

Enhancing Cross-Domain Recommendations through User Convergence and Contrastive Learning

Rabia Khan*, Naima Iltaf*, Fawad ur Rehman†

*Computer Software Engineering Department, National University of Sciences and Technology (NUST), Pakistan

†Department of Computer Science, Bahria University, Pakistan

rabia.khan@mcs.nust.edu.pk, naima@mcs.edu.pk, fawadfazal@gmail.com

Abstract—Cross-domain recommendation aims to leverage user behavior across multiple domains to alleviate data sparsity and enhance personalization. However, inconsistencies in user preferences across domains and the lack of aligned representations often hinder effective knowledge transfer. In this work, we propose a novel framework that employs contrastive learning to enable user convergence, aligning user representations across diverse domains. Our method utilizes contrastive objectives to bring semantically similar user behaviors closer in the shared representation space while preserving domain-specific nuances. By integrating both shared and domain-sensitive user features, the framework effectively captures latent preferences that span multiple domains. Experimental evaluations on benchmark cross-domain datasets demonstrate significant improvements over existing baselines in terms of recommendation accuracy and user embedding consistency. The proposed approach not only enhances cross-domain generalization but also offers a scalable solution for real-world recommendation systems where users interact with heterogeneous content.

Index Terms—cross-domain, recommendation, collaborative filtering, contrastive learning

I. INTRODUCTION

Cross-domain recommendation aims to alleviate data sparsity and cold-start problems by transferring knowledge from a source domain to improve recommendations in a target domain. However, a key challenge lies in effectively capturing and aligning user preferences across heterogeneous domains, especially when user behaviors are complex, sparse, or inconsistent. Traditional collaborative filtering methods and content-based models often struggle to model user intent accurately in such settings, particularly when relying on a single feedback type like ratings or textual reviews. Moreover, assuming that similar textual patterns or behaviors imply shared preferences can lead to incorrect generalizations, as users may express different motivations or sentiments across domains. To address these challenges, we propose a novel framework, MMGCN2: Metapath-based Multi-interest Graph Convolutional Network, that models dynamic user preferences and cross-domain alignment through the lens of opinion dynamics and social influence. By leveraging metapath-based relationships and multi-interest embeddings, our approach captures both domain-specific and shared aspects of user behavior. A graph convolutional architecture facilitates the propagation of preference signals through structured relations, allowing the model to better generalize across sparse and heterogeneous domains. Extensive experiments on benchmark

Amazon datasets demonstrate that MMGCN2 consistently outperforms state-of-the-art baselines, particularly in cold-start and sparse scenarios. Our results validate the effectiveness of incorporating fine-grained user interests and socially-informed opinion modeling for robust and personalized cross-domain recommendations.

II. RELATED WORK

A. Cross-Domain Recommendation

Cross-domain recommendation techniques address the limitations of single-domain systems by transferring knowledge across different but related domains to enhance recommendation accuracy, particularly in scenarios with sparse data or cold-start users. Traditional approaches were grounded in matrix factorization (MF), where methods like Collective Matrix Factorization (CMF), TransferMF, and TCF incorporated shared latent user/item representations across domains. These models assumed the existence of overlapping users or items and typically enforced joint optimization over multiple rating matrices. While effective to some extent, such MF-based techniques often relied on linear assumptions and lacked the flexibility to model complex user-item interactions. To overcome these limitations, deep learning-based methods emerged, leveraging autoencoders, neural collaborative filtering, and multi-view learning frameworks to extract high-level semantic features across domains. These models allowed for the integration of heterogeneous data sources such as user reviews, item metadata, and browsing history. Techniques like cross-domain denoising autoencoders and variational models captured non-linear mappings and domain-invariant features, improving generalization. Additionally, attention mechanisms and multi-head networks were introduced to dynamically weigh information from source and target domains based on relevance, leading to more adaptive and personalized recommendations. Transformer-based architectures and contrastive learning have recently gained traction for their ability to align user preferences across domains without explicit supervision, offering a promising direction in handling domain shifts, user behavior divergence, and non-overlapping user sets.

B. Review based Cross-Domain Recommendation

Early review-based recommendation models extracted user preferences from review text using sentiment cues and aspect terms to improve rating prediction. For example, man-

ually curated aspects from restaurant reviews, combined with sentiment-aware Matrix Factorization, mapped polarity and intensity to latent dimensions for refined user representations [1], [7], [10].

Later work adopted unsupervised methods such as topic modeling and aspect-based analysis. Models like LARA [6] and LDA-MF [19] linked latent topics with preference factors, while aspect-oriented approaches either explicitly extracted aspect terms via rule-based or supervised NLP tools [10] or implicitly learned them through neural and probabilistic models [18]. Neural methods, including CNN-based opinion mining [15], integrated aspect and review-level signals for better prediction [2], [4].

While effective, these methods largely focused on static, item-level analysis. Our approach instead builds a dynamic, user-centric neighborhood across three dimensions: semantic similarity of review content, rating behavior alignment, and rating based similarity yielding a more adaptive understanding of preferences. This triangulated similarity improves personalization and addresses cold-start scenarios by synthesizing diverse signals beyond direct preference data.

III. METHADODOLOGY

The process begins with the extraction of domain-specific features by analyzing metadata and user-item rating matrices within each domain. Subsequently, users common to both domains are leveraged to construct a cross-domain mapping function, guided by their neighborhood relationships. A Knowledge Graph supports this effort by enabling graph-based modeling of user and item representations. This heterogeneous graph encodes not only user-user similarities but also user-item interactions. To capture content-driven similarities between users and items, a neighborhood graph is formed and processed using a multi-layer GCN. Drawing inspiration from [?], a self-supervised pre-training mechanism is integrated into the GCN. This approach applies contrastive loss in the latent space to strengthen the alignment between two augmented graph views, enhancing the consistency and quality of the learned representations. The graphical illustration of framework is given in Figure 1.

