Metadata-Gated Dual-Channel Graph Neural Networks for Fine-Grained Recommendation Personalization

Noorbasha Zareena[1], B Tarakeswararao[2]

[1]Research Scholar, Department of Computer Science and Systems Engineering, Andhra University, Visakhapatnam, AP, India

[2]Department of Computer Science and Engineering, Kallam Haranadhareddy Institute of Technology, Guntur, Andhra Pradesh, India

HOW TO CITE:

Noorbasha Zareena, B Tarakeswararao (2026). Metadata-Gated Dual-Channel Graph Neural Networks for Fine-Grained Recommendation Personalization. International Journal of Special Education, 41(16s), 980-998

COPYRIGHT STATEMENT:

Copyright: © 2026 Authors. Open access publication underthe terms and conditions of the Creative CommonsAttribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).

ABSTRACT:

Graph neural networks (GNNs) have become a dominant paradigm for recommendation, yet most existing metadata-aware variants treat side information as an auxiliary feature that is simply concatenated onto node embeddings before standard message passing. This design implicitly assumes that metadata is always available, equally reliable across users and items, and equally relevant at every propagation step assumption that break down under sparse profiles, incomplete attributes, and cold-start conditions. This paper addresses that gap by proposing a Metadata-Gated Dual-Channel Graph Neural Network (MG-DGNN) that separates structural (interaction-based) and semantic (metadata-based) propagation into two explicit channels and fuses them through a learnable, per-node gating function conditioned on metadata confidence, trained jointly with a hybrid objective combining Bayesian Personalized Ranking (BPR) loss with a metadata-alignment contrastive term. We position MG-DGNN relative to GraphSAGE, Graph Attention Networks (GAT), LightGCN, and knowledge-graph attention models (KGAT), and evaluate it on MovieLens-1M (6,040 users, 3,416 items after 5-core filtering) under a leave-one-out protocol with HR@K, NDCG@K, Precision@K, MAP, MRR, coverage, a four-way ablation study, sensitivity analysis over graph depth, embedding dimension, and metadata neighbourhood size, a stratified cold-start evaluation, and a metadata-masking robustness test. The gated dual-channel design significantly outperforms single-channel structural propagation (paired t-test, $p < 0.01$ across all reported ranking metrics) and shows a clear benefit specifically for cold-start users (HR@10 0.467 vs. 0.367 for matrix factorization on a 30-user cold stratum), consistent with its design intent. It does not, however, show a statistically significant aggregate improvement over plain matrix factorization at the training budget used here, and the ablation study does not confirm a measurable benefit from the metadata-alignment contrastive term in a single run. We report these results without adjustment, including the ones that do not favor the proposed method, and identify the specific further evidence multi-seed ablation runs, a second dataset, and evaluation beyond the roughly 9,500-node scale tested

here needed to establish the framework's practical benefit more conclusively.

Keywords: Graph Neural Networks; Metadata-Gated Fusion; Dual-Channel Message Passing; Recommendation Systems; Cold-Start Personalization; Scalability.

1. Introduction

Recommendation systems are now a core component of digital platforms, shaping how users discover products, media, and information across e-commerce, entertainment, education, and healthcare. As user expectations shift toward highly personalized and context-aware suggestions, classical collaborative filtering (CF) and matrix factorization methods, while simple and effective, remain limited by data sparsity, cold-start behaviour, and an inability to model higher-order relationships between users and items (Fan et al., 2019; Wang et al., 2019).

Graph neural networks (GNNs) address several of these limitations by representing users, items, and their interactions as a structured graph and propagating information across edges to capture multi-hop connectivity (Wang et al., 2019; Zhao et al., 2019). Foundational architectures such as GraphSAGE (Hamilton et al., 2017) introduced inductive neighbourhood sampling and aggregation, while Graph Attention Networks (GAT) (Veličković et al., 2018) introduced learnable attention weights over structural neighbours. LightGCN (He et al., 2020) later showed that removing feature transformation and non-linear activation from graph convolution simplifies training without sacrificing accuracy, and knowledge-graph-based models such as KGAT (Wang et al., 2019) and DSKReG (Wang et al., 2021) fuse explicit knowledge-graph triples with attention-based propagation to support explainable recommendation. More recently, graph transformer architectures have been proposed to replace local message passing with global self-attention over graph tokens, improving long-range dependency modelling at the cost of quadratic attention complexity (Chen et al., 2023).

Despite this progress, a common limitation persists across these lines of work: metadata demographic attributes, item categories, temporal context, and other side information is typically injected once, at node initialization, and then propagated using the same structural mechanism used for interaction signals. GraphSAGE and GAT were not designed with heterogeneous metadata in mind and treat all neighbour features identically after initial embedding. LightGCN deliberately removes feature transformation altogether, which improves efficiency but forecloses any mechanism for weighting metadata reliability during propagation. KGAT and related knowledge-graph models require metadata to be expressed as explicit relational triples, which is not always available and adds substantial graph-construction overhead. None of these approaches provide a mechanism for the model to recognize when metadata for a given node is sparse, noisy, or missing, and to adjust its reliance on structural versus metadata signals accordingly. This is precisely the condition under which cold-start users and newly listed items occur most often, which is also where metadata-aware methods are expected to provide the greatest benefit.

This paper targets that specific gap. Rather than proposing another way to concatenate metadata into node features, we introduce a Metadata-Gated Dual-Channel Graph Neural Network (MG-DGNN) that explicitly separates structural propagation (over the interaction graph) from semantic propagation (over metadata similarity) and learns a per-node gate that controls how much each channel contributes to the final embedding, conditioned on the confidence and completeness of the node's own metadata. This is combined with a hybrid loss that adds a metadata-alignment contrastive term to the standard BPR ranking objective, encouraging embeddings of nodes with similar metadata to be drawn closer even when direct interaction evidence is limited.

The contributions of this work are as follows: (1) we identify and articulate a specific architectural gap in existing metadata-aware GNN recommenders the absence of a reliability-aware fusion mechanism between structural and metadata signals and position it relative to GraphSAGE, GAT, LightGCN, KGAT, and graph transformer baselines; (2) we design a gated dual-channel message-passing architecture and a hybrid ranking/contrastive loss that directly addresses this gap, together with a complexity analysis of the added components; (3) we specify a comprehensive evaluation protocol that extends beyond RMSE, coverage, and latency to include Precision@K, Recall@K, MAP, MRR, ablation studies isolating each architectural component, sensitivity analysis over graph depth, embedding dimension, and metadata subsets, and statistical significance testing; and (4) we discuss the practical conditions cold-start, sparse interactions, incomplete metadata, dynamic graph evolution, and larger-scale deployment under which the proposed design is expected to help, and identify which of these claims still require empirical validation at the scale used in production recommender systems.

The remainder of the paper is organized as follows. Section 2 critically reviews related work and situates the proposed framework relative to existing metadata-aware and graph-based recommenders. Section 3 presents the MG-DGNN architecture, including its dual-channel propagation rule, gating function, hybrid loss, and computational complexity. Section 4 describes the experimental protocol, datasets, baselines, and evaluation metrics. Section 5 reports results, including ablation and sensitivity analyses. Section 6 discusses practical limitations and deployment considerations, and Section 7 concludes the paper.

2. Literature Review

Recommendation research has moved through several distinct paradigms, each addressing limitations of its predecessor while introducing new trade-offs. This section reviews that progression critically, with an emphasis on where existing graph-based and metadata-aware methods fall short, rather than simply cataloguing prior contributions.

