IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 27 October 2023, accepted 21 December 2023, date of publication 25 December 2023,
date of current version 29 December 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3346918

RESEARCH ARTICLE

A Flexible Two-Tower Model for Item Cold-Start Recommendation

WON-MIN LEE AND YOON-SIK CHO
Department of Artificial Intelligence, Chung-Ang University, Seoul 06974, South Korea

Corresponding author: Yoon-Sik Cho (yoonsik@cau.ac.kr)

This work was supported in part by the Institute of Information and Communications Technology Planning and Evaluation (IITP) Grant funded by the Korean Government (MSIT), Artificial Intelligence Graduate School Program of Chung-Ang University, under Grant 2021-0-01341; and in part by the National Research Foundation of Korea (NRF) Grant funded by the Korean Government (MSIT) under Grant 2021R1F1A1063389.

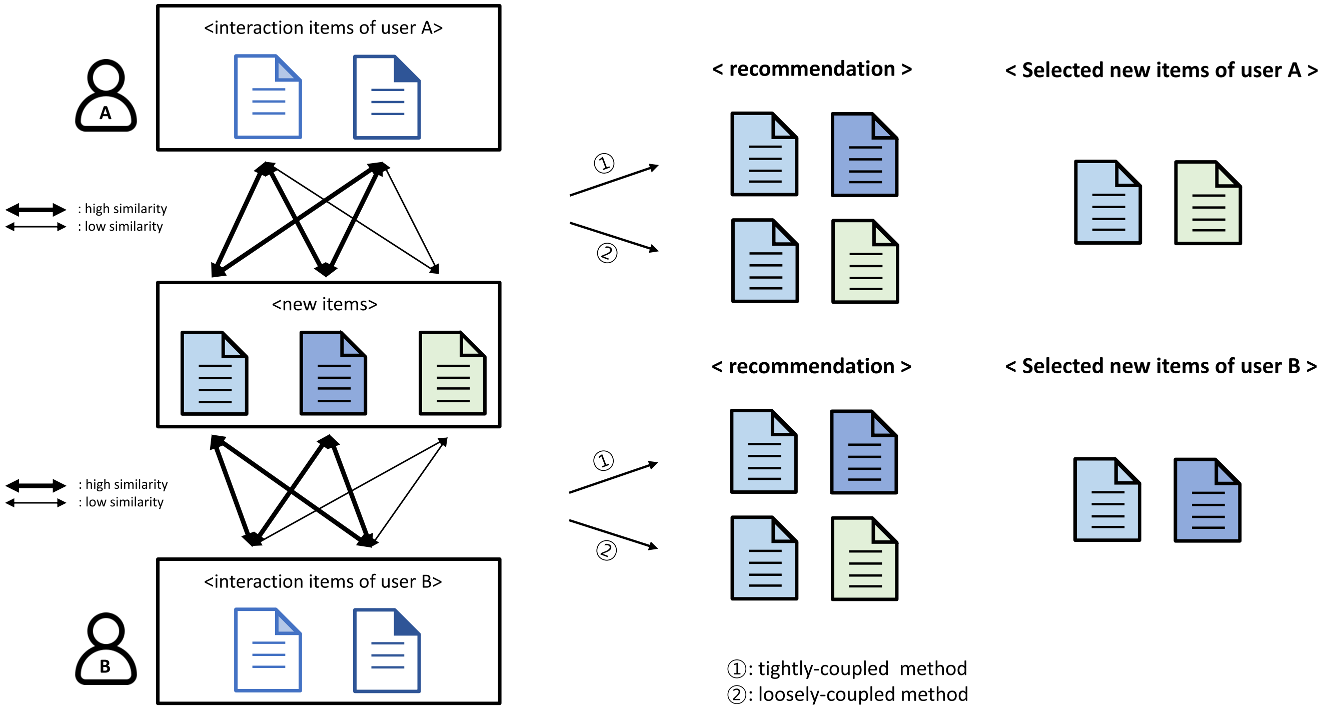
ABSTRACT One of the main challenges in recommendation system is the item cold-start problem, where absence of historical interactions or ratings in new items makes recommendation difficult. In order to solve the cold-start problem, hybrid neural network models using meta data of the item as a feature is widely used. However, existing cold-start models tend to focus too much on utilizing the side information of items, which may not be flexible enough to capture the interaction information of users. In this study, we propose a flexible framework for better capturing the interaction information of users. Specifically, we incorporate the multiple choice learning scheme into the two-tower neural network which is a popular recommendation model that consists of two towers - one for users and one for items. In our proposed framework, we construct two encoders. One of the two encoders, the *tightly-coupled* encoder, focuses on the side information of items with which the user has interacted, the other one, *loosely-coupled* encoder, focuses the user's interaction information. We utilize Gumbel-Softmax to stochastically select the encoder, enhancing the flexibility that considers not only item feature but also user interaction information. We evaluate our proposed framework on two datasets - the MLIMDb dataset which is a combination of widely used the MovieLens and IMDb datasets based on common movies, and the CiteULike dataset. The experimental results show that our proposed framework achieves state-of-the-art results on cold-start recommendation. In the Recall@150 experiments on the CiteULike dataset, we achieved improvement of approximately 2.7% compared to the base model. In the Recall@150 experiments on the MLIMDb dataset, we achieved improvement of approximately 5.2% compared to the base model. We further show our proposed model improves the performance in the warm-start settings. In the Recall@100 experiments on the Citeulike dataset, we observed an improvement of approximately 1.3% compared to the base model. In the Recall@100 experiments on the MLIMDb dataset, we observed an improvement of approximately 3.9% compared to the base model. Our proposed framework provides a flexible approach for capturing the diverse aspects of users in recommendation systems, even for cold-start items. As demonstrated through extensive experiments, our proposed model outperforms several State-Of-The-Art (SOTA) models on both datasets.