A. Text-Driven Modeling of User Similarity Across Domains

Let $\mathcal{U} = \{u_1, u_2, \dots\}$ and $\mathcal{V} = \{v_1, v_2, \dots\}$ denote the sets of users and items, respectively. The matrices $\mathcal{R}^s$ and $\mathcal{R}^t$ represent user-item ratings in the source and target domains. Here, $\mathcal{R}_{uv}^s$ indicates the rating assigned by user u to item v in the source domain, while $\mathcal{R}_{uv}^t$ denotes the corresponding rating in the target domain. The key objective is to predict ratings in the sparse target domain, leveraging the known rating matrices $\mathcal{R}^s$ and $\mathcal{R}^t$, along with the user and item sets $\mathcal{U}$ and $\mathcal{V}$.

Let U_L represent users who are present in both domains. The training data comprises vector pairs $\{U_{v*}^s, U_{v*}^t\}_{v \in U_L}$, capturing shared user behavior across domains. The task is to infer missing ratings $\mathcal{R}_{ux}$ in the target domain by using known ratings $\mathcal{R}_{ui}$ from the source domain. A user profile

$\mathcal{P}_U$ is constructed by selecting items rated above a threshold $\theta \in [1, 5]$.

All reviews c_{ui} written by user u for positively rated item i are concatenated into a document, where l is the number of sentences:

$$c_{ui} = \{s_1, s_2, s_3, \dots, s_l\}$$

The full corpus $\mathcal{C}$ is formed by combining documents from m users and n positively rated items from both domains:

$$\mathcal{C} = \{c_1, c_2, c_3, \dots, c_{m+n}\}$$

Even when users rate the same item, their preferences may vary. This variation can be understood from their review texts, allowing for finer personalization. The model input is therefore defined as:

$$\mathcal{X} = \{u, i, r_{ui}, c_{ui}\}$$

To process informal and out-of-vocabulary (OOV) terms commonly found in user reviews, we employ a WordPiece tokenizer. It splits unfamiliar words into known subword units or characters. The tokenizer yields token IDs and attention masks, which are input into a transformer-based architecture. BERT serves as a contextual encoder and can be viewed as a function $\mathcal{B} : P \rightarrow \mathbb{R}^{N \times h}$, where h is the hidden dimension and $N = 512$ is the padded token sequence length.

SBERT (Sentence-BERT) is a Siamese adaptation of BERT designed to produce sentence-level embeddings by pooling token representations. It addresses the challenge of efficient similarity computation by mapping all sentences into a shared vector space. The architecture uses twin networks with shared parameters and is applied in tasks such as Semantic Textual Similarity (STS), semantic search, and paraphrase mining. The resulting sentence embeddings are compared via cosine similarity, and the model is trained using backpropagation to minimize mean squared error. Semantic similarity between user metadata vectors—query q and document d —is computed as:

$$S_s(q, d) = \frac{\sum(P_{(n,d)} \times P_{(n,q)})}{\sqrt{\sum(P_{(n,d)})^2 \times \sum(P_{(n,q)})^2}} \quad (1)$$

Here, n denotes the number of terms, P represents the weight of the document term, q is the query vector, and d is the target document vector.

Although semantic similarity is effective for modeling user interests, it should be complemented with behavioral and rating-based patterns, which are covered in the next section.

B. Rating-Based User Similarity Computation

The user-item rating matrix is employed to identify users who tend to assign similar ratings to the same items. The similarity rating matrix S_r of size $m \times m$, captures the pairwise similarity scores among the users based on their rating behavior. It is given by,

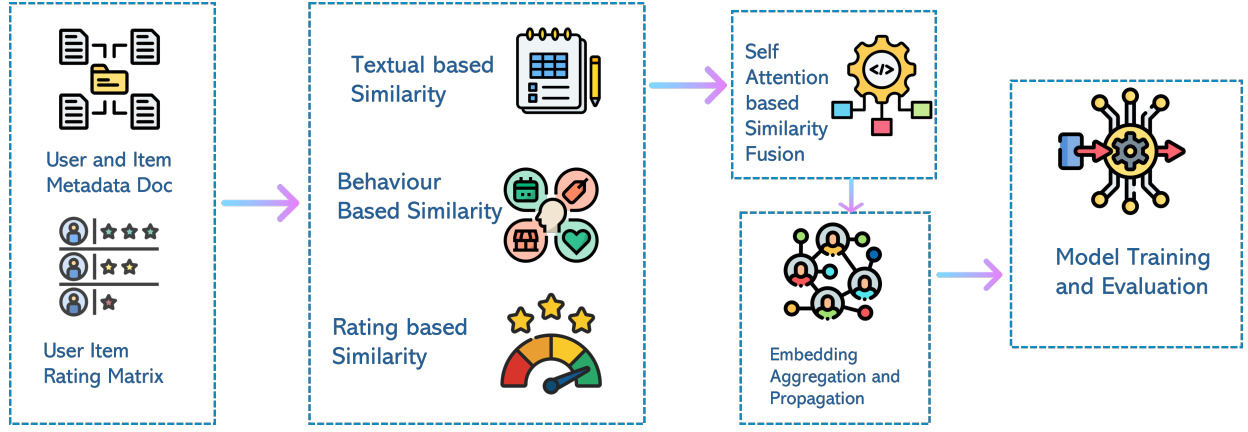


Fig. 1. Architecture of MMGCN2

$$S_r = \begin{bmatrix} s_{10} & s_{11} & s_{12} & \dots & s_{1m} \\ s_{20} & s_{21} & s_{22} & \dots & s_{2m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ s_{i0} & s_{i1} & s_{i2} & \dots & s_{im} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ s_{m0} & s_{m1} & s_{m2} & \dots & s_{mm} \end{bmatrix}^{m \times m}$$

By computing the cosine similarity (or its variation) between these vectors, we quantify the behavioral similarity between users.