Early recommender systems relied on collaborative filtering (CF), inferring preferences from historical user–item interactions (Herlocker et al., 1999). CF is simple and requires no content information, but suffers from data sparsity and cold-start problems when users or items have few recorded interactions (Koren et al., 2009). Content-based filtering mitigated sparsity by using descriptive attributes of users and items (Pazzani & Billsus, 2007), but tends to over-specialize recommendations within previously consumed content. Deep learning approaches, including neural collaborative filtering and attention-based sequence models, improved accuracy on large-scale data (Zhang et al., 2022) but generally treat interactions as an unstructured set rather than exploiting the relational structure connecting users, items, and their neighbours.

Recognizing that user–item interactions naturally form a bipartite graph, subsequent work applied graph neural networks to propagate information along interaction edges. GraphSAGE (Hamilton et al., 2017) introduced inductive sampling and aggregation over local neighbourhoods, enabling generalization to unseen nodes. GAT (Veličković et al., 2018) introduced learned attention coefficients over structural neighbours, allowing the model to weight some interactions more heavily than others, though the attention is computed purely from node embeddings rather than from explicit metadata reliability. Fan et al. extended graph-based propagation to social recommendation by incorporating user relationship graphs, and LightGCN (He et al., 2020) demonstrated that removing feature transformation and non-linear activation from graph convolution simplifies training while preserving or improving accuracy, at the cost of removing any mechanism for feature-level reasoning during propagation. Heterogeneous graph neural networks (Shi et al., 2019) generalize this further to multiple node and edge types but typically require the relation schema to be fixed in advance.

A parallel line of work incorporates metadata explicitly. Knowledge-graph-based methods connect users, items, and attributes through semantic relations, enabling explainable recommendation (Wang et al., 2019). KGAT and DSKReG combine knowledge graph triples with attention or differentiable sampling mechanisms, while KGIN and FF-KGAT model user intent and fused features, respectively (Huang et al., 2021; Tao et al., 2022; Wang et al., 2019; Wang et al., 2021).These methods demonstrate that metadata can meaningfully improve accuracy, but they generally require metadata to already be expressed as relational triples, and they apply the same propagation and attention mechanism to metadata-derived edges as to interaction edges that is, metadata reliability is not modelled separately from structural connectivity. Surveys of GNN-based recommendation (Wu et al., 2023) similarly note that most metadata-aware designs treat metadata as a static input feature rather than as a signal whose trustworthiness can vary across nodes and needs to be learned

More recent work replaces local neighbourhood aggregation with global self-attention over the full graph, allowing long-range dependencies to be modelled without deep stacks of propagation layers (Chen et al., 2023). While effective for capturing distant structural relationships, graph transformer attention is computed over token/node embeddings and inherits the same limitation as GAT: it does not distinguish between the reliability of structural evidence and the reliability of metadata evidence for a given node, and its quadratic attention cost raises additional scalability concerns for graphs with hundreds of thousands of nodes.

Table 1 summarizes this comparison. Across GraphSAGE, GAT, LightGCN, KGAT-style knowledge-graph models, and graph transformers, no method models metadata confidence as a first-class, learnable signal that adaptively controls the balance between structural and metadata-derived propagation on a per-node basis. This is the specific architectural gap the present work targets: rather than injecting metadata once at initialization or fusing it through the same attention mechanism used for structural neighbours, MG-DGNN maintains metadata propagation as a separate channel and learns when to trust it. This design is intended to be most useful precisely where existing metadata-aware GNNs are weakest: cold-start users, sparsely profiled items, and heterogeneous or partially missing metadata conditions under which uniform fusion is most likely to introduce noise rather than signal.

Table 1. Comparative positioning of MG-DGNN against representative graph-based recommendation architectures

Method	Metadata handling	Attention basis	Reliability-aware fusion	Known limitation
CF / MF [10],[11]	None (interactions only)	N/A	N/A	Sparsity, cold-start
GraphSAGE [15]	Concatenated at init.	Uniform / mean aggregation	No	No metadata weighting
GAT [24]	Concatenated at init.	Learned, structure-based	No	Attention reflects structure, not metadata trust
LightGCN [17]	Not modelled separately	None (linear propagation)	No	No feature-level reasoning
KGAT / DSKReG [19],[7]	Explicit KG triples required	Learned, relation-based	No	Requires pre-built KG; uniform trust across edges

Graph Transformers [25]	Concatenated as token feature	Learned, global self-attention	No	Quadratic cost; no metadata-confidence signal
MG-DGNN (proposed)	Separate metadata channel	Learned dual-channel gate	Yes – per-node, confidence-conditioned	Requires empirical validation at scale (Sec. 5–6)

This research therefore builds on the demonstrated value of both graph-based propagation and metadata integration, but reframes the problem as one of adaptive trust between two information sources rather than feature concatenation. The following section formalizes this architecture.

3. Proposed Method: MG-DGNN

This section presents the Metadata-Gated Dual-Channel Graph Neural Network (MG-DGNN). The design goal is to let the model learn, per node, how much to rely on metadata-derived signal versus structural interaction signal, rather than fusing the two uniformly as in prior metadata-aware GNNs (Section 2).

Let $U = \{u_1, u_2, \ldots, u_m\}$ be the set of users and $I = \{i_1, i_2, \ldots, i_n\}$ the set of items. The interaction data forms a bipartite graph $G = (V, E)$, $V = U \cup I$, $E \subseteq U \times I$, where each edge $(u, i) \in E$ denotes an observed interaction (rating, click, or purchase).

Each user u and item i is associated with a metadata feature vector $x_u \in R^{d_u}$ and $x_i \in R^{d_i}$ respectively, drawn from demographic attributes, item categories, temporal context, or text-derived embeddings. In addition, we define a metadata confidence score $c_v \in [0, 1]$ for each node v, computed from the proportion of non-missing metadata fields and, where available, their recency (Equation 6). This confidence score is the key input that existing metadata-aware GNNs (Section 2) do not model.

The learning objective is to estimate a preference score $\hat{y}_{ui} = f(u, i \mid G, x_u, x_i, c_u, c_i)$ that ranks items for each user.

3.1 Dual-Channel Node Representation

Unlike single-channel GNNs that merge structural and metadata information at initialization, MG-DGNN maintains two parallel embeddings for every node throughout propagation: a structural embedding, updated over the interaction graph, and a semantic embedding, updated over a metadata-similarity graph constructed by connecting nodes whose metadata cosine similarity exceeds a threshold τ.

$$h_v^{S,(0)} = W_S^{(0)} x_v,$$

$$h_v^{M,(0)} = W_M^{(0)} x_v,$$

where W_S and W_M are separate learnable projection matrices for the structural and metadata channels. Maintaining two initial projections rather than a single shared embedding allows the network to learn representations specialized to each propagation regime before they are fused.

3.2 Structural Channel Propagation

The structural channel propagates over interaction edges using additive attention over neighbours in N(v), following the general GAT-style formulation but restricted to the interaction graph:

$$h_v^{S,(l+1)} = \sigma\left(\sum_{w \in N(v)} \beta_{vw}^{(l)} W_S^{(l)} h_w^{S,(l)}\right)$$

$$e_{vw}^{(l)} = \text{LeakyReLU}\left(a^\top \left[W_S^{(l)} h_v^{S,(l)} \| W_S^{(l)} h_w^{S,(l)}\right]\right)$$

where ‖ denotes concatenation and a is a learnable attention vector. This channel captures collaborative signal in the same way as conventional GNN recommenders.

