# TransRecG: Transformer-Based Recommendation Systems using Graph Embeddings

Course project for CSE 6240: Web Search and Text Mining, Spring 2023

Sai Prasath Suresh
Georgia Institute of Technology
Atlanta, Georgia, USA
saiprasath3344@gatech.edu

Ishwarya Sivakumar
Georgia Institute of Technology
Atlanta, Georgia, USA
ishsiva@gatech.edu

Jeongjin Park
Georgia Institute of Technology
Atlanta, Georgia, USA
jpark3141@gatech.edu

Balaram Behera
Georgia Institute of Technology
Atlanta, Georgia, USA
balaramdb@gatech.edu

## ABSTRACT

In recent years, Deep Learning-based Recommendation Systems (DLRS) have become the industry standard. However, many existing DLRS ignore the temporal information that can be extracted from the user's history of interactions. They also fail to capture the higher order connectivity features like user-user and movie-movie(or item) relations. To address the above problems, we propose TransRecG: a Transformer-based Recommendation System using Graph Embeddings. First, we create a user-movie-attribute knowledge graph (KG). A Neural Graph Collaborative Filtering (NGCF) model that uses a Relational Graph Convolutional Neural Network (RGCN) is trained on this KG to capture the user-movie links and high order user-user and movie-movie relations. Second, we use a Transformer to understand the user's behavior based on the sequential order of the movies watched by them. We use embeddings learnt by the RGCN model to initialize the user and movie embeddings for the Transformer. Therefore, our proposed TransRecG model captures both the higher order connectivity information, and the sequential order with which the users interact with the movies. The model predicts the rating a user will give for a movie based on their watch history. The movie with the highest rating can be recommended as the next movie that the user will watch. We analyze the performance of our model on the MoviesLens1M dataset which contains 6,040 users and 3,883 movies, and show that the proposed model outperforms existing baselines. We also demonstrate how the model efficiently handles cold start users, and how the number of samples in the training dataset affect the model's performance.

## 1 INTRODUCTION

Many DLRS rely solely on the user-movie graph for generating embedding and performing recommendations, and ignore the underlying user-user and movie-movie relations. Capturing the user-user and movie-movie relations can help improve the quality of recommendations given by a recommendation system (RS). This is because similar users generally prefer watching similar movies. While user-user interactions are difficult to capture due to privacy and security concerns, similarities between users can be identified by comparing the movies they have watched. To address these challenges, we propose to augment the RS with embeddings from a NGCF model that captures the high order user-user and movie-movie relations. This is achieved by training a Relational Graph Convolutional Network (RGCN) on a user-movie-attribute knowledge graph (KG) for learning the user and movie embeddings.

Session-based recommendation systems have shown superior performance over conventional recommendation systems because of their ability to capture sequential information [8]. However, many DLRS considers each movie individually without considering the sequential ordering of the movies, because of which the systems fail to capture the evolving behavior of the users, and the co-occurrence of movies. To solve this problem, we propose to use a transformer-based recommendation system that captures the sequential ordering of movies, and personalize the movie recommendations by using the user embeddings generated from the KG.

In this project we use the NGCF architecture with the RGCN model trained on a user-movie-attribute KG to capture the user-user and movie-movie interactions alongside user-movie interactions. The user and movie embeddings learnt by the RGCN model is then augmented with the Transformer which helps in capturing both the sequential and the higher order connectivity information. Therefore, the proposed TransRecG model 1 can provide more personalized (using user embeddings), dynamic (using sequence of movies), and relevant (using higher order information) recommendations. The model's architecture is described in Figure 1. The model is trained on the MovieLens1M dataset for the next movie rating prediction task, and outperforms the baseline Transformer-based models and NGCF models. Specifically, our model showed the lowest Mean Absolute Error, 0.741, against 8 different baselines. As the model provides personalized, dynamic and relevant recommendataions it can impact many industry-based recommendation systems like Netflix (movies), Amazon (products), and Spotify (songs).