The similarity score between two users, say user1 and user2, is given by:

$$S_r(u_1, u_2) = \frac{\sum_{i=1}^m (R(u_1, i) - \bar{R}(u_1)) (R(u_2, i) - \bar{R}(u_2))}{\sqrt{\sum_{i=1}^m (R(u_1, i) - \bar{R}(u_1))^2} \sqrt{\sum_{i=1}^m (R(u_2, i) - \bar{R}(u_2))^2}} \quad (2)$$

This formulation corresponds to the Pearson Correlation Coefficient (PCC), which captures the linear correlation between the rating patterns of two users, normalized by their individual average ratings. Compared to traditional cosine similarity, PCC is more robust in handling scenarios where users may have rated different subsets of items, as it focuses on co-rated items and adjusts for user biases in rating scales.

C. Behavior-Driven User Similarity Modeling

The user-item rating matrix offers only partial insight into a user's behavioral preferences. To clarify this limitation, consider the example from Table I, where user1 assigns a rating of 1 to the movie *PussinBoots*, which indicates a negative response. This rating could suggest that user1 dislikes the genres associated with *PussinBoots*, namely *Animation*, *Children*, and *Adventure*. However, the same user assigns a rating of 4 to the movie *LionKing*, which falls under the genres: *Animation*, *Adventure*, and *Drama*. This inconsistency highlights that a single rating does not accurately capture a user's overall preference when a movie spans multiple genres. Therefore, to more precisely understand user preferences across genres, a behavioral analysis is necessary [11].

TABLE I
SUMMARY OF RATING AND GENRES OF RATED MOVIES

userid	movietitle	Genre	ratingvalue	score
1	Puss in Boots	Animation	1	1
		Children		1
		Adventure		1
1	Lion King	Animation	4	1
		Adventure		1
		Drama		1
1	Disenchanted	Adventure	1	1
		Comedy		1
		Animation		1
1	Wallace and Gromit	Animation	1	1
2	The Matrix	Sci-Fi	5	1
		Action		1
2	The Family Guy	Animation	2	1

The user-item rating matrix alone provides limited insight into behavioral preferences. For instance, in Table I, user1 rates *Puss in Boots* (Animation, Children, Adventure) as 1, suggesting dislike. Yet, they rate *Lion King* (Animation, Adventure, Drama) as 4, indicating enjoyment of Animation despite the earlier low score. This shows that single ratings cannot reliably capture genre-level preferences when movies span multiple genres.

To address this, the behavioral similarity metric S_b analyzes correlations between user ratings and item genre metadata, computing a separate behavioral score per genre based on the number of movies the user has rated in that category. For example, user1's score for Animation is 4, as they have rated four Animation movies: *Puss in Boots*, *Lion King*, *Disenchanted*, and *Wallace and Gromit*. This approach refines recommendations by recognizing that user1 still favors Animation overall, even if some individual movies in the genre receive low ratings.

D. Self-Attention-based Similarity Fusion

We propose to assign dynamic weights to similarity type to capture their varying importance across different user context instead of assigning fixed weights. A self-attention mechanism dynamically determines the contribution of each similarity

measure when aggregating them into a unified score. For each user pair (u, v) , we consider a similarity vector:

$$\mathbf{S}(u, v) = [S_b(u, v), S_r(u, v), S_s(u, v)]$$

where S_b , S_r , and S_s represent behavioral, rating, and semantic similarities, respectively. This vector is treated as an input sequence, and attention weights are learned to highlight the relative importance of each similarity type.

The attention computation follows the scaled dot-product attention formulation:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V$$

where the query (Q), key (K), and value (V) matrices are obtained through linear transformations of the similarity vector:

$$Q = \mathbf{S}(u, v)W^Q, \quad K = \mathbf{S}(u, v)W^K, \quad V = \mathbf{S}(u, v)W^V$$

Here, $W^Q, W^K, W^V \in \mathbb{R}^{3 \times d}$ are trainable parameter matrices, and d_k denotes the dimension of the key vectors. The dynamic weight computation of three similarity metrics is graphically shown in Figure 2.

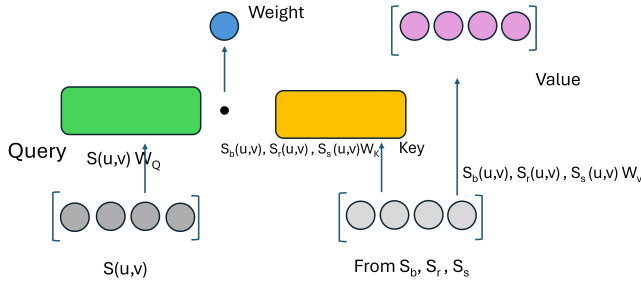


Fig. 2. Self-attention-based similarity weight computation

The contextualized representation is produced by:

$$\mathbf{H}_{\text{ctx}} = \text{Attention}(Q, K, V) \quad (3)$$

To project $\mathbf{H}_{\text{ctx}}$ into a final similarity score and model higher-order relationships, we apply an MLP defined as:

$$\mathbf{z} = \text{ReLU}(\mathbf{H}_{\text{ctx}}W_1 + b_1) \quad (4)$$

$$S(u, v) = \sigma(\mathbf{z}W_2 + b_2) \quad (5)$$

where W_1, W_2 and b_1, b_2 are learnable parameters, $\text{ReLU}(\cdot)$ introduces non-linearity, and $\sigma(\cdot)$ denotes the activation function (e.g., sigmoid or softmax) for producing the final prediction.

By adding contextualization, the model can dynamically weigh similarity components based on relevance, while the MLP provides a flexible non-linear mapping, making the overall framework more expressive and robust.

E. Graph Embedding via GCN

Based on the user similarity computed in the previous section, a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ is constructed, where $\mathcal{V}$ represents the set of nodes and $\mathcal{E}$ the set of edges. Edges connecting nodes of the same type capture similarity relationships, whereas edges between different node types represent user-item interactions. When a user u_i rates an item v_j , the rating is normalized as $\mathcal{R}_{ij} / \max(\mathcal{R})$, to adjust for users' individual rating biases. This normalization helps standardize varying rating patterns across users.