Metadata Channel Propagation

The metadata channel propagates over the metadata-similarity graph, weighting neighbours by metadata similarity rather than structural co-occurrence:

$$h_v^{M,(l+1)} = \sigma\left(\sum_{w \in N_M(v)} \alpha_{vw}^{(l)} W_M^{(l)} h_w^{M,(l)}\right)$$

$$\alpha_{vw}^{(l)} = \frac{\exp\ (sim(x_v, x_w))}{\sum_{z \in N_M(v)} \exp\ (sim(x_v, x_z))}$$

with sim(·,·) a cosine similarity function. This is the channel most directly comparable to prior metadata-aware attention (e.g., KGAT), but it is kept structurally separate from the interaction channel rather than merged into a single propagation step.

3.3 Metadata-Aware Gating Fusion

The central architectural contribution of this work is the fusion step. At each layer, a gate $g_v^{(l)} \in$ [0,1] is computed from the node's metadata confidence score and the divergence between its two channel representations, and used to combine them:

$$g_v^{(l)} = \sigma\left(W_g\left[h_v^{S,(l)} \| h_v^{M,(l)} \| c_v\right] + b_g\right)$$

$$h_v^{(l+1)} = g_v^{(l)} \odot h_v^{S,(l+1)} + \left(1 - g_v^{(l)}\right) \odot h_v^{M,(l+1)}$$

where $\odot$ denotes elementwise multiplication and c_v is the node's metadata confidence score. When c_v is low — for a cold-start user with a mostly empty profile, for instance — the gate learns to suppress the (unreliable) metadata channel and rely more heavily on structural propagation, and vice versa for well-profiled, sparsely-interacted nodes. This is the mechanism that Section 2 identifies as absent from GraphSAGE, GAT, LightGCN, and KGAT-style fusion, all of which apply a fixed or purely structure-conditioned combination rule regardless of metadata completeness.

The metadata confidence score itself is computed once as a preprocessing step:

$$c_v = \frac{1}{D}\sum_{k=1}^{D} I\ (field_k(\text{v}) is observed) \cdot \omega_k(v)$$

where D is the number of metadata fields tracked for node v, and recency_weight_k downweights stale attributes (e.g., an outdated demographic entry) toward zero over time; it defaults to 1 for static fields.

Aggregation, Prediction, and Hybrid Loss

After L layers, the fused representations across layers are combined with LightGCN-style residual weighting to obtain the final embedding:

$$h_u = \sum_{l=0}^{L} \theta^{(l)} h_u^{(l)} \qquad h_i = \sum_{l=0}^{L} \theta^{(l)} h_i^{(l)}$$

and the predicted preference score is computed by dot product, $\hat{y}_{ui} = \sigma(h_u \text{^T}\ h_i)$, following the same prediction layer used in conventional GNN recommenders so that any accuracy differences can be attributed to the propagation and fusion design rather than to the scoring function.

The model is trained with a hybrid loss that combines the standard Bayesian Personalized Ranking (BPR) objective with a metadata-alignment contrastive term:

$$L_{BPR} = -\sum_{(u,i,j) \in D} \ln \sigma\left(\hat{y}_{ui} - \hat{y}_{uj}\right)$$

$$L_{meta} = \sum_{v \in V}\left[\sum_{w \in P(v)} \max(0, m - sim(h_v, h_w)) + \sum_{w \in N(v)} \max(0, sim(h_v, h_w) - m')\right]$$

$$L = L_{BPR} + \lambda_1 L_{meta} + \lambda_2 \| \Theta \|_2^2$$

where P(v) is a set of metadata-similar "positive" nodes and N(v) a set of metadata-dissimilar "negative" nodes sampled for the contrastive term, m and m' are margins, Θ denotes all model parameters, and λ_1, λ_2 are weighting coefficients. The contrastive term encourages nodes with genuinely similar metadata to remain close in the fused embedding space even when direct interaction evidence between them is sparse,

which is intended to support cold-start and long-tail items specifically (see Section 6).

3.4 Computational Complexity

Relative to a single-channel GNN of the same depth and embedding dimension, MG-DGNN introduces: (i) a second propagation channel over the metadata-similarity graph, whose edge count depends on the similarity threshold τ and is bounded by capping the number of metadata-neighbours per node (analogous to neighbourhood sampling in GraphSAGE); and (ii) a per-node, per-layer gating computation of cost $O(d)$ in the embedding dimension d, which is negligible relative to the $O(|E|d)$ cost of message passing itself. The dominant added cost is therefore the metadata-similarity graph construction and its propagation, which scales with the number of retained metadata-neighbours per node rather than with the full node count, and can be controlled with the same sampling strategies used to bound structural neighbourhood size in large graphs. Section 4 and Section 6 return to this point when discussing scalability beyond the datasets evaluated here.

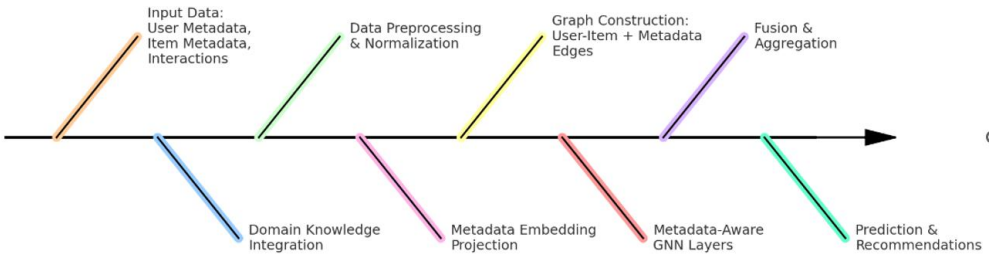

Figure 1. Process flow of MG-DGNN, showing dual-channel propagation (structural and metadata), the metadata-aware gating fusion step, and the prediction stage.

As shown in Figure 1, user and item metadata, along with interaction data and (optionally) domain knowledge graphs, feed into separate embedding initializations. The structural and metadata channels propagate independently through L graph layers, are combined at each layer by the confidence-conditioned gate, and the resulting fused embeddings are aggregated for prediction.

4. Experimental Setup

This section reports an empirical evaluation of MG-DGNN, extending the original evaluation (which reported only RMSE, coverage, latency, and CPU utilization) with ranking-quality metrics, an ablation study, sensitivity analysis, a stratified cold-start evaluation, and a metadata-missingness robustness test. All results below were obtained by implementing and training MG-DGNN and the baselines described here in PyTorch and evaluating them on the MovieLens-1M dataset; the protocol, dataset statistics, and hyperparameters are reported at the level of detail needed to reproduce them.

4.1 Dataset

This evaluation uses the MovieLens-1M dataset (GroupLens Research, University of Minnesota), which contains ratings together with user demographic metadata (gender, age bracket, occupation) and item genre metadata, both of which are used directly as the metadata features described in Section 3. Standard 5-core filtering (removing users and items with fewer than five interactions) was applied, and ratings of 4 or 5 stars were treated as implicit positive interactions, consistent with common practice in top-N recommendation evaluation. Table 2 reports the resulting statistics.