The rest of the report is organised as follows: Section 2 summarizes the previous works in this field. Section 3 describes the raw dataset, data processing techniques, and data statistics. Section 4 formalizes the next movie rating prediction task, and describes the loss function and optimizers used in this project. Then, Section 5 elaborately discusses the various models used in this project, along with their architectures. Section 6 compares various models, and

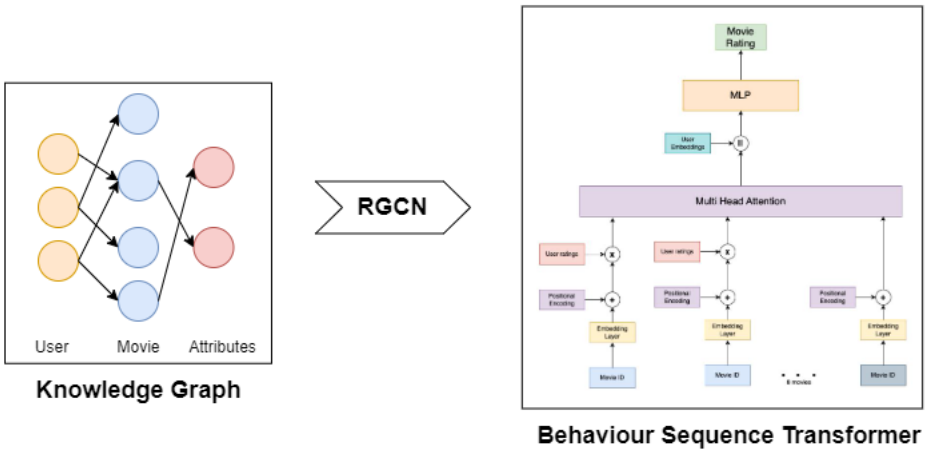



**Figure 1: TransRecG: Overview of Model Architecture**

analyses the user cold start problem. Finally, Section 7 concludes the project, and discusses potential future works.

## 2 PREVIOUS WORKS

In this section, we review some of previous attempts made to improve the Recommendation System for movies using Graph Neural Network (GNN). Wang et al. [7] proposed the Neural Graph Collaborative Filtering (NGCF) Model to exploit the user-item graph structure by propagating the embeddings on the graph and performing the edge prediction on the generated embeddings. Though the NGCF model captures high-order connectivity features using GCN, it does not capture underlying user-user or item-item relations. To address this problem, Wang et al. [6] proposed the Knowledge Graph Attention Network (KGAT) which uses graph attention models on a user-movie-attribute KG for generating the user and movie embeddings. This helps in capturing high order user-user and movie-movie relations and also helps with the cold start problem for new movies.

However, both the NGCF and KGAT techniques ignore the sequential nature in which the movies are watched by the user. To capture the sequential ordering of movies, Chen et al. [1] proposed the Behaviour Sequence Transformer (BST) recommendation system that uses transformers to capture the sequential ordering of movies to perform recommendation, and customizes the recommendations using the user embeddings. However, BST does not capture high-order user-user or item-item relations as it considers every user as an independent entity and learns their embeddings individually.

Therefore, we augment the transformer-based recommendation system described in BST with NGCF embeddings generated from a user-movie-attribute knowledge graph using an RGCN. This model captures both the high order user-user and item-item relations, and also the sequential order in which the items are consumed by the user.

## 3 DATASET DESCRIPTION

For this project we use the publicly available MovieLens1M dataset [2] which has the following three data files:

- `users.dat`: Contains the `user_id`, `sex`, `age_group`, `occupation` and `zipcode`.
- `movies.dat`: Contains the `movie_id`, `title` and `genres` (multi-valued attribute)
- `ratings.dat`: Contains `user_id`, `movie_id`, rating and `unix_timestamp`.

The users are those who joined MovieLens platform in 2000, and this dataset was released in 2003 [2].

### 3.1 Data Preparation

**Sequential Data Creation.** The unix_timestamp of ratings is utilized to sort the ratings data. It is then split into three separate sets - train, test, and validation - using the "Split based on Timestamp" approach. The training set comprises all ratings that precede the timestamp 975768738.0, accounting for 80% of the data. The validation set consists of ratings after 975768738.0 but before 978133376.4, representing 10% of the dataset. The remaining ratings after 978133376.4 are designated as the test set. The data leakage problem is avoided by splitting the dataset in this method.