The similarity graph $S \in \mathbb{R}^{m \times m}$ is created by retaining only those user-user similarities (from Equation 7) that exceed a threshold θ , which is heuristically set to 0.6. This graph structure spatially positions users with similar preferences closer together. For nodes with sparse connections, embeddings are refined by propagating information from neighboring users with similar interests.

The generated heterogeneous graph incorporates both interaction-based relationships (from the adjacency matrix) and semantic similarity (from the matrix SS), where each node has a corresponding embedding. Two main inputs are considered:

- An input embedding matrix of dimensions $N \times e$, where N is the total number of nodes and e is the embedding dimension obtained from the prior layer.
- An $N \times N$ adjacency matrix representing the graph structure $\mathcal{G}$.

These embeddings are iteratively refined through multi-hop neighborhood aggregation and graph convolution operations. The graph captures both (1) similarity among semantically similar nodes and (2) interactions derived from adjacency connections. For cold-start users with limited interactions in the target domain, recommendations are inferred by associating their preferences with those of similar users in the source domain. The graph structure enables higher-order connectivity through L -hop neighbors, allowing user interests to be shaped by multiple levels of context. The GCN takes two inputs: the adjacency matrix and the node embeddings from the previous stage. These node embeddings, representing local neighborhoods, are processed through dense layers and aggregated accordingly. The adjacency matrix is derived from the user-item rating matrix $\mathcal{R}$, preserving graph structure while enabling lower-dimensional node representations. Similarity-based embeddings ensure semantically aligned nodes remain close in the learned space.

Let $Y \in \mathbb{R}^{(m+n) \times D}$ denote the embedding matrix of users and items. The GCN takes the embedding matrix Y and the graph $\mathcal{G} = \langle \mathcal{U} \cup \mathcal{V}, A \rangle$, where A is constructed from the rating matrix $\mathcal{R}$, and initializes the embeddings as:

$$\forall i \in U \cup V, \quad h_i^0 = e_i \quad (6)$$

The GCN performs iterative propagation and non-linear transformations, as expressed in Eq. 9. At each layer, the representation of each node is updated, and the final output embedding $H^{(K)}$ integrates neighborhood information up to

K hops. Node embeddings are aggregated from neighbors using both adjacency and similarity relationships. For a user node i , the aggregated latent factor h_i is computed as:

$$h_i^k = \sigma \left(W \cdot \left\{ \sum_{o \in \mathcal{N}(i)} \beta_i h_o^k \right\} + b \right), \quad \forall o \in \mathcal{N}(i) \quad (7)$$

Here, σ is a non-linear activation function, and the aggregation is an element-wise mean over the embeddings $\{h_o^k \mid o \in \mathcal{N}(i)\}$. The weight β_i is set as:

$$\beta_i = \frac{1}{|\mathcal{N}(i)|}$$

For multi-layer GCNs, the forward propagation at layer k is defined as:

$$h_i^k = ReLU \left(W \cdot \left\{ \sum_{o \in \mathcal{N}(i)} \frac{1}{|\mathcal{N}(i)|} h_o^{k-1} \right\} + b \right) \quad (8)$$

In this setup, each node's representation at step k is computed by aggregating the representations from step $k-1$. The base case ($k=0$) corresponds to the input embeddings. While deeper GCNs can capture higher-order structures, excessive depth may lead to over-smoothing or vanishing gradients. Empirical results suggest that a depth of $k=2$ provides optimal performance [17].

The final embedding is refined by concatenating a node's own representation with its aggregated neighborhood embedding:

$$h_i^k = \sigma \left(W \cdot [h_i^{k-1} \oplus n_i^{k-1}] \right) \quad (9)$$

where n_i^{k-1} is the aggregated neighborhood embedding, and σ is an activation function, such as $ReLU$. This concatenation enriches the representation by capturing both local and contextual information.

To further enhance the embeddings, contrastive learning is applied to align similar users/items and distinguish dissimilar ones. This approach encourages the model to learn more discriminative representations by aligning positive pairs while pushing apart negative ones across domains. The contrastive loss function is defined as:

$$\mathcal{L}_{CL} = -\frac{1}{N} \sum_{i=1}^N \log \frac{\exp(\text{sim}(h_u^i, h_v^i)/\tau)}{\sum_{j=1}^M \exp(\text{sim}(h_u^i, h_v^j)/\tau)} \quad (10)$$

where:

- h_u^i : embedding of the i -th user in the positive pair
- h_v^i : corresponding item/user embedding in the positive pair
- h_v^j : item/user embedding in the negative pair
- $\text{sim}(\cdot, \cdot)$: similarity function
- N : number of positive pairs
- M : number of negative pairs

- τ : temperature parameter to control the distribution sharpness

The objective is to minimize the distance between embeddings of positive pairs (u^i, v^i) while maximizing the distance between user u^i and unrelated items v^j . Gradients from this loss are propagated through the network, and model parameters—including embeddings and neural network weights—are updated using the Adam optimizer, with improvements occurring iteratively.

F. Learning Cross-Domain Mapping Function

Latent embeddings computed for cold-start users in the source domain are not directly applicable in the target domain due to semantic disparities between domain-specific embedding spaces. For instance, a user inclined toward *SciFi* in the *Movies* domain may exhibit a preference for *ScienceandMaths* in the *Books* domain. Although such cross-domain preferences often reflect consistent user interests, an explicit modeling of this transfer is necessary, as the latent spaces are not aligned across domains.

To address this, we utilize linked users U_L to bridge the semantic divide. The enriched embedding space of the source domain is leveraged to infer preferences for cold-start users in the target domain through a mapping function f_m . This function learns to project a user's latent representation from the source domain to the corresponding space in the target domain. The task is framed as a supervised regression problem aimed at learning this cross-domain mapping.