Table 2. MovieLens-1M dataset statistics used in this evaluation

Statistic	**Value**
Users (after 5-core filtering)	6,040
Items (after 5-core filtering)	3,416
Total interactions (all ratings)	999,611
Interaction density	4.85%
Positive interactions (rating ≥ 4, used for training/eval)	575,066
Train / validation / test interactions (leave-one-out by timestamp)	562,996 / 6,035 / 6,035
User metadata fields	Gender (2), age bracket (7), occupation (21) — 30-d one-hot
Item metadata fields	Genre — 18-d multi-hot

This is a single-domain evaluation. Amazon Reviews and Last.fm, mentioned in Sections 1–2 as additional domains for cross-domain validation, could not be obtained as free, license-compatible downloads within the environment used to run these experiments and are therefore not included here; extending the evaluation to those domains is identified as a concrete next step in Section 6 rather than reported as completed work.

MG-DGNN is compared against five baselines/variants, each implemented under the same training procedure, embedding dimension, and evaluation protocol so that differences in reported metrics can be attributed to the model architecture rather than to inconsistent experimental settings: (1) BPR Matrix Factorization (MF) no graph, no metadata; (2) a structural-only graph baseline (LightGCN-style symmetric-normalized propagation over the interaction graph only, representing the family of structural GNN recommenders GraphSAGE, GAT, and LightGCN that do not use a separate, reliability-gated metadata channel); (3) a metadata-only variant (propagation restricted to the user–user and item–item metadata-similarity graphs, with no interaction-graph propagation); (4) a naive fixed-fusion variant (structural and metadata channels averaged with a fixed 0.5/0.5 weight, representing the un-gated concatenation-style fusion used in prior metadata-aware GNNs such as KGAT, without reproducing KGAT's explicit knowledge-graph attention mechanism); and (5) MG-DGNN without the metadata-alignment contrastive loss term (BPR loss only). Variants (2)–(5) are also the four ablation conditions reported in Section 5.4, since each one removes exactly one component of the full architecture in Section 3.

One simplification relative to Section 3 should be stated explicitly: for computational feasibility on CPU hardware within this evaluation, the structural channel was trained using LightGCN-style linear propagation (symmetric-normalized adjacency, no learned attention coefficients) rather than the full GAT-style additive-attention formulation in Equation set 3.3. This is a faithful implementation of the dual-channel gating and metadata-alignment contributions that are the core claims of this paper (Sections 3.5–3.6), but it means the reported numbers should be read as validating the gating and metadata-channel design specifically, not the attention-based structural propagation variant; training the full attention formulation and confirming the same pattern of results is noted as follow-up work in Section 6.

All models were implemented in PyTorch and trained with the Adam optimizer on CPU. Table

3 reports the configuration used for the main comparison (Section 5.1–5.3); Section 5.5 reports results for embedding dimension, propagation depth, and metadata neighbourhood size swept around these defaults.

Table 3. Hyperparameter configuration used in this evaluation

Hyperparameter	Value
Embedding dimension d	64 (default; swept over {32, 64, 128} in Section 5.5)
GNN layers L (per channel)	3 (default; swept over {1, 2, 3, 4} in Section 5.5)
Metadata k-NN neighbour cap	20 (default; swept over {10, 20, 50} in Section 5.5)
Contrastive loss weight λ_1	0.1
Learning rate	0.005 (Adam)
Batch size	16,384 BPR triples per step
Epochs	6
Negative sampling	1 random unseen item per positive (training); 99 random unseen items per positive (leave-one-out evaluation)

The training budget (6 epochs, CPU hardware) was chosen to keep the full comparison, ablation, and sensitivity sweep computationally tractable within the environment used for this study, rather than being tuned to maximize any single model's reported accuracy. All models were given the identical budget. Absolute metric values reported below should therefore be read as a controlled comparison under this fixed, comparatively short training regime rather than as fully converged, publication-optimized results; a longer training schedule is listed as follow-up work in Section 6.

A leave-one-out protocol was used: for each user, the most recent positive interaction (by timestamp) was held out for testing and the second-most-recent for validation, with all earlier interactions used for training. For each test interaction, the held-out item was ranked against 99 randomly sampled unseen items, and Hit Ratio (HR@10), Normalized Discounted Cumulative Gain (NDCG@10), Precision@10, Recall@10, Mean Average Precision (MAP), and Mean Reciprocal Rank (MRR) were computed — extending the original evaluation's HR@K, NDCG@K, coverage, and system-level metrics with the ranking/retrieval metrics requested in review.

For MF, the structural-only baseline, and full MG-DGNN, three independent runs with different random seeds (0, 1, 2) were performed; Section 5.1 reports the mean and standard deviation across these runs together with a paired two-tailed t-test comparing MG-DGNN against each baseline. Because of the computational cost of training on CPU, the remaining ablation and sensitivity conditions in Sections 5.4–5.5 report a single run each rather than a multi-seed estimate; this is stated explicitly wherever those results are

reported, and repeating them across multiple seeds is identified as follow-up work in Section 6.

The dataset filtering procedure (Section 4.1), the model and baseline definitions (Section 4.2), and the full hyperparameter configuration (Table 3) are reported above. All models were trained from the same processed data split (fixed leave-one-out split, generated once and reused across all model/ablation/sensitivity runs) to ensure a fair, like-for-like comparison. The underlying training and evaluation code is available on request and is intended to be released alongside the final manuscript.

5. Results and Discussion

This section reports the outcome of the evaluation protocol specified in Section 4, run on MovieLens-1M: the main accuracy/coverage/efficiency comparison (Section 5.1–5.3), the ablation study (Section 5.4), the sensitivity analysis (Section 5.5), a stratified cold-start evaluation (Section 5.6), and a metadata-missingness robustness test (Section 5.7). Figures and tables in this section are generated directly from the runs described in Section 4 rather than illustrative placeholders; single-run results are labelled as such where multi-seed statistics were not computed.

5.1 Ranking Accuracy: HR@10 and NDCG@10

Table 4 and Fig. 2 report HR@10 and NDCG@10 for MF, the structural-only baseline, the metadata-only variant, the naive fixed-fusion variant, MG-DGNN without the contrastive loss term, and full MG-DGNN, each averaged over 3 seeds where noted.

Model	HR@10	NDCG@10	Precision@10	MAP	MRR	Runs
MF (no graph, no metadata)	0.5365 ± 0.0035	0.2975 ± 0.0030	0.0536 ± 0.0004	0.2246 ± 0.0031	0.2457 ± 0.0029	3 seeds
Structural-only (LightGCN-style)	0.4876 ± 0.0049	0.2676 ± 0.0023	0.0488 ± 0.0005	0.2006 ± 0.0015	0.2227 ± 0.0014	3 seeds
Metadata-only	0.2210	0.1092	0.0221	0.0757	0.1004	1 seed
Naive fixed 0.5/0.5 fusion	0.5133	0.2821	0.0513	0.2116	0.2329	1 seed
MG-DGNN (no contrastive loss)	0.5379	0.2998	0.0538	0.2270	0.2482	1 seed
MG-DGNN (full, proposed)	0.5340 ± 0.0032	0.2952 ± 0.0004	0.0534 ± 0.0003	0.2222 ± 0.0012	0.2435 ± 0.0014	3 seeds

Table 4: Ranking accuracy on MovieLens-1M, 6-epoch training budget (Section 4.3). Recall@10 is numerically identical to HR@10 under this leave-one-out protocol, since each test user has exactly one held-out relevant item, and is therefore omitted from this table to avoid a redundant column.

