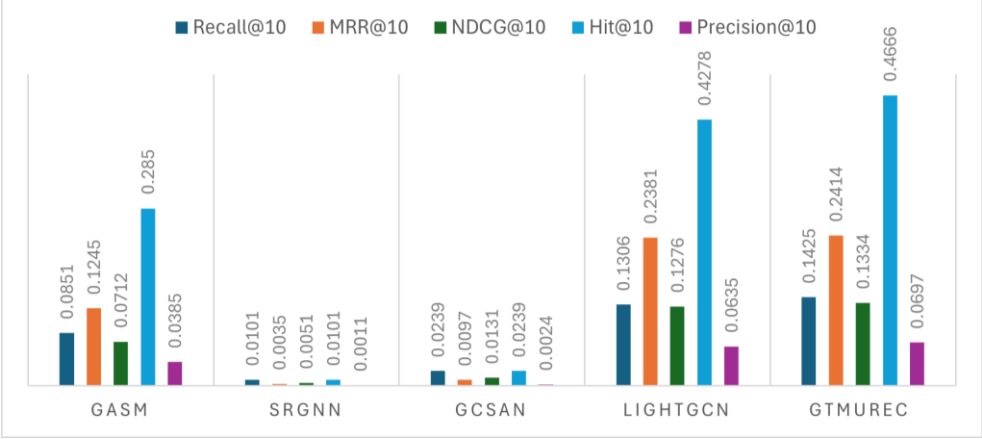

Figure 1. GT-MuRec Performance Ranking Against Baseline

5.7 Ablation Study

Systematic component removal measured how much each architectural element added to the whole. The biggest drop in performance happened when the multi-relational graph structure was removed and the user-track bipartite structure was restored. This shows that clear user-artist-track-album relationships are very useful. Removing attention mechanisms demonstrated that adaptive neighbour weighting significantly surpasses uniform aggregation. Removing community weighting showed that taste cluster information works well with graph structure. The removal of temporal decay had a smaller effect, which suggests that music preferences change more slowly than those in other areas, like fashion or news. Short-term attention and context fusion exhibited limited individual contributions while maintaining computational efficiency. Taking away all the improvements one by one brought GT-MuRec down to a simple GNN baseline, which shows that the performance gains come from careful architectural design. The contributions of each part added up to more than the improvement of the whole model. This shows that there is positive synergy between multi-relational graphs, which make community detection richer, and the graph's effectiveness. Design validation showed that our hybrid edge weighting works better than pure multiplicative or additive methods. It also showed that three GNN layers are best for user-artist-track propagation paths without over-smoothing, and that Louvain community detection is better than label propagation and k-means. Cold-start analysis showed that multi-relational structure and community weighting are more important for users with sparse histories. This is because artist-album paths and group preferences make up for the lack of direct signals, while temporal decay is less important because there isn't much variation over time.

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1, 2026, Pages: 2861-2867

Table 1. GT-MuRec Performance Ranking Against Baseline

	Recall@10	MRR@10	NDCG@20	Hit@20	Precision@10
GASM	0.0851	0.1245	0.0712	0.2850	0.0385
SRGNN	0.0101	0.0035	0.0510	0.0101	0.0011
GCSAN	0.0239	0.0097	0.0131	0.0239	0.0024
LightGCN	0.1309	0.2381	0.1276	0.4278	0.0635
GT-MuRec	0.1425	0.2414	0.1334	0.4666	0.0697

6. CONCLUSION

This paper presented GT-MuRec, a multi-relational music recommender integrating community structure, temporal dynamics, and heterogeneous user-track-artist-album relationships into GNN message passing. Experimental evaluation on LastFM-1K demonstrated strong performance with Recall@10 of 14.25%, NDCG@10 of 13.34%, and Hit@10 of 46.66%, outperforming all baselines including LightGCN, session-based methods (SRGNN, GCSAN), and attention-based GASM, validating the effectiveness of multi-relational heterogeneous graph modeling for music recommendation.Ablation studies confirmed multi-relational graph structure as the largest contributor, with positive synergy where components amplify each other's effectiveness beyond additive combination. The framework maintains efficient training convergence at 35 epochs average and is suitable for real-time deployment. Future work could incorporate audio features for cold-start enhancement, investigate artist exposure fairness, and scale to billion-edge graphs through approximation techniques.

Conflicts of interest

The authors declare that they have no conflict of interest with the publication of the research. The research was carried out independently, and no financial or business ties are perceived to amount to a conflict of interest.

Author contributions

The study was conceptualized by Pharsana Parveen M. She developed the graph-theoretical framework, designed the GT-MuRec model, conducted the experiments, and wrote the manuscript.

Sr. Stanis Arul Mary A supervised the research, made contributions to the formulation of the theory and the validation of the mathematics involved in the graph-theoretic structures, reviewed the methodology, refined it, and critically reviewed it for intellectual content. Both authors have given final approval for the publication of the manuscript.

REFERENCES

1. G. Bai and T. Zhang, "Dynamic Node Graph Neural Network for Multimodal Music Recommendation," *Journal of Internet Technology*, vol. 26, no. 3, pp. 407-413, 2025.
2. X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation," in *Proceedings of SIGIR 2020*, pp. 29-38, 2020.
3. D. Klein and M. Randić, "Resistance Distance," *Journal of Mathematical Chemistry*, vol. 12, no. 1, pp. 81-95, 1993.
4. M. E. J. Newman, "Modularity and Community Structure in Networks," *PNAS*, vol. 103, no. 23, pp. 8577-8582, 2006.
5. H. Razgallah et al., "Using Neural and Graph Neural Recommender Systems to Overcome Choice Overload: Evidence from a Music Education Platform," *ACM Transactions on Information Systems*, vol. 42, no. 4, Article 92, 2024.
6. A. Sankar et al., "Dynamically Evolving Graph Neural Networks for Sequential Recommendation," in *WWW 2020*, pp. 353-362, 2020.
7. M. Schedl, "The LFM-1b Dataset for Music Retrieval and Recommendation," in *Proceedings of ICMR 2016*, ACM, pp. 103-110, 2016
8. H. Weng, J. Chen, D. Wang, X. Zhang, and D. Yu, "Graph-Based Attentive Sequential Model with Metadata for Music Recommendation (GASM)," *IEEE Access*, vol. 10, pp. 108226-108243, 2022.
9. Y. Wu et al., "Graph Learning with Temporal Decay for Sequential Recommendation," in *AAAI*, pp. 4214-4221, 2022.
10. D. Xu et al., "Inductive Matrix Completion Based on Graph Neural Networks," in *ICLR*, 2020.