INDEX TERMS Recommendation system, cold-start problem, hybrid neural network.

I. INTRODUCTION

The Recommendation System (RS) is used in various fields, which is beneficial in many ways [1]. It generates revenue by providing high-quality products, saves users' time and efforts when finding items of their interest, and facilitates access to new products. RS in the industry has attracted a great deal of attention in the past few years and consequently many algorithms have been proposed. Research in this field has been active in recent years. In [2], [3], and [4] authors propose algorithm to understand user needs within the context of user-system dialogues and provide recommendation. These

The associate editor coordinating the review of this manuscript and approving it for publication was Wanqing Zhao.



<interaction items of user A>
A
: high similarity
: low similarity
<new items>
: high similarity
: low similarity
<interaction items of user B>
B
①
②
①
②
< recommendation >
< Selected new items of user A >
< recommendation >
< Selected new items of user B >
①: tightly-coupled method
②: loosely-coupled method

FIGURE 1. **The recommended flow of cold-start item in our proposed method. Other aspects besides item information should also be considered for cold-start recommendation.**

algorithms require a sophisticated technology that can grasp the context of user conversations while delivering appropriate recommendations. In [5], [6], and [7] authors propose algorithm to recommend the next item for user interaction, relying on information about the sequence in which users have interacted with items. However, one of the main challenges in RS is recommending new items, where no previous interaction or rating records exist. Such a problem is called 'cold-start problem', where many algorithms [8], [9], [10], [11], [12], [13] have been proposed to address this problem.

In various domains of RS, work on solving the cold-start problem using additional feature [14], [15], [16], [17] is actively under way. One of the most popular approach is the hybrid method, which combines the Collaborative Filtering (CF) and Content Based Filtering (CBF) methods. CF predicts unseen interactions between users and items using only feedback history, but it faces the challenge of reduced performance when feedback is sparse. CBF relies on user and item side information for predictions, making it applicable when such information is available. However, it cannot capture user interaction patterns. To address these constraints, hybrid methods combine both approaches, mapping side information and feedback to separate low-dimensional representations and combining them to predict the final interaction. The hybrid method also has been successfully applied in many real-world recommendation systems, such as Amazon and Netflix. One of the main advantages of the hybrid method is that it can effectively utilize the side information for handling the cold-start problem. When implemented in neural networks, two-tower approach [17], [18], [19], [20], [21] is favored for construction hybrid models. In two-tower neural networks, two encoders are employed separately for learning representations of users and items. In the item encoder, the feature of the item is used as an input and transformed into a lower dimensional representation. In the user encoder, the information of user is used as an input and transformed into a lower dimensional representation. Under this framework, information about the item or the user can be effectively exploited, and thus have proven to be effective in cold-start scenarios.

The recently introduced method [18] based on this two-tower RS has further improved the performance for item cold-start recommendation. In [18], the item representation was directly shared into the user encoder in an attempt to better unify the two representations, which achieved state-of-the-art results for item cold-start recommendation. While this approach might be effective in recommending items with similar features, we believe this over-constrained setting is limited beyond item features. As pointed in previous study [15], a user who only reads news in sports may still be interested in news on crime, and we believe that relying excessively on the similarity of side information, as in the approach in [18], leads to diminished performance.