The objective is to train f_m to predict a user's latent embedding in the target domain based on their representation in the source domain. The training minimizes the discrepancy between the predicted and actual latent embeddings of the user, using the following formulation:

$$\min_{\gamma} \sum_{u \in \mathcal{U}} L(\widehat{U}_i^t, U_i^t) \quad (11)$$

$$\widehat{U}_i^t = f_m(U_i^a; \gamma)$$

The final stage of the model estimates the relevance score $\hat{\mathcal{R}}_{ui}$ between target users and items. The input embedding matrix Y_{in}^a for domain a is composed of both joint embeddings for overlapping users (U^j) and distinct user embeddings (U^a), such that $Y^a = U^j : U^a$. Similarly, the item embedding matrix for domain a is denoted as $Z_{in}^a = V^a$.

These embeddings serve as input to a Multi-Layer Perceptron (MLP) parameterized by weights W^a and activated by $ReLU$. The resulting user and item embeddings are:

$$Y_i = \alpha(\dots \alpha(Y_{in}^a, W^a). W^a)$$

$$Z_j = \alpha(\dots \alpha(Z_{in}^a, W^a). W^a)$$

where α denotes the $ReLU$ activation function.

The relevance between user i and item j is then estimated using cosine similarity:

$$\hat{\mathcal{R}}_{ui} = \cos(Y_i^a, Z_j^a) = \frac{Y_i^a \cdot Z_j^a}{\|Y_i^a\| \cdot \|Z_j^a\|} \quad (12)$$

The model is optimized by minimizing a weighted binary cross-entropy loss:

$$\text{loss}(\mathcal{R}_{ui}, \hat{\mathcal{R}}_{ui}) = \frac{\mathcal{R}_{ui}}{\max(\mathcal{R})} \log \hat{\mathcal{R}}_{ui} + \left(1 - \frac{\mathcal{R}_{ui}}{\max(\mathcal{R})}\right) \log (1 - \hat{\mathcal{R}}_{ui}) \quad (13)$$

where $\max(\mathcal{R})$ represents the maximum observed rating within the domain.

IV. EXPERIMENTAL EVALUATION

To evaluate the performance of the proposed model, we conduct a comprehensive series of experiments. The objective is to assess how well the model performs in comparison to existing state-of-the-art approaches on real-world datasets and to understand how various hyperparameter configurations affect the model's recommendation accuracy.

The evaluation is carried out using the Amazon 5-Core datasets [19], which include product ratings, user-generated textual reviews, and rich item metadata. In our experimental setup, the Amazon-Movies dataset serves as the source domain, while Amazon-Books and Amazon-Music are treated as target domains. The Amazon-Books dataset contains 8,560 explicit ratings $R_{ui} \in \{1, 2, 3, 4, 5\}$ from 38,200 users on 14,540 books. Amazon-Music has 2,560 interactions involving 8,600 users and 2,860 music items. Amazon-Movies provides 20,580 user-item interactions from 50,240 users on 15,370 items. Summary statistics of the datasets are shown in Table II.

All metadata undergo preprocessing steps including lemmatization, tokenization, and removal of stopwords. The source domain data is filtered to retain users who have at least five metadata fields, and items enriched with descriptive content are preserved.

TABLE II
STATISTICS OF AMAZON DATASETS

Dataset	Domain	Users	Items	Rating & MetaData
Amazon	Movies	50,240	15,370	20,580
	Books	38,200	14,540	8,560
	Music	8,600	2,860	2,560

To obtain contextualized sentence embeddings for item metadata, we use SBERT [10], a Siamese architecture built on the bert-base model. We employ the *stsb-mpnet-base-v2* variant, fine-tuned on the STS-B dataset containing 8,626 sentence pairs annotated with similarity scores from 0 to 5. The underlying *BERT_{base}* model comprises 12 encoder layers with 12 attention heads each, producing embeddings of dimensionality $d = 768$. Training is performed using the Adam optimizer, a learning rate of $5e-2$, a batch size of 32, and early stopping if validation accuracy does not improve for 5 consecutive epochs. Input sequences are truncated at 512 tokens.

To simulate the cold-start scenario, 50% of bridge users are designated as cold-start. This set is further split with 30% used for testing, 20% for validation, and the remaining 50% for training. The Graph Convolutional Network (GCN) is implemented in PyTorch with a batch size of 128 and a learning rate of 0.001, using the Adam optimizer. Each node in the GCN includes up to 25 neighbors, and the similarity threshold varies between 0.4 and 0.65. Parameters are initialized using a Gaussian distribution, and the embedding size l is varied across values [8, 16, 32, 64, 128]. For every observed interaction (positive sample), five negative samples are randomly generated. Training halts when the RMSE fails to improve for five consecutive epochs. During contrastive learning, real user-item interactions are considered as positive pairs, while negative pairs are sampled at random. Regularization terms are controlled using $\lambda_1 = 1 \times 10^{-2}$ and $\lambda_2 = 1 \times 10^{-3}$.

We evaluate our proposed model against several established baseline methods. These include EMCDR [16], a framework that jointly learns user representations in both domains and maps them; DHCD [14], which integrates content-based and collaborative signals for hybrid cross-domain learning; CDLFM [4], which performs latent feature mapping; NGCF [5], a graph-based model leveraging neighborhood information; DGCF [24], which applies disentangled representation learning on graphs; SEAGULL [25], which incrementally aligns graph structures; CVPM [26], a meta-learning-based preference transfer method; and GDCCDR [27], which performs graph-based disentangled contrastive learning for personalized transfer.

To assess recommendation performance, we employ Root Mean Square Error (RMSE) and Mean Absolute Error (MAE). RMSE is defined as:

$$RMSE = \sqrt{\sum_{u,i \in T_E} \frac{(\mathcal{R}_{ui} - \hat{\mathcal{R}}_{ui})^2}{|T_E|}} \quad (14)$$

where $\mathcal{R}_{ui}$ and $\hat{\mathcal{R}}_{ui}$ are the actual and predicted ratings respectively, and $|T_E|$ is the number of test interactions.

MAE calculates the average magnitude of prediction error:

$$MAE = \frac{\sum_{i=1}^n |\mathcal{R}_{ui} - \hat{\mathcal{R}}_{ui}|}{n} \quad (15)$$

While both metrics reflect prediction accuracy, RMSE penalizes larger errors more heavily than MAE.