The ratings are grouped based on the user and sub-sequences of size $n$ with a step size of 2 is created. In this project we use $n = 8$. The task of the proposed model is as follows: given the previous $n - 1$ ratings provided by a user, predict the rating that the user will give for the $n^{th}$ movie.

### 3.2 Raw Data Statistics

There are 6040 users and 3883 movies in the dataset. The user-movie-attribute KG is constructed from the training set. It has a total of 100,010 nodes and 1,006,617 edges Each user has rated at least 20 movies, and the average number of ratings per user is 164.7. The ratings range from 1 to 5 stars, with a mean of 3.58 stars, and

include no explicit ground-truth labels. Some distinct statistics from the dataset are written below:

- "Drama" is the most popular genre in movies.
- The Top 5 popular movies (based on the number of ratings) are: "American Beauty" (1999), "Star Wars: Episode IV - A New Hope" (1977), "Star Wars: Episode V - The Empire Strikes Back" (1980), "Star Wars: Episode VI - Return of the Jedi" (1983), "Jurassic Park" (1993).
- The movies with the highest rating in the top 5 genres with most movies are: "Drama" - "Silence of the Lambs" (1991), "Comedy" - "American Beauty" (1999), "Action" - "Star Wars: Episode IV - A New Hope" (1977), "Thriller" - "Sixth Sense" (1999), "Romance" - "Rebecca" (1940)

To find movies with the highest rating in the top 5 genres, after data pre-processing, we counted the number of rating per movie, and calculated mean and sorted the movies of the same genre based on these two values.

# 4 EXPERIMENTAL SETTINGS

We model the RS task as a regression task where the model predicts the rating that the user will provide for the $n^{th}$ given the ratings of the previous $n-1$ movies. It can be defined as follows: given a user $U$'s watch history $H(u) = \{m_i\}_{i=1}^{n-1}$, and the corresponding ratings given by the user $R(u) = \{r_i\}_{i=1}^{n-1}$ we need to learn a function $F$ that predicts the rating that the user $U$ will give for the $(m_n)^{th}$ movie. The next movie to be recommended is the movie with the highest user rating.

There are 2 components in our model: (1) A NGCF model with RGCN weights trained on the KG, that generates user and movie embeddings used in the BST and (2) A BST which takes as input the user and movie embeddings and learns the function $F$. Root Mean Squared Loss (RMSE) is used for optimizing the RGCN and BST parameters. The RGCN and BST are trained separately. AdamW optimizer was used. For evaluating the performance of the models `Mean Absolute Error` metric. The train-val-test split is 80-10-10 and there are 757077 data points in the training set, 92113 in the validation set and 91185 data points in the test set when we set $n = 8$. There are 611 users in the validation set and 200 users in the test set that do not have ay data points in the training set, and hence suffer from the cold-start problem.

As the dataset only contains 1 million data points, all our experiments are run on the Google Colab platform.

# 5 METHODS

## 5.1 Baseline Description

The proposed model's performance is compared against two baseline models: NGCF [1] and BST [2] [3]

*5.1.1 Neural Graph Collaborative Filtering Model.* The NGCF [7] model utilizes a GCN to generate node embeddings from the user-movie bipartite graph. There exists an edge between the user and the movie, if the user has rated the movie. All nodes are initialized with the adjacency matrix representation. We use a 2 layer message passing GCN with an hidden dimension of 18, for generating the user and movie embeddings. For all ratings in the dataset, we concatenate the corresponding user and movie embeddings, and pass it through a MLP for predicting the rating.

*5.1.2 Behavior Sequence Transformer.* The BST [1] model uses a transformer encoder-based architecture for predicting the rating that the a given user will provide for $n^{th}$ movie given the previous $n-1$ movies the user has reviewed. In this project, we consider $n = 8$. The model generates user and movie embeddings of size 63 using their metadata. A single multi-head attention layer with 9 attention heads is used for capturing the temporal relations between various movies. Before the movie embeddings are passed to the Transformer, positional encoding is added to them to preserve the ordering, and the resulting embeddings are multiplied by their corresponding ratings. The output of the Transformer is then concatenated with the user embeddings, and are then passed to a MLP for predicting the ratings.

Both the baseline models (NGCF and BST) parameters are optimized on the rating prediction task using the RMSE Loss function and the AdamW optimizer.