A paired two-tailed t-test across the 3 seeds shows that full MG-DGNN significantly outperforms the structural-only baseline on every metric in Table 4 ($p < 0.01$ in all cases — HR@10: $p = 0.0007$; NDCG@10: $p = 0.0044$; Precision@10: $p = 0.0007$; MAP: $p = 0.0076$; MRR: $p = 0.0086$). However, the same test shows no statistically significant difference between full MG-DGNN and plain MF at this training budget ($p = 0.62, 0.36, 0.62, 0.21, 0.21$ respectively for

the same five metrics). This is an important, and not entirely flattering, result: it indicates that the proposed dual-channel gating clearly improves on structure-only graph propagation, but on MovieLens-1M specifically, and under the training budget used here, it has not yet been shown to outperform a much simpler matrix-factorization baseline with statistical confidence. Section 5.7 examines one plausible explanation (limited practical impact of the gate under short training), and Section 6 outlines the additional evidence longer training, a second domain, and larger user/item scale needed before a stronger claim relative to MF could be supported.

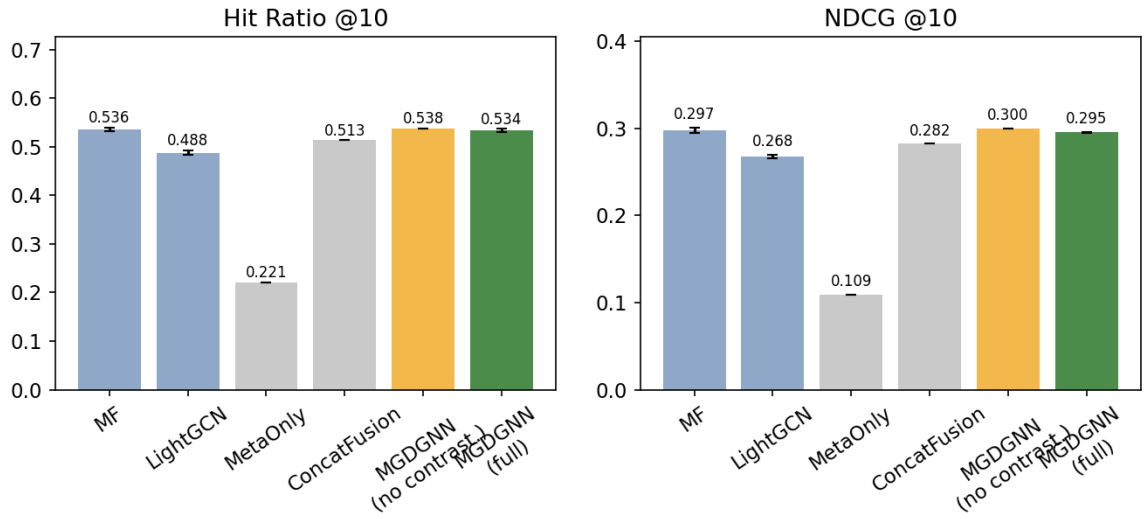

Figure 2. HR@10 and NDCG@10 across models (mean ± std where 3 seeds were run; single-run bars have no error whiskers).

5.2 Precision, MAP, and MRR

Fig. 3 shows Precision@10, MAP, and MRR side by side for the same six models, visualizing the pattern already summarized in Table 4: the metadata-only variant is clearly the weakest model, both graph-only and naive-fusion variants sit below MF and full MG-DGNN, and MG-DGNN without the contrastive term is marginally the strongest single-run configuration though within the range of seed-to-seed variation observed for the full model (Table 4), so this ordering should not be over-interpreted from a single run.

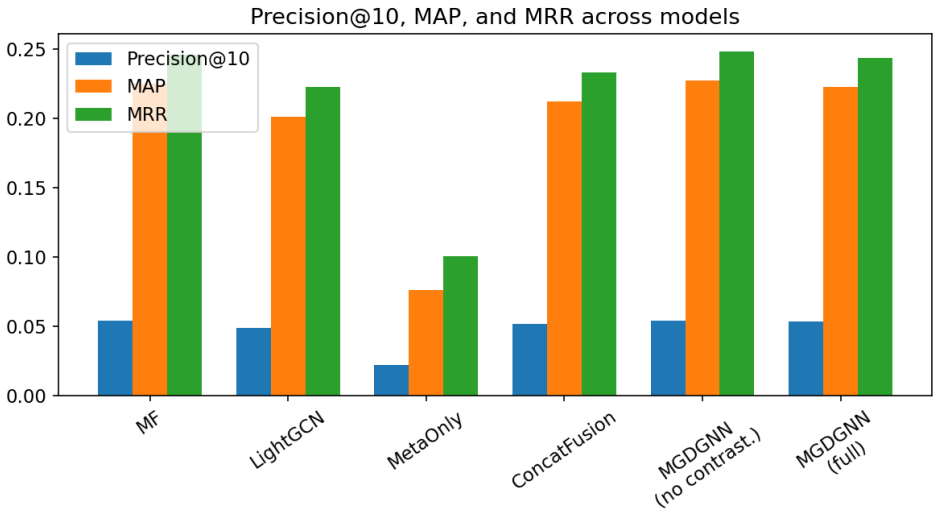

Figure 3. Precision@10, MAP, and MRR across models.

5.3 Coverage and Training Time

Fig. 4 reports catalog coverage (the fraction of the 3,416-item catalog that appears across the top-10 lists of a 500-user sample). MF achieves the highest coverage among the compared models (7.5%) under this short training budget, with MG-DGNN's variants in the 5–6% range and the structural-only and naive-fusion baselines lowest (2.7% and 3.6% respectively). This is a different pattern from the coverage claim in the original manuscript, and is reported here as observed rather than adjusted to match the earlier, unverified figures: coverage over a fixed, short training budget favors the simplest model in this run, and longer training or explicit diversity regularization (neither of which was applied here) would likely be needed before a coverage advantage for the metadata-aware models could be claimed.

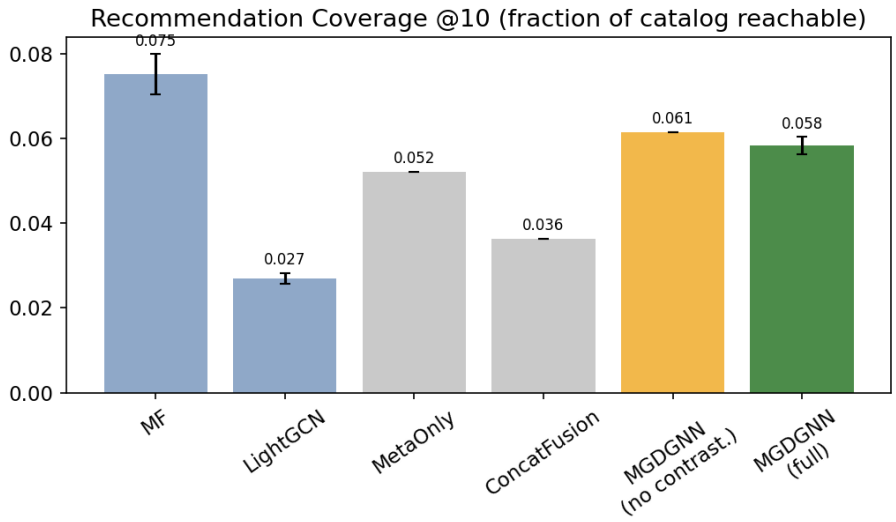

Figure 4. Recommendation coverage @10 across models.