For ranking-based evaluation, we use Hit Ratio@K (HR), Recall@K, and Normalized Discounted Cumulative Gain@K (nDCG@K). HR measures whether the correct item appears in the top-K recommendation list:

$$HitRatio@K = \frac{\text{Number of hits in } K}{|T|} \quad (16)$$

DCG and its normalized version nDCG are defined as:

$$CD_K = \sum_{t=1}^K rel_t, \quad DCG_K = \sum_{t=1}^K \frac{2^{rel_t} - 1}{\log_2(i + 1)}$$

$$nDCG@K = \frac{DCG_K}{IDCG_K} \quad (17)$$

where $IDCG_K$ is the ideal DCG score for the ranked list.

Finally, Recall@K measures the proportion of relevant items retrieved among the top-K recommendations:

$$Recall@K = \frac{TP_K}{TP_K + FN_K} \quad (18)$$

where TP_K is the count of true positive recommendations and FN_K is the number of relevant items not retrieved in the top-K results.

V. RESULTS AND ANALYSIS

To conduct a fair comparison with existing neural-based recommendation models, we maintain uniform hyperparameter settings across all experiments. Specifically, the embedding dimension is set to $l = 128$, the minibatch size is fixed at 1024, and network parameters are initialized with a Gaussian distribution. The learning rate and regularization coefficient are both configured at 0.001, and each model is trained for 100 epochs. These settings remain consistent while evaluating two ranking-based metrics: Hit Ratio at K (HR@K) and Normalized Discounted Cumulative Gain at K (nDCG@K), which are illustrated in Figure 3. The values of K range from 10 to 100 to analyze performance at varying recommendation list lengths.

Our framework outperforms EMCDR, CDLFM, and DHCD by overcoming the sparsity limitations of matrix factorization-based embeddings through a transformer-based encoder with stronger representational capacity. Unlike CDLFM, which uses only ratings, we adopt a multi-aspect approach that incorporates ratings, behavioral biases, and semantic metadata to identify semantically aligned neighbors, enriching embeddings with meaningful signals.

On the Amazon dataset, the model improves HR@TopK by 5.09% over CDLFM, with additional gains in MAE and RMSE across various sparsity levels as shown in Table III and Table IV. By integrating similarity and adjacency graphs with contrastive learning, our method enhances robustness, mitigates over-smoothing in GCN layers, and captures fine-grained user-item interaction patterns more effectively than NGCF and DGCF.

SEAGULL uses progressive structural alignment to match interaction patterns across domains, but its reliance on structural similarity limits performance in highly heterogeneous domains. Our model avoids this by applying multi-head attention over both domain-specific and domain-shared representations, enabling dynamic user modeling without structural alignment. CVPM transfers valence preferences via meta-optimization for sentiment, whereas our approach directly encodes sentiment into user behavior using a RoBERTa-based sentiment-aware encoder. This is combined with contrastive learning to ensure semantic alignment and domain-invariant user embeddings, mitigating domain divergence and semantic drift. GDCCDR applies disentangled contrastive learning over

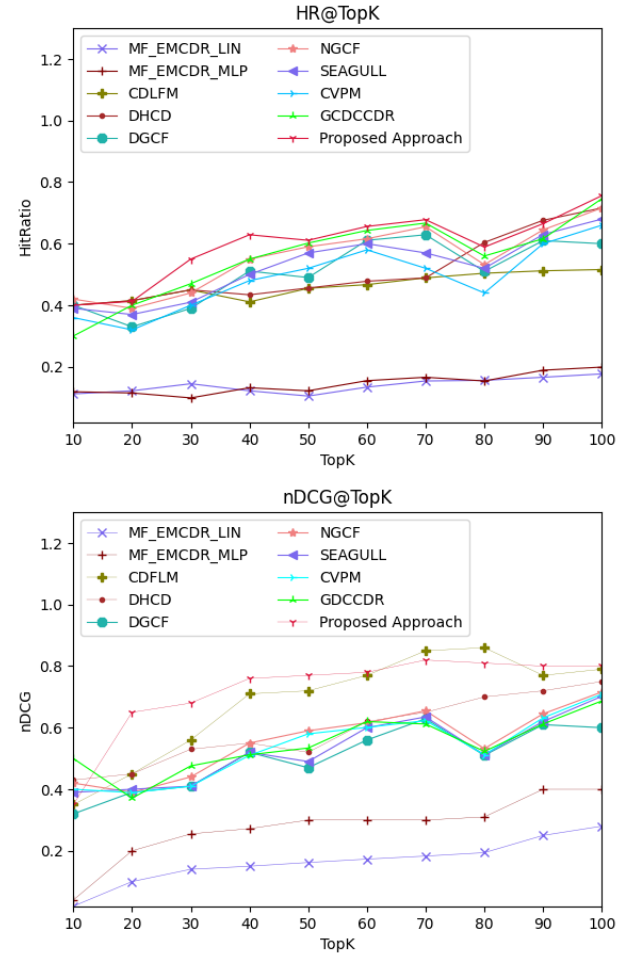


Fig. 3. TopK performance of all methods on Amazon Datasets w.r.t hitRatio@K and nDCG@K

interaction graphs, but its need for rich graph structures hinders performance in sparse domains. In contrast, our review-centric contrastive pre-training learns effectively even in low-resource or cold-start scenarios.

The proposed framework outperforms EMCDR, CDLFM, and DHCD by overcoming the limitations of matrix factorization-based embeddings, which struggle in sparse data. Leveraging a transformer-based encoder, our model offers higher representational capacity and robustness. Unlike CDLFM, which only uses ratings, we adopt a multi-aspect approach incorporating ratings, behavioral biases, and semantic metadata, enabling the identification of users with aligned behaviors and consistent tendencies. This yields a 5.09% HR@TopK improvement on the Amazon dataset.