## 5.2 Proposed Model

The NGCF model does not consider the sequential ordering of the movies, whereas the BST model fails to capture higher order user-user or movie-movie relations. The TransRecG model overcomes these drawbacks by combining the approaches described in the NGCF and BST models. The proposed TransRecG model has 3 main components: (1) the user-movie-attribute KG, (2) NGCF using RGCN, and (3) BST.

*5.2.1 Knowledge Graph Construction.* The KG contains a node for all `user_id`, `movie_id` and `genre`. The KG is composed of two bipartite graphs. (1) The *user-movie graph*: if the user has rated a movie then there exist an edge between the user and the movie. (2) The *movie-attribute graph*: if the movie belongs to a particular `genre` then there exist an edge between the genre and movie. As a single movie can have multiple genres, there can exist more than one edge for a movie in the movie-attribute graph.

The KG is an heterogeneous graph in terms of it's edges. There are 5 different edge types in the user-movie subgraph, and the edge type between a particular user and a movie corresponds to the rating (1-5) the user has given to that movie. All edges in the movie-attribute subgraph belongs to the same type. Here, we consider `genre` as the only attribute for the movies, and all the edges belong to the same type "belongs to". Figure 2 shows small portion of the KG that was constructed.

We initialize the `movie_id` and `genre` node embeddings with their corresponding GloVe representations (dimension = 50). For example, if the `genre` is "Comedy," then the node is initialized with the GloVE embedding of the word "Comedy." The user embeddings are randomly initialized.

Using a KG for generating embeddings helps in capturing meaningful user-user and movie-movie relations which helps with personalization in the recommendation tasks. As similar users tend to

[1] https://github.com/xiangwang1223/neural_graph_collaborative_filtering
[2] https://keras.io/examples/structured_data/movielens_recommendations_transformers/
[3] https://github.com/jiwidi/Behavior-Sequence-Transformer-Pytorch

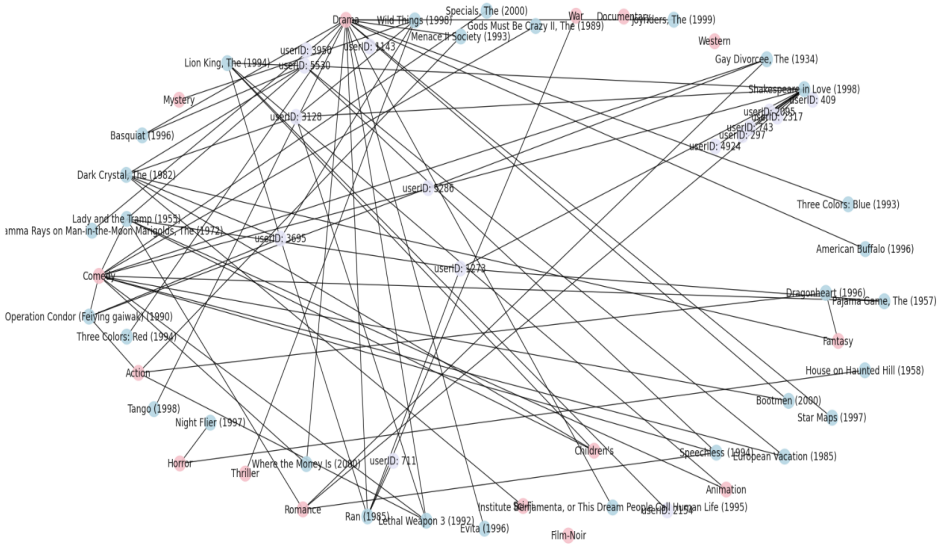



**Figure 2: Generated Knowledge Graph**

watch similar movies, the similarities between users and movies can be captured using the KG.

*5.2.2 NGCF Model.* Similar to the baseline model, we train an RGCN model (instead of a GCN) for generating the user and movie embeddings from the KG using the NGCF framework. The RGCN model learns different sets of weights for different edge types, instead of learning one set of weights for all the edges. By utilizing an RGCN, the model is able to reason more effectively about the complex relationships between entities in the Knowledge Graph, leading to improved performance on tasks that require relational reasoning. In particular, this will eventually help with the user cold-start problem by allowing us to explore movie-movie relations as we have a lack of user-user and user-movie information as we will see in our final results. For this project, we use a 2 layer message passing RGCN model, with hidden dimensions of size 50, trained on the rating prediction task. We use the AdamW optimizer and the RMSE Loss for optimizing the model parameters.