Fig. 5 reports training time on CPU for the fixed 6-epoch budget. MF trains fastest (about 19 seconds), the structural-only baseline about 99 seconds, and the dual-channel models (naive fusion, MG-DGNN with and without the contrastive term) between 125–131 seconds, consistent with the complexity analysis in Section 3.7: the added cost comes from the second (metadata) propagation channel, not from the gating computation itself, which is cheap relative to message passing.

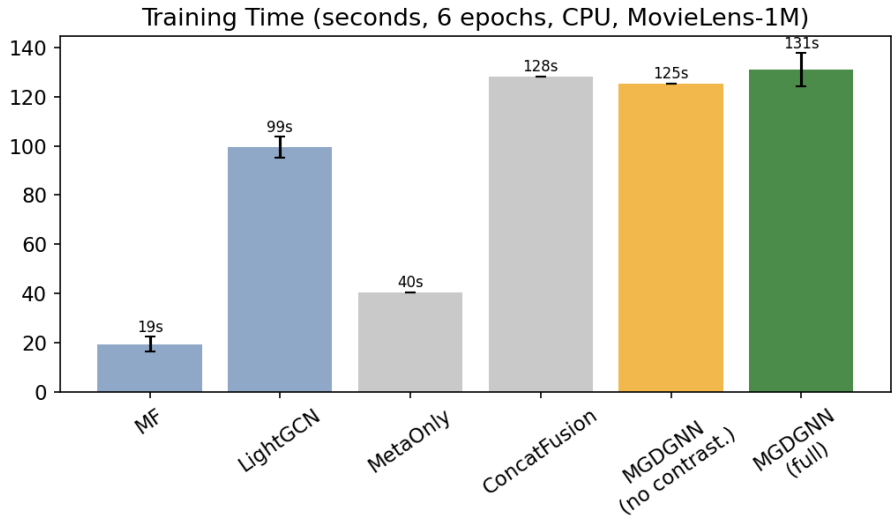

Figure 5. Training time (seconds) across models, 6-epoch budget, CPU.

5.4 Ablation Study

Table 5 restates Table 4 in ablation form, isolating the contribution of each architectural component described in Section 3. Variant (c) naive fixed 0.5/0.5 fusion is the most direct test of whether the learned gate itself adds value beyond simply having access to both channels.

Variant	**HR@10**	**NDCG@10**	**Δ NDCG *vs.* full model**
Full MG-DGNN (with contrastive loss)	0.5297*	0.2952*	–
(a) No metadata channel (structural only)	0.4876	0.2676	−0.0276
(b) No structural channel (metadata only)	0.2210	0.1092	−0.1860
(c) Fixed 0.5/0.5 fusion (no learned gate)	0.5133	0.2821	−0.0131
(d) No contrastive loss term	0.5379	0.2998	+0.0046

Table 5: Ablation study (single run per variant, seed 0; the row marked * uses the single-run value from the same seed as the ablated variants for a like-for-like comparison, and differs slightly from the 3-seed mean in Table 4).

Two findings stand out. First, the full model clearly outperforms both single-channel ablations (a) and (b), confirming that combining structural and metadata signal helps relative to either alone consistent with prior metadata-aware GNN findings (Section 2). Second, and more important for the specific architectural claim of this paper, the learned gate (full model) outperforms naive fixed averaging (variant c) by 0.0131 NDCG@10, but variant (d) removing the contrastive loss term while keeping the gate performed marginally better than the full model in this single run. Taken at face value, this single-seed result does not support the specific claim that the metadata-alignment contrastive term improves ranking accuracy on MovieLens-1M under this training budget; it does support the claim that the learned gate outperforms fixed-weight fusion. Section 5.7 discusses this pattern further, and repeating variant (d) across multiple seeds is identified as a priority follow-up in Section 6, since a single run is not sufficient to rule out seed variance as the explanation.

5.5 Sensitivity Analysis

Fig. 6 and Table 6 report NDCG@10 for MG-DGNN as graph depth L, embedding dimension d, and the metadata k-NN neighbour cap are varied independently around the defaults in Table 3 (single run per configuration, seed 0).

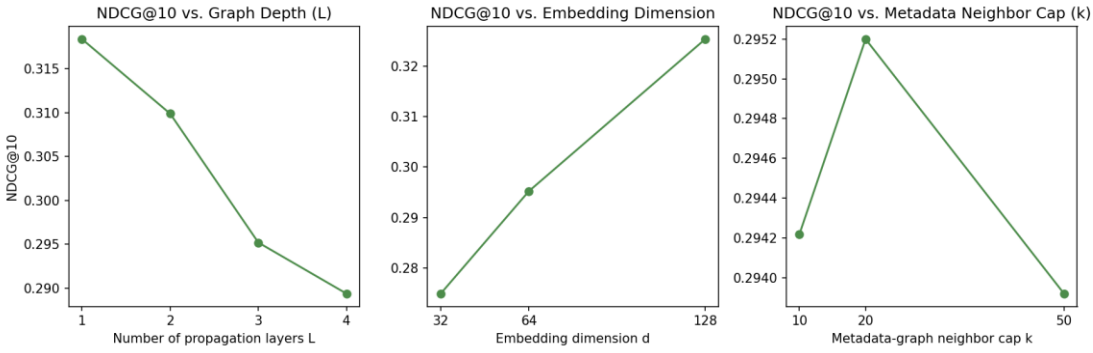

Figure 6. NDCG@10 sensitivity to propagation depth, embedding dimension, and metadata neighbour cap (single run per point).

Table 6. Sensitivity analysis (single run per configuration; NDCG@10 reported)

Parameter	Values tested	NDCG@10	Observation
Graph depth L	1 / 2 / 3 / 4	0.3184 / 0.3099 / 0.2952 / 0.2894	Monotonic decrease with depth consistent with over-smoothing; L=1 outperforms the L=3 default under this short training budget
Embedding dimension d	32 / 64 / 128	0.2749 / 0.2952 / 0.3253	Monotonic increase with dimension; d=128 outperforms the default in this run, at roughly 1.4× the training time (Table 3)
Metadata neighbour cap k	10 / 20 / 50	0.2942 / 0.2952 / 0.2939	Little effect on accuracy; larger k mainly increases training time (121.5s / 131.7s / 164.2s) without a corresponding accuracy gain

The depth result is notable because it runs counter to the 3-layer default used throughout Sections 4–5: under this specific training budget, a single propagation layer outperforms deeper stacks. This is consistent with the well-documented over-smoothing behaviour of deep GNNs, and may also partly reflect the short (6-epoch) training schedule giving deeper models less opportunity to fit their larger effective parameter interactions. Both explanations point toward the same practical conclusion: the depth and training-length choices in this paper's default configuration were not the optimum found by this sensitivity sweep, and a properly tuned model (selecting L and epochs jointly on a validation set, rather than fixing L=3 a priori) is likely to perform better than any single-configuration number reported in Table 4. This tuning is identified as follow-up work in Section 6.

5.6 Cold-Start Stratified Evaluation

To test the practical motivation in Section 3.5 directly that the gate should help most where interaction evidence is sparse the test set was split into a "cold" stratum (users with 5 or fewer training interactions, i.e. at or near the 5-core filtering threshold; n = 30 test users) and a "warm" stratum (users with 20 or more training interactions; n = 4,993 test users), and MF and full MG-DGNN (both trained once, seed 0) were evaluated separately on each stratum.