In this paper, we study the item cold-start problem addressing the aforementioned limitation through improving the flexibility of user representation, which is achieved through the additional user encoder for capturing *loosely* *coupled* representations. Our proposed scheme has an item encoder, and two user encoders, where one of the user encoders shares the item embeddings as in [18], and the other user encoder does not share the item embeddings but more focuses on interaction information of users. We refer to this scheme as the *'Flexible Two-Tower Model'*. We take inspiration from the *multiple choice learning* scheme [22], which can choose the best performing user encoder at each user-item interaction. Specifically, we use Gumbel-Softmax for stochastically selecting the promising method at each interaction. By selecting two user encoders, we can focus on item feature information or interaction data, depending on the user, which allows us to make better recommendations. Thus, in the training phase, we expect the tightly coupled user encoder to focus on capturing the item features, while the loosely coupled user encoder can focus on the relationship between user and items. We conduct our experiments on completely cold-start items which hasn't been seen in the training phase. Figure 1 illustrates how our proposed scheme can capture these two scenarios. In Figure 1, the items in the new items and interaction items indicate that the blue color range items are similar to each other, and the green color range item is dissimilar to the blue color range items. Through the list of interaction items of user A and user B, we can see that the common items that both users interacted with have features closer to the blue color range than the green color range. However, in reality, user B only selects items with blue features, similar to previous interaction pattern, while user A deviates from previous interaction pattern and selects not only the item with blue features but also item with green features among the new items. As shown in this example, users may actually select new items with features that they did not interact with before or only a few times. If a recommendation system only recommends tightly coupled items with high similarity, it may lose potential recommendation predictions to some users like user A. In our proposed scheme, we can use a loosely-coupled method that considers the relationship between users and items, in addition to the tightly coupled method.

The evaluation is conducted by comparing the Recall metric with the previous cold-start models. Our experimental results reveal that our proposed method improves the state-of-the-art model in cold-start item recommendation. Moreover, in warm-start settings, our proposed method achieves competitive performance outperforming the previous model [18].

The main contributions of our work can be summarized as follows:

- We propose a flexible design of two-tower recommendation model which can better capture diverse user's behavioral patterns.
- Our proposed training scheme with Gumbel-Softmax stochastically selects the most relevant encoder for each interaction between user and items. In turn, in the training phase, each encoder learns attentive representation.
- Our experimental results support our theoretical claims and also demonstrate that our method empirically achieves state-of-the-art results for cold-start item recommendation.

II. RELATED WORK

RS is used in various fields and many applications suffer from cold-start problem [14], [15], [16], [23], [24]. To address this problem, researchers have proposed various approaches [25]. Specifically, hybrid models have been widely used in the field of deep neural networks [17], [18], [19], [20], [21], [26], [27], [28], which leverage side information of users or items. The two-tower model is a popular hybrid model in neural network [17], [18], [19], [20], [21].

A. COLD START PROBLEM IN VARIOUS DOMAINS

The cold start problem is present in recommender systems from various domains, where side information is often utilized. In the music domain, authors in [14] leveraged music and artist side information in graph autoencoder architecture to effectively incorporate the genre, country, and mood of music into node embedding representations learned from the graph. Moreover, they tackled the cold start artist problem by automatically ranking the top-k most similar neighbors of new artists using a gravity-inspired mechanism. In the news domain, authors in [15] proposed a framework to perform recommendation with personalized user interest and also incorporated news popularity to alleviate the cold-start problem. In their work, entities obtained from news titles, content embedding vectors such as words and click-through rates are used as feature. The personalization score is calculated from the embedding vector obtained from the news title, and the news popularity score is calculated from the news content and click-through rate. These two scores are combined for news recommendations. In the movie domain, [16] proposed an attentive graph neural network model to leverage movie side information such as categories, directors in graph neural network. It highlights the importance of exploiting the attribute graph rather than the interaction graph in addressing strict cold-start problem in neural graph RS. In the online store domain, [24] addresses the cold-start problem in the Next Basket Recommendation System (NBRS). To solve this problem, Authors proposed the model that incorporates various couplings between users and items, ensuring the effective transmission of user/item information. The model improves the Particle Swarm Optimization algorithm to optimize the weights and biases of the Deep Auto-Regressive network for learning heterogeneous couplings across baskets, ultimately recommending the next basket. Furthermore, it integrates the Adaptive Response to Particle Adjustment Strategy (ARPAS) into our framework. Reference [23] addresses the cold-start problem by leveraging item feedback information from various domains, including Movies, Books, Games, and Perfumes where users have interacted with items. The authors propose a cross-domain recommendation network that integrates a sparse local sensitivity mechanism into geometric deep learning algorithms. Additionally, they introduce a local sensitivity adapter to capture crucial local geometric information within the recommendation system's structure.