*5.2.3 Behavior Sequence Transformer.* The architecture of the BST is similar to that of the baseline BST. However, in TransRecG model the user and movie embeddings are initialized with their corresponding node embeddings from the NGCF. Therefore, the benefits are two fold: (1) The model can adapt to the varying preferences of the user by analysing the previous movies the user has watched, and finding similar movies using the movie embeddings learnt from NGCF. For example, a user who has watched "Harry Potter 1" (HP1) could be recommended HP2, as many users watch HP2 after HP1 (from BST model), and the movies HP1 and HP2 are very similar (from NGCF model). (2) The model can personalize recommendations by using the user embeddings to understand user preferences. For example, some users might have watched HP1 and HP2 because they prefer watching fiction, whereas other users might have

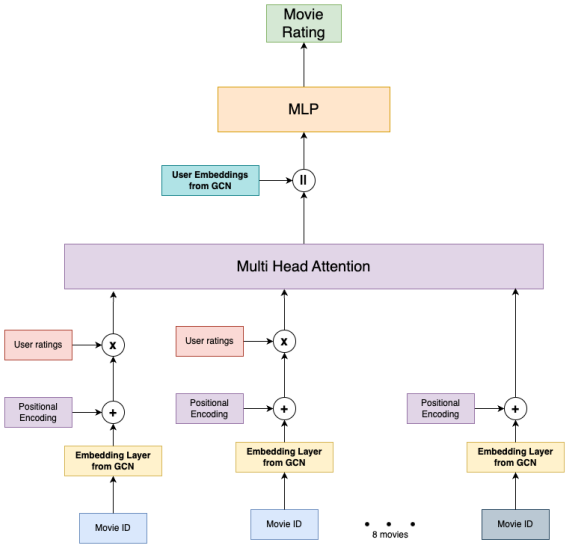



**Figure 3: Behavior Sequence Transformer**

watched it for "Daniel Radcliffe." The architecture of TransRecG's BST is shown in Figure 3.

Node embeddings from the NGCF model of size 50 are passed through a linear layer to get an embedding of size 36. The node embeddings are then concatenated with the BST embeddings learnt from user/movie metadata, before being passed to the BST. The Transformer uses 9 attention heads to learn temporal information from the movie embeddings. The output of the the Transformer model is then merged with the user embeddings before being passed

**Table 1: Performance of Various Models**

| Model | Description | MAE |
|---|---|---|
| GRU | No pre-trained embedding. Sequence size 8 | 0.792 |
| BST-2 | No embedding from GCN. Sequence size 2 | 0.782 |
| BST-4 | No embedding from GCN. Sequence size 4 | 0.767 |
| BST-8 | No embedding from GCN. Sequence size 8 | 0.760 |
| BST-15 | No embedding from GCN. Sequence size 15 | **0.754** |
| NGCF-BP | GCN on a User-Movie Bipartite Graph | 1.148 |
| NGCF-KG | RGCN on a User-Movie-Attribute KG | 1.095 |
| BST-8 + NGCF-BP | Embeddings from the NGCF on Bipartite Graph | 0.746 |
| **TransRecG** (BST-8 + NGCF-KG) | Embeddings from the NGCF on User-Movie KG | **0.741** |

to a 5 layer MLP with layers of size 913, 1024, 512, 256 and 1. The output of the MLP is the predicted rating, and is the output of the TransRecG model. The BST and MLP model parameters are optimized on this rating prediction task using an RMSE Loss, AdamW optimizer with a learning rate of 5e-4 and a batch size of 256.

In summary, the TransRecG model is able to cater to the varying preferences of the users, and also provide personalized recommendations, by capturing sequential, and higher-order relations between users and movies.

## 6 EXPERIMENTS AND RESULTS

We report the Mean Absolute Error (MAE) for the rating prediction task for the models. The performance of various models was compared and analyzed to understand their strengths and weaknesses. This project explores the ability of different models to handle the cold-start problem, and also analyzes the trade off between the number of training data points and performance of the model.