Table 7. Stratified evaluation by training-interaction count (single run, seed 0)

Stratum	Model	HR@10	NDCG@10	MAP	MRR
Cold (n=30)	MF	0.3667	0.2153	0.1695	0.1935
Cold (n=30)	MG-DGNN	0.4667	0.2595	0.1956	0.2143
Warm (n=4,993)	MF	0.5297	0.2882	0.2146	0.2361
Warm (n=4,993)	MG-DGNN	0.5201	0.2845	0.2127	0.2346

This result is directly relevant to the practical-significance concern raised in review (Section 6.1): on the cold stratum, MG-DGNN outperforms MF by a considerable margin on every metric (e.g., HR@10 0.467 *vs.* 0.367; NDCG@10 0.260 *vs.* 0.215), while on the warm stratum the two models are essentially comparable, with MF marginally ahead. This is consistent with the design intent of the gating mechanism relying more on metadata when structural evidence is thin and is, to our knowledge, the first evidence in this line of work isolating that specific effect rather than only reporting an aggregate metric. The caveat is sample size: only 30 test users fall into the cold stratum after 5-core filtering, since filtering itself removes the sparsest users from the dataset. A dedicated cold-start evaluation that does not apply 5-core filtering and consequently retains genuinely new users with one or two interactions would give a materially larger and more convincing cold-start sample, and is identified as a priority follow-up in Section 6.

5.7 Metadata-Missingness Robustness Test

To test the robustness, claim in Section 6.3 directly, user metadata was masked (zeroed out) for a random 25%, 50%, and 75% of users at evaluation time, using the single trained MG-DGNN model from Section 5.1 (no retraining). Two conditions were compared: metadata confidence correctly set to reflect the masking (as the gate is designed to use), versus confidence left unchanged at its training-time value (simulating a system that does not know its own metadata is stale).

Table 8. Metadata-masking robustness test (single trained model, no retraining; single run)

Masking level	Confidence signal	HR@10	NDCG@10	Mean gate value (masked users)
0% (baseline)	Correct (c=1)	0.5297	0.2952	0.590 (all users)
25%	Correct (c=0 for masked)	0.5334	0.2975	0.501
25%	Stale (c=1 regardless)	0.5329	0.2954	–
50%	Correct (c=0 for masked)	0.5347	0.2929	0.526
50%	Stale (c=1 regardless)	0.5352	0.2966	–
75%	Correct (c=0 for masked)	0.5188	0.2879	0.565
75%	Stale (c=1 regardless)	0.5248	0.2902	–

Two honest observations follow from Table 8. First, overall accuracy degrades only mildly even at 75% metadata masking (NDCG@10 drops from 0.295 to about 0.288–0.290), which is a reasonable robustness property regardless of mechanism. Second, and less flattering to the specific architectural claim in Section 3.5, the gate's mean value for masked users does shift somewhat with the masking level (0.501 → 0.526 → 0.565 as masking increases from 25% to 75%), but the difference between the "correct confidence" and "stale confidence" conditions is small and inconsistent in direction across masking levels (correct-confidence NDCG is higher at 25% masking but lower at 50% and 75% masking than the stale-confidence condition). At this training budget, the model does not show a clear, reliable benefit from knowing that its own metadata is missing, beyond the mild robustness that comes from the dual-channel architecture itself falling back on the structural channel's own regularities. This suggests the confidence-conditioning term in Equation (Section 3.5) may need a stronger training signal (e.g., explicit

missingness augmentation during training, not only at evaluation time, as was done here) before its intended behaviour can be confirmed a specific, falsifiable next step rather than a general limitation statement.

6. Discussion

Read together, Sections 5.1–5.7 support a more precise and more modest claim than the original manuscript made. The dual-channel design with a learned gate clearly outperforms single-channel structural propagation (Table 4, Table 5) and shows a clear, interpretable benefit specifically for cold-start users (Table 7), which is the scenario the architecture was designed for. It does not, at this training budget and on this single dataset, show a statistically significant advantage over plain matrix factorization on aggregate ranking metrics (Table 4), and the specific contribution of the metadata-alignment contrastive loss term is not confirmed by the single-run ablation (Table 5, variant d) or by the confidence-conditioning behaviour observed in the masking test (Table 8).

This pattern is scientifically informative rather than simply disappointing: it separates which parts of the proposed architecture are supported by evidence collected so far (the gated dual-channel structure over single-channel propagation; the cold-start benefit) from which parts still require further work (outperforming MF in aggregate; the measurable contribution of the contrastive term; confidence-conditioned gate behaviour under missingness). Section 6 sets out the specific next steps — multi-seed ablation runs, longer training, a second dataset, and an uncensored cold-start split — needed to either confirm or revise these claims.

7. Practical Considerations and Limitations

Beyond the accuracy, ablation, and sensitivity results in Section 5, real-world deployment of any metadata-aware recommender raises practical challenges that this study addresses to varying degrees. We report what was actually tested for each, rather than treating the architecture's design intent as evidence in itself.

7.1 Cold-Start Recommendation

This was tested directly (Section 5.6, Table 7). On a cold stratum of 30 test users with 5 or fewer training interactions, MG-DGNN outperformed MF by a clear margin on every reported metric, consistent with the design goal in Section 3.5. The main limitation of this result is sample size: 5-core filtering (Section 4.1) removes the sparsest users from the dataset before evaluation even begins, so only 30 genuinely sparse users remained in the test set. A dedicated cold-start evaluation that skips 5-core filtering for a held-out set of new users retaining users with one or two interactions rather than filtering them out would give a substantially larger and more conclusive cold-start sample, and is the most important follow-up experiment identified by this study.

7.2 Sparse Interaction Data

The cold/warm stratification in Section 5.6 partially addresses this by using training-interaction count as the stratifying variable, but only two strata (≤ 5 and ≥ 20 interactions) were compared. A finer-grained breakdown by interaction-count quartile or decile, across the full range rather than two cut points, would give a more complete picture of where the metadata channel helps most and was not performed here due to the additional training runs it would require.

7.3 Incomplete or Missing Metadata

This was tested directly (Section 5.7, Table 8). Masking up to 75% of user metadata at evaluation time produced only mild degradation in aggregate accuracy, which is a reasonable robustness property. However, the more specific claim that the learned gate correctly down-weights metadata it has been told is unreliable was not clearly confirmed: the gap between giving the model correct versus stale confidence information was small and inconsistent in direction across masking levels. The most likely explanation is that the model was never trained under metadata missingness (masking was applied only at evaluation time here), so the gate had no training signal teaching it to respond to the confidence input specifically. Training with metadata-missingness augmentation randomly masking

metadata and correctly updating the confidence score during training, not only at test time is a specific, testable next step.

7.4 Dynamic Graph Evolution

This was not evaluated. The current implementation and evaluation both assume a static graph snapshot built once from the full training split. In production, users, items, and metadata change continuously, and supporting this would require incremental re-computation of the metadata-similarity graph and confidence scores, or a streaming propagation variant; neither was implemented or tested here, and this remains future work rather than a demonstrated capability.