Graph-based methods like NGCF and DGCF already surpass MF baselines by exploiting multi-hop neighbor aggregation. Building on this, we integrate similarity and adjacency graphs for finer preference learning, while contrastive learning enhances robustness and discriminability, mitigating over-smoothing in deep GCNs. Unlike SEAGULL, which depends on structural alignment and struggles in heterogeneous domains, our multi-head attention dynamically models

TABLE III
MAE PERFORMANCE COMPARISON AMONG BASELINES WITH VARYING SPARSITY IN AMAZON DATASET

Dataset	Model	10%	20%	30%	40%	50%
Amazon (Movies → Books)	MF_EMCDR_LIN [16]	0.8784	0.8725	0.8648	0.8481	0.8298
	MF_EMCDR_MLP [16]	0.8650	0.8672	0.8197	0.8271	0.8046
	CDLFM [18]	0.7809	0.7598	0.7638	0.7588	0.7688
	DHCD [14]	0.8162	0.821	0.8511	0.8112	0.7921
	NGCF [5]	0.7850	0.893	0.882	0.812	0.823
	DGCF [24]	0.867	0.824	0.855	0.793	0.821
	MMGCN2	0.761	0.735	0.758	0.745	0.766
Amazon (Movies → Music)	MF_EMCDR_LIN [16]	0.7645	0.7610	1.029	1.055	1.178
	MF_EMCDR_MLP [16]	0.7781	0.7810	1.199	1.204	1.236
	CDLFM [18]	0.7855	0.7799	1.079	1.188	1.198
	DHCD [14]	0.7850	0.7788	1.029	1.130	1.136
	NGCF [5]	0.895	0.992	0.998	1.15	1.109
	DGCF [24]	0.878	0.882	0.867	0.864	0.877
	MMGCN2	0.761	0.762	0.758	0.752	0.759

TABLE IV
RMSE PERFORMANCE COMPARISON AMONG BASELINES WITH VARYING SPARSITY IN AMAZON DATASET

Dataset	Model	10%	20%	30%	40%	50%
Amazon (Movies → Books)	MF_EMCDR_LIN [16]	0.91	0.95	0.975	0.977	1.0
	MF_EMCDR_MLP [16]	0.8952	0.989	1.0038	1.014	1.052
	CDLFM [18]	0.952	1.069	1.155	1.169	1.126
	DHCD [14]	0.996	1.099	1.198	1.212	1.278
	NGCF [5]	0.934	0.939	1.122	1.076	0.996
	DGCF [24]	0.99	0.989	1.02	1.08	1.067
	MMGCN2	0.867	0.999	1.0032	1.014	0.985
Amazon (Movies → Music)	MF_EMCDR_LIN [16]	0.873	0.996	1.029	1.055	1.178
	MF_EMCDR_MLP [16]	0.856	1.066	1.199	1.204	1.236
	CDLFM [18]	0.851	0.998	1.079	1.188	1.198
	DHCD [14]	0.925	0.989	1.029	1.130	1.136
	NGCF [5]	0.895	0.992	0.998	1.15	1.109
	DGCF [24]	1.08	0.987	1.12	1.16	1.121
	MMGCN2	0.825	0.967	0.978	1.009	1.105

both domain-specific and shared representations without such assumptions.

Compared with CVPM, which transfers sentiment preferences via meta-optimization, we directly encode sentiment into user behavior using a RoBERTa-based sentiment-aware encoder, coupled with contrastive learning for semantic alignment and domain-invariant embeddings. Unlike GDCCDR, which requires rich graph structures for disentangled contrastive learning, our review-centric contrastive pre-training effectively handles sparse and cold-start scenarios.

VI. CONCLUSION

The presented method utilizes a graph neural network framework to predict user-item ratings by incorporating metadata from multiple domains. This cross-domain integration effectively addresses the cold-start problem in the target domain by leveraging richer information from the source domain. To identify users with similar preferences, the approach combines the SBERT model with cosine similarity, allowing the model to measure semantic closeness among users. Influence scores are computed by analyzing common semantic traits, behavioral data, and rating patterns across users. The model strategically

combines interaction data and auxiliary metadata to build a graph structure, enabling hierarchical learning of user representations through graph attention mechanisms. These mechanisms account not only for the individual user's preferences but also for those of semantically similar users. An attention-based transfer strategy is further introduced for users who exist in both domains, enhancing cross-domain knowledge transfer. The system integrates a GCN architecture enriched with metadata from each domain to improve recommendation outcomes. Moreover, contrastive learning is embedded into the propagation of GCN embeddings to refine the quality and stability of learned user and item representations. The effectiveness of this approach is supported by extensive empirical evaluation, where it outperforms existing baselines in terms of MAE and RMSE, particularly in scenarios involving sparse data.

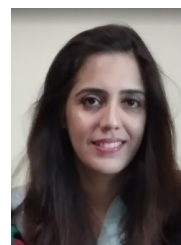
REFERENCES

- [1] Qiming Diao, Minghui Qiu, Chao-Yuan Wu, Alexander J. Smola, Jing Jiang, and Chong Wang. Jointly Modeling Aspects, Ratings and Sentiments for Movie Recommendation (KDD)(2014)
- [2] Z. Cheng, Y. Ding, L. Zhu, and M. Kankanhalli, Aspect-aware latent factor model: Rating prediction with ratings and reviews, in Proceedings of the 2018 world wide web conference, 2018, pp. 639–648.