### 6.1 Model Comparison

The performance of different models are summarized in Table 1. A unidirectional GRU model without attention is used for establishing the baseline for a sequential RS. All the BST and NGCF models are implemented as described in Section 5.1.2 and Section 5.1.1 respectively. Different sets of sequence lengths were analysed for the baseline BST model. Then, we implement the proposed TransRecG model as described in Section 5.2. We also analyse the performance of the NGCF (with RGCN) model on the KG without a BST, and the performance of the BST with node embeddings from NGCF on the user-movie bipartite graph.

- Sequence based models (GRU and BST) perform better than non-sequence based models (NGCF). Even the baseline GRU model outperforms the NGCF techniques. Hence, we can conclude that sequential recommendation systems are more powerful that graph based model, especially when initial knowledge about the graph entites are limited.
- While analysing the impact of sequence lenghts on the BST model performance, we found that longer sequences lead to better performance.
- Both the graph embedding augmented BST models (BST-8 + NGCF-BP and TransRecG) outperformed all BST and NGCF models. Therefore, higher order user-user, and movie-movie connectivity information helps in improving recommendations.
- TransRecG with graph embeddings from RGCN + KG performs better than GCN + BP models. The nodes are initialized with the adjacency matrix in the GCN + BP models, whereas the nodes are initialized using GloVE embeddings in the RGCN + KG model. Therefore, more meaningful node initializations lead to better performance.

### 6.2 Cold Start Problem

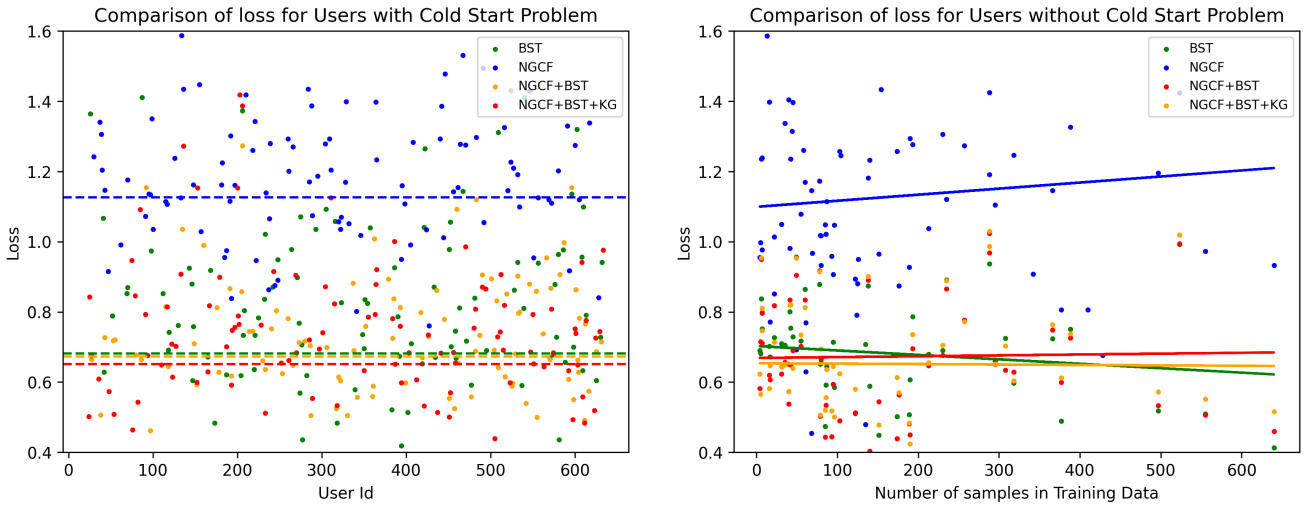
The comparison of the loss of the models for the users having cold start problem is shown in Figure 4(a). The dotted lines show the mean loss of the model. The NGCF model considers one user and one movie at a time during the rating prediction task. As there exist no information about cold start users in the training dataset, the model does not learn any meaningful user embeddings. Therefore, during the test time the NGCF model has the worst performance. The sequence based BST models perform much better than the NGCF model, as they are able to leverage the sequential information given as input to generate good recommendations, even without proper user embeddings.