B. TWO-TOWER NETWORK

In deep neural network literature, hybrid models have been actively used for solving cold-start problems. In [17], [19], [20], and [21], both user and item features are utilized to address the cold start problem. The authors in [17] combined two of two-towers, where one of the two-tower is specialized for computing the inner product of user and item embeddings, and the other two-tower is specialized for computing the prediction score using user and item features. In each tower, the GMF and MLP structures are applied to compute the prediction score, and they are connected in the NeuMF layer to obtain the final prediction score, which solves the cold start problem. In [19] and [21], both models use a user tower that with the user's ID and feature information, and an item tower with the item's ID and feature information. In [21], two stacked denoising autoencoders are used to learn the user and item features. The extracted user feature vector and embedded user ID, as well as the extracted item feature vector and embedded item ID, are concatenated to form the user and item latent vectors. These latent vectors are then fed into a neural network to learn the user-item relationship and generate a predicted rating. A hybrid model in [19] combines content-based and collaborative filtering approach, which can effectively integrate content and past feedback information. The attention mechanism is employed to control the proportion of the two types of information for each user-item pair. Additionally, an adaptive learning strategy called 'cold sampling' is performed to address the cold start problem. Reference [20] consists of a user tower that takes the user's preference and side information as inputs to generate the user's latent representation, and an item tower that takes the item's preference and side information as inputs to generate the item's latent representation. This model focuses on integrating preference and side information, thus exhibits performance degradation when past interaction data is less available. The shared attention model [18] is also based on the two-tower framework. One of the encoders takes users' information of past interaction with the items, and the other takes all item features as inputs. This model is expressed as a shared model because the item representation is shared with first layer of the user encoder. Shared attention model currently achieves state-of-the-art results on item cold-start setting. However, as the recommendations mainly rely on the feature similarity between items, items beyond its feature cannot be recommended effectively. Items that are not similar in features tend to be ignored in recommendations, which may limit to provide novel recommendations [29]. To address this problem, in addition to the method which only uses item feature similarities in a tightly shared manner, we propose a flexible model that can capture user behaviors beyond item similarities in side information.

III. PROPOSED MODEL

A. PRELIMINARIES

In this paper, a set of M users is denoted as $\mathcal{U} = \{u_1, \ldots, u_M\}$ and a set of N items is denoted as $\mathcal{V} = \{v_1, \ldots, v_N\}$. $\mathbf{R}$ denotes the $M \times N$ interaction matrix with implicit feedback; when user m interacted with item n in the past, $\mathbf{R}_{mn} = 1$, otherwise $\mathbf{R}_{mn} = 0$. Side information can be exploited for recommendation with implicit feedback, which can be from items or users. Due to privacy concerns, item side information is often used in the literature. We use $N \times D$ item side information $\mathbf{X}$ as feature in our study for RS, where D is the dimension of feature of each item. In item cold-start recommendation, the goal is to predict whether users will like a new item based on the available data without any past information, where two-tower model is often used.

Hybrid approaches can effectively combine the advantages of collaborative filtering and content-based RS. In neural network, two-tower is an intuitive structure for feeding side information of users or items. Two-tower network consists of two encoders for generating user representations and item representations, where we denote item encoder as $g^{item}(\cdot)$ and user encoder as $g^{user}(\cdot)$. Following the implementations in [18], each encoder is a Multi-Layer Perceptron (MLP) using Hyperbolic Tangent (tanh) function as activation function. Once the representations are generated, the dot product of the user representation and the item representation is compared with the ground-truth positive or negative pairs in training phase.

Using the two-tower framework, [18] proposed a shared model to effectively solve the cold-start item recommendation. In their approach, the item representation from the item encoder is shared with the hidden item representation of the user encoder. The user encoder in [18] takes the user's past interactions with the items as its input, instead of the user information; the item encoder takes the item features as its input. The item representation and user representation can be obtained as follows:

$$\mathbf{z}_n^{item} = g^{item}(\mathbf{X}_n), \tag{1}$$

$$\mathbf{z}_m^{user} = g^{user}(\mathbf{R}_{m,:}), \tag{2}$$

where the first layer of g^{user} is directly borrowed from the item representations in shared model. Reference [18] applied attention mechanism to the input of user encoder for further performance improvement. However, the shared model tend to focus too much on the item features similarities, and thus can lose flexibility. Throughout the paper, we refer the model proposed in [18] as *tightly coupled* model.