- [3] A. Da'u, N. Salim, I. Rabi, and A. Osman, "Weighted aspect-based opinion mining using deep learning for recommender system," *Expert Systems with Applications*, vol. 140, p. 112871, 2020.
- [4] C. Wu, F. Wu, J. Liu, Y. Huang, and X. Xie, "Arp: Aspect-aware neural review rating prediction," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019, pp.2169–2172.
- [5] Wang, X., He, X., Wang, M., Feng, F., & Chua, T.-S. (2019). Neural Graph Collaborative Filtering. *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 165–174. <https://doi.org/10.1145/3331184.3331267>
- [6] Liye Shi, Wen Wu, Wenxin Hu, Jie Zhou, Jiayi Chen, Wei Zheng, Liang He, DualGCN: An Aspect-Aware Dual Graph Convolutional Network for review-based recommender, *Knowledge-Based Systems* (2022), vol.242, 108359, ISSN 0950-7051,<https://doi.org/10.1016/j.knosys.2022.108359> (2022)
- [7] Gayatri Ganu, Yogesh Kakodkar, and Am 'eLie Marian. 2013. Improving the Quality of Predictions Using Textual Information in Online User Reviews. *Inf.Syst.* 38, 1 (March 2013), 1–15.
- [8] Julian McAuley and Jure Leskovec. Hidden Factors and Hidden Topics: Understanding Rating Dimensions with Review Text (RecSys '13). ACM, 8 (2013)
- [9] F. Zhu, Y. Wang, J. Zhou, C. Chen, L. Li and G. Liu, A Unified Framework for Cross-Domain and Cross-System Recommendations. In *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 2, pp. 1171–1184, doi: 10.1109/TKDE.2021.3104873(2023)
- [10] Y. Zhang, G. Lai, M. Zhang, Y. Zhang, Y. Liu, and S. Ma. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pp. 83–92(2014)
- [11] Huan Yan, Chunfeng Yang, Donghan Yu, Yong Li, Depeng Jin, and Dah Ming Chiu. Multi-Site User Behavior Modeling and Its Application in Video Recommendation. *IEEE Transactions on Knowledge & Data Engineering* 33, 180–193(2021)
- [12] Y. Yoro, M. Hirano, K. Oka, and Y. Tagawa, Electron spectroscopy studies on magneto-optical media and plastic substrate interface, *IEEE Transl. J. Magn. Japan*, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].
- [13] M. Young, *The Technical Writer's Handbook*. Mill Valley, CA: University Science, 1989.
- [14] Doan, T.-N., & Sahebi, S. (2019). Review-Based Cross-Domain Collaborative Filtering: A Neural Framework (Deep Hybrid Cross-Domain, DHCD model). In *Proceedings of the Third Workshop on Recommendation in Complex Scenarios*.
- [15] A. Da'u, N. Salim, I. Rabi, and A. Osman, "Weighted aspect-based opinion mining using deep learning for recommender system," *Expert Systems with Applications*, vol. 140, p. 112871, 2020.
- [16] M. TABASSUM and M. LALITHA, "Opinion word expansion and target extraction through double propagation," 2016.
- [17] Cheng Zhao, Chenliang Li, and Cong Fu. 2019. Cross-Domain Recommendation via Preference Propagation GraphNet. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. Association for Computing Machinery, New York, NY, USA, 2165–2168. <https://doi.org/10.1145/3357384.3358166>(2019)
- [18] F. Zhu, Y. Wang, J. Zhou, C. Chen, L. Li and G. Liu, A Unified Framework for Cross-Domain and Cross-System Recommendations. In *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 2, pp. 1171–1184, doi: 10.1109/TKDE.2021.3104873(2023)
- [19] Julian McAuley and Jure Leskovec. Hidden Factors and Hidden Topics: Understanding Rating Dimensions with Review Text (RecSys '13). ACM, 8 (2013)
- [20] Jianmo Ni, Jiacheng Li, and Julian McAuley. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the Empirical Methods in Natural Language Processing (EMNLP)*, 2019.
- [21] Shiwen Wu, Yuanxing Zhang, Wentao Zhang, Kaigui Bian, and Bin Cui. 2021. Enhanced review-based rating prediction by exploiting aside information and user influence. *Knowledge-Based Systems* 222, 107015.(2021)
- [22] Hongwei Wang, Fuzheng Zhang, Mengdi Zhang, Jure Leskovec, "Knowledge-aware Graph Neural Networks with Label Smoothness Regularization for Recommender Systems" ,In *Proceedings of the 25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining(KDD)*, 2019
- [23] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019, "Neural graph collaborative filtering". In *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*. pp.165–174.
- [24] Gan, Mingxin & Ma, Yingxue. Knowledge transfer learning from multiple user activities to improve personalized recommendation. *Soft Computing*. 26. 10.1007/s00500-022-07178-6.(2022)
- [25] Zhao, Y., Zhang, J., Li, C., Li, Z., Zhang, W., & Wang, J. (2022). SEAGULL: Cross-Domain Recommendation via Progressive Structural Alignment. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM)*, pp. 225-234.
- [26] Zhao, Y., Li, Z., Zhang, W., Li, C., Zhang, J., & Wang, J. (2023). CVP: Cross-domain Transfer of Valence Preferences via a Meta-optimization Approach. In *Proceedings of the 29th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 345-355.
- [27] Liu, X., Li, Y., Zhang, H., Sun, J., & Liu, Y. (2023). GDCCDR: Graph Disentangled Contrastive Learning with Personalized Transfer for Cross-Domain Recommendation. In *Proceedings of the ACM Web Conference 2023 (WWW)*, pp. 1775-1784.



Rabia Khan is lecturer from National University of Sciences and Technology (NUST), Pakistan. She is currently doing her PhD Degree in Software Engineering from the same institution and has more than 8 research publications in international journals and conferences to her credit. She has organized many workshops and symposiums related to Data Mining and Machine Learning in her department. Her research interests include Data Mining, Natural Language Processing and Machine Learning



Naima Iltaf has received the Ph.D. degree in Software Engineering from National University of Sciences and Technology, Pakistan, in 2013. She is currently a Professor at Department of Computer Software Engineering, NUST. She is author or co-author of more than 40 articles published in international journals and conferences. She is engaged with a few academic and industrial research projects as PI and Co-PI. Her research interests include Data mining, text mining, Natural language Processing and Recommender Systems.



Fawad ur Rehman received his MS degree from Bahria University, Pakistan in 2025, where he was a distinction holder. He has been associated with the field for the past ten years and has conducted research in Data Mining, Information Retrieval, and Multi-modal Recommendation.