Adding graph embeddings to the BST model further reduces the loss as movie-movie relations are learnt through the 2 layer message passing network, and this helps in improving the recommendations. Interestingly, the proposed TransRecG model achieves the best performance as it uses node embeddings learnt from the KG which are initialized using GloVE embeddings.

### 6.3 Tradeoff between training samples and test loss

The comparison of the loss of the models for the non cold start users is shown in Figure 4(b). In this figure, the number of samples denotes the number of data points the user has in the training set. The lines represent the trends of the loss of the various models. We observe that the NGCF loss increases for users with more data points in the training dataset. As users with large number of data points tend to have varied preferences, and as the NGCF model fails to capture the dynamic preferences of the users (no sequential information), the test loss increases for users with more traning data points.

In contrast, for the BST model the test loss decreases with increase in training data points. This shows that the BST model is able




(a) Loss Comparison for users with Cold Start Problem

(b) Tradeoff between training data size and test loss

**Figure 4: Model Performance Analysis for cold start and non-cold start users**

to capture the dyanmic preferences of the user, and with more data points its able to learn more about sequential order of the movies, as well as the users. However, the performance of the BST models augmented with graph embeddings almost remains unchanged. As we directly concatenate the user and movie embeddings from the poorly performing NGCF model (for users with large number of data points), this affects the performance of our models. We believe that adding an attention based model for weighting the graph and metadata based embeddings before concatenating them can improve the performance of the model.

## 7 CONCLUSION

In this project we built a Transformer based graph augmented RS that can leverage the temporal information and higher order connectivity features for providing personalized, dynamic and relevant recommendations.

Potential future works: (1) Currently, the BST model uses a sequence length of 8 due to limited compute. For upto sequence length of 15, we found that model performance improves with increasing sequence lengths. However, the effect of even longer sequence lengths should be further analysed. (2) We implement the Knowledge Graph (KG) as a heterogeneous graph only in terms of the edges. Creating the KG as a fully heterogeneous graph with both different types of nodes and edges, is a relevant future work as user-movie-attribute graphs are composed of different types of entities in nature. (3) Understanding the effect of TransRecG model performance with attention weights for balancing between the graph and meta data embeddings, especially for users with large number of training data points.

## CONTRIBUTION

- Sai Prasath Suresh: NGCF, RGCN, BST, TransRecG, Analysis and comparison of Model performances, Presentation, Report
- Ishwarya Sivakumar: data processing, BST, TransRecG, Analysis and comparison of Model performances, presentation, report
- Jeongjin Park: raw data statistics, assisted in data pre-processing, knowledge graph and GloVE, presentation file
- Balaram Behera: Baseline NGCF, assisted in reviewing presentation

## REFERENCES

[1] Qiwei Chen, Huan Zhao, Wei Li, Pipei Huang, and Wenwu Ou. 2019. Behavior sequence transformer for e-commerce recommendation in alibaba. In *Proceedings of the 1st International Workshop on Deep Learning Practice for High-Dimensional Sparse Data*. 1–4.

[2] F Maxwell Harper and Joseph A Konstan. 2015. The MovieLens datasets: History and Context. *Acm transactions on interactive intelligent systems (tiis)* 5, 4 (2015), 1–19.

[3] Lei Sang, Min Xu, Shengsheng Qian, and Xindong Wu. 2021. Knowledge graph enhanced neural collaborative filtering with residual recurrent network. *Neurocomputing* 454 (2021), 417–429.

[4] Walid Shalaby, Sejoon Oh, Amir Afsharinejad, Srijan Kumar, and Xiquan Cui. 2022. M2TRec: Metadata-aware Multi-task Transformer for Large-scale and Cold-start free Session-based Recommendations. In *Proceedings of the 16th ACM Conference on Recommender Systems*. 573–578.

[5] Guy Shani and Asela Gunawardana. 2011. Evaluating recommendation systems. *Recommender systems handbook* (2011), 257–297.

[6] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. KGAT: Knowledge Graph Attention Network For Recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 950–958.

[7] 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*. 165–174.

[8] He Zhang, Xingliang Yuan, Quoc Viet Hung Nguyen, and Shirui Pan. 2023. On the Interaction between Node Fairness and Edge Privacy in Graph Neural Networks. *arXiv preprint arXiv:2301.12951* (2023).