B. PROPOSED SCHEME FOR ENHANCING FLEXIBILITY

Motivated by aforementioned limitations, we propose a novel ensemble approach of two-tower network by having additional user encoder. We resort to multiple-choice learning [22] scheme for training an ensemble of two (multiple)

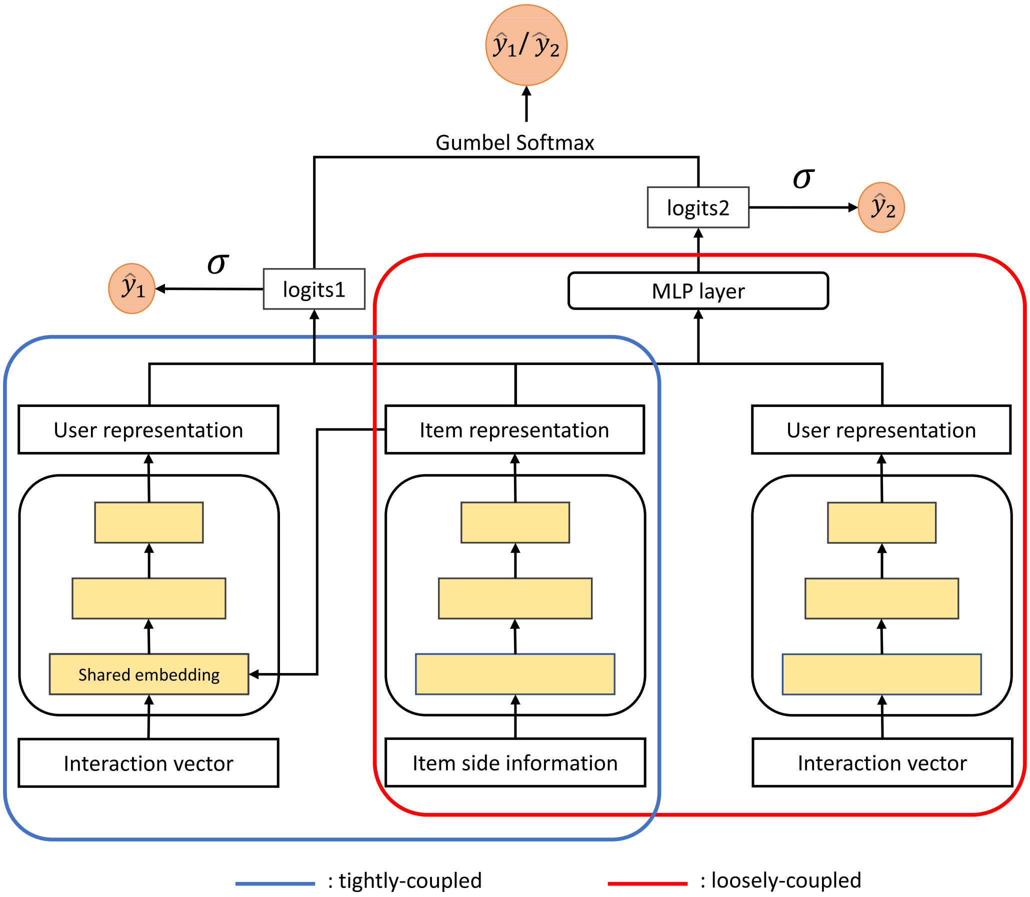


FIGURE 2. **Overall structure of proposed model.**

models, where the overall structure of our model is provided in Figure 2colorblue. In order to improve the limitations of the tightly coupled model from [18], we add additional model, namely, *loosely coupled* model. As the naming implies, the layer of user encoder in loosely coupled model does not share any information of item representations. We expect the user encoder in loosely-coupled model can further capture the user representations that can be missed in the strict assumption in tightly coupled model. As can be seen in Figure 2, while each item generates its own item representations, each user generates *tight* user representations and *loose* user representations from different encoders. At each user-item interaction, the multiple-choice learning scheme selects the most relevant user representations from the two. We defer the details of our multiple-choice learning scheme to the following section.

As in [18], we use Multi-Layer Perceptron (MLP) with Hyperbolic Tangent (tanh) activation function for all encoders in our model. While the item encoder is denoted as $g^{item}(\cdot)$, we differentiate the user encoders as encoder1:$g^{user1}(\cdot)$ for tightly-coupled, and user encoder2: $g^{user2}(\cdot)$ for loosely coupled. The item encoder takes D-dimensional item features as input and obtains a low-dimensional item representation through $g^{item}(\cdot)$ in Equation 1. Thus, for user representations, we generate using the following encoders:

$$\mathbf{z}_m^{user1} = g^{user1}(\mathbf{R}_{m,:}), \tag{3}$$

$$\mathbf{z}_m^{user2} = g^{user2}(\mathbf{R}_{m,:}), \tag{4}$$

where the encoder $g^{user1}(\cdot)$ is borrowed from the encoder $g^{user}(\cdot)$ in the shared model [18], and the encoder $g^{user2}(\cdot)$ is a three-layered MLP with all trainable weights in each layer.