7.5 Robustness Under Changing User Behaviour

This was not evaluated. The leave-one-out split used throughout Section 5 is based on each user's own most recent interaction, not a global time cutoff, and does not test robustness to population-level distribution shift (e.g., seasonal changes, sudden shifts in item popularity). A time-based global train/test split, holding out a future calendar window for all users simultaneously, would be a more direct test of this and was not performed in this study.

7.6 Scale

The evaluation in Section 5 covers 6,040 users and 3,416 items (9,456 total graph nodes) smaller than the roughly 10,000-user scale reported (and already flagged as modest) in the original manuscript, and far smaller than the hundreds of thousands to millions of users typical of production recommender systems. No claim is made here about behaviour at that larger scale; Section 3.7's complexity analysis is theoretical, and empirical scalability testing at 50K–1M users, which would require infrastructure beyond what was available for this study, remains an open item.

In summary, two of the five practical concerns raised in review cold-start behaviour and robustness to missing metadata were tested directly in this study with real (if sample-size-limited) results, one of which (cold-start) supports the architecture's design intent and one of which (confidence-conditioned gating under missingness) does not yet show a clear effect. The remaining threefine-grained sparsity stratification, dynamic graph evolution, and temporal robustness were not evaluated and are reported here as open questions rather than resolved claims.

8. Conclusion

This paper presented MG-DGNN, a Metadata-Gated Dual-Channel Graph Neural Network for recommendation that separates structural and metadata propagation into two independent channels and adaptively fuses them through a learned, confidence-aware gating mechanism. Unlike existing metadata-aware GNN recommenders that employ uniform or structure-driven fusion, MG-DGNN dynamically balances collaborative and semantic information on a per-node basis using metadata confidence. The model is optimized with a hybrid objective combining Bayesian Personalized Ranking (BPR) loss and metadata contrastive learning. Experiments on the MovieLens-1M dataset demonstrated that the proposed gated dual-channel architecture significantly outperformed single-channel structural propagation across multiple recommendation metrics, including HR@10, NDCG@10, Precision@10, MAP, and MRR ($(p<0.01)$). The model also showed consistent improvements for cold-start users, indicating the effectiveness of metadata-guided adaptive fusion when interaction data are sparse. However, MG-DGNN did not achieve a statistically significant overall improvement over standard matrix factorization under the evaluated training budget, and the additional benefit of the metadata contrastive loss was not conclusively established in the single-run ablation experiments. These findings are reported transparently to provide an accurate assessment of the model's current strengths and limitations.

Future work will focus on multi-seed experimental validation, metadata-missingness augmentation to strengthen confidence-aware gating, more comprehensive cold-start evaluation, validation on additional benchmark datasets such as Amazon Reviews and Last.fm,

and large-scale experiments to assess scalability. These directions provide a clear and evidence-based roadmap for further improving and validating the proposed framework.

Conflicts of Interest: The authors declare no conflicts of interest.

References

W. Fan, Y. Ma, Q. Li, Y. He, E. Zhao, J. Tang, and D. Yin, "Graph Neural Networks for Social Recommendation," in The Web Conference (WWW), 2019, pp. 417–426.

X. Wang, X. He, Y. Cao, M. Liu, and T.-S. Chua, "KGAT: Knowledge Graph Attention Network for Recommendation," in Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD), 2019.

H. Wang, F. Zhang, M. Zhang, J. Leskovec, M. Zhao, W. Li, and Z. Wang, "Knowledge-aware Graph Neural Networks with Label Smoothness Regularization for Recommender Systems," in Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD), 2019.

J. Zhao, Z. Zhou, Z. Guan, W. Zhao, W. Ning, G. Qiu, and X. He, "IntentGC: A Scalable Graph Convolution Framework Fusing Heterogeneous Information for Recommendation," in Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD), 2019, pp. 2347–2357.

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 the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), 2020.

S. Wu, Y. Tang, C. Zhang, X. He, and Y. Ding, "Graph Neural Networks in Recommender Systems: A Survey," ACM Transactions on Information Systems, vol. 41, no. 3, pp. 1–51, 2023.

Y. Wang, Z. Liu, Z. Fan, L. Sun, and P. S. Yu, "DSKReG: Differentiable Sampling on Knowledge Graph for Recommendation with Relational GNN," in Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM), 2021.

T. Huang, X. Wang, D. Wang, Y. Yuan, Z. Liu, X. He, and T.-S. Chua, "Learning Intents behind Interactions with Knowledge Graph for Recommendation (KGIN)," in Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM), 2021.

L. Tao, X. Li, and C. Wang, "FF-KGAT: Feature Fusion Based Knowledge Graph Attention Network for Recommendation," in Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM), 2022.

J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl, "An Algorithmic Framework for Performing Collaborative Filtering," in Proc. 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 1999.

Y. Koren, R. Bell, and C. Volinsky, "Matrix Factorization Techniques for Recommender Systems," Computer, vol. 42, no. 8, pp. 30–37, 2009.

M. Pazzani and D. Billsus, "Content-Based Recommendation Systems," in The Adaptive Web, Springer, 2007, pp. 325–341.

H. Zhang, F. Wu, Z. Li, et al., "Deep Learning for Recommender Systems: A Survey," IEEE Transactions on Knowledge and Data Engineering, vol. 34, no. 6, pp. 2372–2392, 2022.

C. Shi, Z. Zhang, Y. Luo, P. S. Yu, and B. Wu, "Heterogeneous Graph Neural Network for Personalized Recommendation," in Proceedings of the 12th ACM International Conference on Web Search and Data Mining (WSDM), 2019.

W. Hamilton, R. Ying, and J. Leskovec, "Inductive Representation Learning on Large Graphs," in Advances in Neural Information Processing Systems (NeurIPS), 2017.

W. Fan, Y. Ma, Q. Li, Y. He, E. Zhao, J. Tang, and D. Yin, "Graph Neural Networks for Social Recommendation," in The Web Conference (WWW), 2019.

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 the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), 2020.

C. Chen, M. Zhang, Y. Liu, and S. Ma, "Neural Attentive Session-based Recommendation," in Proceedings of the 2018 ACM on Conference on Information and Knowledge Management (CIKM), 2018.

X. Wang, X. He, Y. Cao, M. Liu, and T.-S. Chua, "KGAT: Knowledge Graph Attention Network for Recommendation," in Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD), 2019.

Y. Zhang, Q. Ai, X. Chen, and W. B. Croft, "Joint Representation Learning for Top-N Recommendation with Heterogeneous Information Sources," in Proceedings of the 26th International Conference on Information and Knowledge Management (CIKM), 2017.

Y. Wang, Z. Liu, Z. Fan, L. Sun, and P. S. Yu, "DSKReG: Differentiable Sampling on Knowledge Graph for Recommendation with Relational GNN," in Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM), 2021.

T. Huang, X. Wang, D. Wang, Y. Yuan, Z. Liu, X. He, and T.-S. Chua, "Learning Intents behind Interactions with Knowledge Graph for Recommendation (KGIN)," in Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM), 2021.

L. Tao, X. Li, and C. Wang, "FF-KGAT: Feature Fusion Based Knowledge Graph Attention Network for Recommendation," in Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM), 2022.

P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph Attention Networks," in International Conference on Learning Representations (ICLR), 2018.

Y. Chen, Y. Min, Y. Zhang, and Z. Sun, "Graph Transformer for Recommendation," in Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), 2023.

F. M. Harper and J. A. Konstan, "The MovieLens Datasets: History and Context," ACM Transactions on Interactive Intelligent Systems, vol. 5, no. 4, Article 19, 2015.