From tightly coupled model, the output $\hat{y}_1 \in \{0, 1\}$ is predicted using the sigmoid function.

$$\Pr(\hat{y}_1 = 1) = \sigma(\mathbf{z}^{user1} \cdot \mathbf{z}^{item}), \tag{5}$$

where σ is the sigmoid function, and the logit in sigmoid is the inner product of the user and item representations. Likewise, the output of loosely coupled model can be obtained as below:

$$\Pr(\hat{y}_2 = 1) = \sigma(g^r(\mathbf{z}^{user2} \cdot \mathbf{z}^{item})), \tag{6}$$

where the MLP layer $g^r(\cdot)$ is additionally added for capturing the complex relationship between user-item interactions. We empirically found that the extra MLP layer in the loose coupled model contribute to further performance improvement. Finally, the model generates the final output from $\hat{y_1}$ and $\hat{y_2}$, where the final output is selected from one of the two. This ensemble approach offers great flexibility by allowing us to combine different assumptions. Thus, recommendation beyond item side information can still be conducted, and vice versa. In the following, we present the details of our ensemble scheme.

C. MULTIPLE CHOICE LEARNING WITH GUMBEL-SOFTMAX

We combine the tightly coupled model and loosely coupled model through multiple-choice learning [22] scheme, where item encoder is fixed and user encoder is selected stochastically. Multiple-choice learning outputs the multiple hypotheses, unlike traditional learning methods, and chooses the most plausible solution for each data instance (or user-item interaction in our context). Specifically, we select a user encoder using Gumbel-Softmax [30], where Gumbel-Softmax can be viewed as channel selector. Here, the Gumbel-Softmax trick [30] makes the decision process differentiable. We let the two logits from the two encoders compete each other through the equation below. As a result, we proceed with the training by stochastically selecting the encoder. Gumbel-Softmax aids in encoder selection, contributing to the performance improvement of our model. It yields better results compared to using a regular Softmax, as demonstrated in the Results section. For every interaction between user m and item n, we obtain representations of tightly-coupled and loosely-coupled encoders through the following Equation 1, 2, 3, 4,

$$\mathbf{c}_{mn} = \text{Gumbel\_Softmax}(\mathbf{z}_m^{user1} \cdot \mathbf{z}_n^{item}, g^r(\mathbf{z}_m^{user2} \cdot \mathbf{z}_n^{item})), \tag{7}$$

based on [22], we select the encoder through Equation 7. $\mathbf{c}_{mn}$ is an one-hot indicator vector for interaction between user m and item n and describes the sampled model.[1] Gumbel-Softmax have proven to be effective in many domains when sharp mapping is preferred [31], [32], [33], [34]. As such, we use the Gumbel-Softmax as the gating mechanism between the two user encoders: tight vs loose. Thus, we can have the final output of our model as below.

$$\hat{y} = \begin{cases} \hat{y}_1, & \text{if } \mathbf{c} = [1, 0] \\ \hat{y}_2, & \text{if } \mathbf{c} = [0, 1] \end{cases} \tag{8}$$

Through this proposed framework, we can capture users' diverse behavioral aspects. In the forward-process (or generative process), Gumbel-Softmax stochastically selects one of the user encoder for generating user-item interaction. The generated signal can be triggered by one of the multiple assumptions, where we have two in this study.

D. TRAINING PROCESS

The overall training process of our proposed model is provided in Algorithm 1. Each user obtains two user representations from tightly coupled model and loosely coupled model. Each item also obtains its representation through the item tower. The probability of user-item interactions are predicted independently from the two scenarios, which is reserved as P_1 and P_2. The final probability off a given user-item interaction is selected from the reserved two prediction with respect to the sampled indicator $\mathbf{c}$. The final output is compared with the ground-truth, and each user encoder and a item encoder gets updated through backpropagation. Gumbel-Softmax trick allows the model to be trained in end-to-end.

Algorithm 1 Training Process

```
Input: $\mathbf{R}$, $\mathbf{X}$
Output: $\hat{\mathbf{y}}$
Initialize: user encoders $g^{user1}(\cdot)$, $g^{user2}(\cdot)$, and item encoder $g^{item}(\cdot)$
$\mathbf{z}^{user1}$: representation of user in encoder 1 (tight)
$\mathbf{z}^{user2}$: representation of user in encoder 2 (loose)
$\mathbf{z}^{item}$: representation of item
for epoch do
    for Iters $I$=(number of trainset) / batch size do
        1. Representation
            $\mathbf{z}^{user1} = g^{user1}(\mathbf{R})$
            $\mathbf{z}^{user2} = g^{user2}(\mathbf{R})$
            $\mathbf{z}^{item} = g^{item}(\mathbf{X})$
        2. Probability
            $P_1 = \sigma(\mathbf{z}^{user1} \cdot \mathbf{z}^{item})$
            $P_2 = \sigma(\mathbf{g}^{\mathbf{r}}(\mathbf{z}^{\mathbf{user2}} \cdot \mathbf{z}^{\mathbf{item}}))$
        3. User encoder selection
            $\mathbf{c}$ = Gumbel-Softmax($\mathbf{z}^{user1} \cdot \mathbf{z}^{item}$, $g^r(z^{user2} \cdot \mathbf{z}^{item})$ )
            $P(y = 1) = \mathbf{c}\,[P_1, P_2]^\intercal$
        4. Backpropagation
            update $g^{user1}(\cdot)$, $g^{user2}(\cdot)$, $g^{item}(\cdot)$
    end for
end for
```

IV. EXPERIMENTS

In this section, we provide an explanation of the experimental setup and baseline models used in the experiments conducted to evaluation the effectiveness of our proposed 'Flexible Two-tower' model in addressing the item cold-start problem.

A. EXPERIMENTAL SETUP

1) DATASET

For the evaluation, we use two datasets:

1) CiteULike [35]. This dataset has been widely used in previous studies [19], [20], [36], [37], [38] on the cold-start problem. The dataset has been originally introduced in collaborative topic modeling [35], where the probabilistic topic modeling was introduced for recommending scientific articles to users. The dataset contains 204,987 user-article (or item in the RS context) interactions from 5,551 users across 16,980 articles (items), where the interaction matrix has a sparsity of 99.8%. In the interaction matrix $\mathbf{R}$, $\mathbf{R}_{mn} = 1$ means that user m has saved article n in his

[1] In our experiments, we set the Gumbel-Softmax temperature τ to 0.7.

internet library, and $\mathbf{R}_{mn} = 0$ otherwise. Along with the interaction matrix, the title and abstract have also been used for the task. The authors in [35] concatenated the title and abstract on each article, and performed tf-idf to choose the top 8,000 vocabulary. Afterwards, the dimension of item features was fixed at 300 and has been widely used in many studies. Out of 16,980 items, 3,396 items have been reserved for cold-item recommendation in [20], which we use the same train-test split in our study as in [18], [20], and [35]. The items which have been reserved for testing are completely cold-items, where no previous interaction records are available. These cold-items are also regarded as a new item to the RS. In this study, we additionally conduct an experiment for warm item, which we have 2,264 items for testing. The number of warm start items were chosen to have reasonable number of items for training and testing.

2) MLIMDb. This dataset consists of MovieLens and IMDb data, which are commonly used for researching and developing recommendation systems, including the cold-start problem [39], [40], [41], [42], [43], [44]. The MovieLens 100K dataset contains rating information of 1,682 movies evaluated by 943 users. The IMDb dataset contains information on over 300,000 movies. In previous studies [16], [45], [46], two datasets, MovieLens and IMDb, are combined to obtain side information. Similarly, we combine the two datasets to have interaction matrix and item (movie) side information. We name our dataset as MLIMDb. In the MovieLens dataset, user IDs and movie IDs were used for interaction matrix, while in the IMDb dataset, information on movie directors, writers, actors, and genres were used for item feature matrix. We used the director, writer, actor, and genre information of the movie as item features, treating the unique ids of each entity in each category as a single word in a sentence. For example, in the MLIMDb dataset, the movie 'Toy Story' has John Lasseter as the director, Pete Doctor, Andrew Stanton, Joe Ranft, Joss Whedon, Joel Cohen, Alec Sokolow as writers, Tom Hanks, Tim Allen, Don Rickles, Jim Varney as actors, and Adventure, Animation, Comedy as genres, and their unique ids are nm0005124, nm0230032, nm0004056, nm0710020, nm0923736, nm0169505, nm0812513, nm0000158, nm0000741, nm0725543, nm0001815, Adventure, Animation, and Comedy respectively, then the item feature for Toy Story would be 'nm0005124 nm0230032 nm0004056 nm0710020 nm0923736 nm0169505 nm0812513 nm0000158 nm0000741 nm0725543 nm0001815 Adventure Animation Comedy'. Then, we applied tf-idf to the bag-of-words from the item feature. The dimension is 1303, using only words that appear more than twice. The combined dataset based on the common movies includes 943 users, 1,146 items, and user-item interaction matrix is 91.0% sparse. In the interaction matrix $\mathbf{R}$,$\mathbf{R}_{mn} = 1$ means that user m has interacted with movie n, and $\mathbf{R}_{mn} = 0$ otherwise. Out of 1,146 items, 229 items were reserved for cold item recommendation. 187 items were reserved for warm item recommendation. The number of cold and warm start items was chosen to provide a reasonable number of items for training and testing.

Throughout our set of experiments, we fix the testing data to evaluate performance on cold and warm start items across all the models for fair comparison in each dataset.

2) IMPLEMENTATION DETAILS

We implement our proposed model using PyTorch 1.6.0. We set the mini-batch size to 32, the max epoch is set to 50. The learning rate is 0.001 in the CiteULike dataset and 0.0001 in the MLIMDb dataset. The Gumbel-Softmax temperature was fixed to 0.7 throughout the training. The item and tightly user encoder use MLPs with three layers, where the dimensions are [250, 200, 100], and loosely user encoder is [1000,500,100] for the CiteULike dataset. In the MLIMDb dataset, the MLP layer dimensions for all encoders are set to [1000, 500, 100]. Suitable dimensions were tried with respect to the size of the feature dimension. As previously mentioned, we use the same train-test split as in [18]. In the train set, the ratio of positive(1) and negative(0) interactions between users and items is 1:10. when the train data set unbalanced, our model can be biased toward a large class, which adversely affects the training [47]. To address this problem, we use re-weighting scheme [48], which uses the effective number of samples for each class to re-balance the loss. In our experiment, we integrate it to Binary Cross Entropy Loss (BCELoss). When the ith sample of the batch belongs to positive or negative class, the sample weight is calculated as follows:

$$W_i = \begin{cases} \frac{1}{\frac{1-\beta^{n^+}}{1-\beta}} & \text{if belongs to positive interaction} \\ \frac{1}{\frac{1-\beta^{n^-}}{1-\beta}} & \text{if belongs to negative interaction} \end{cases} \tag{9}$$

where the n^+ and n^- are the number of samples of the positive interactions and negative interactions respectively in a batch. The β is a hyperparameter which we fixed to 0.99. All experiments of our model are conducted using a NVIDIA RTX A5000 GPU.

3) EVALUATION

For our evaluation, we use CiteULike, MLIMDb datasets. CiteULike have been used widely in the literature, while MLIMDb dataset is the newly introduced datasets in our study. CiteULike dataset consist of 16,980 items of which 3,396 are reserved for cold-start evaluation. This 4:1 split is the standard split, where training/test dataset have been fixed across all the studies for fair comparison. To tackle the problem in more challenging scenario, we additionally perform in 3:1 setting by adding 849 more items in the previous testing data. We also want to evaluate our model in warm-start setting, where we test the 2,264 fixed warm-items to compare with [18]. We constructed the MLIMDb dataset in a similar way to the CiteULike dataset. Similar to the evaluation using the CiteULike dataset, we perform item cold-start evaluation on two set of test data: 4:1 split and 3:1 split. For 4:1 split, we have 229 movies for testing, and for 3:1 split, we have 286 movies for testing. To perform evaluation on warm-start items, 187 items were used as test data.

We use Recall as our evaluation metric. This is one of the commonly used metrics in many recommendation system research to assess how well the proposed model predicts the items that users actually interact with. We demonstrate the effectiveness of our proposed model through various evaluations for the top 20, 50, 100, and 150 recommended items.

B. BASELINES

We compare the proposed model with the following strong baseline models for cold-start RS.

- **Shared Model (−attention)** [18] is a two-tower network for RS, where the item embedding from the item encoder is shared to the user encoder for cold-start item predictions.
- **Shared Model (+attention)** [18] further improves the shared model by applying attention mechanism to the objective function.
- **SimPDO** [49] utilizes the objective function to train a one-class recommendation system model, which effectively solves the item cold start problem.
- **NeuMF** [50] combines Generalized Matrix Factorization (GMF) and Multi Layer Perceptron (MLP). By combining the linearlity of MF and the non-linearity of DNN, the correlation between user and item can be learned.
- **DropoutNet** [20] tackles the cold-start problem through training the model to reconstruct the input from the corrupted version.
- **ACCM** [19] tries to take both advantages of content-based and collaborative filtering in an attention-based unified approach.
- **DeepMusic** [36] uses a latent factor model for recommendation, and predicts the latent factors from music audio when it cannot be obtained from usage data.
- **CTR** [37] is a collaborative filtering based on probabilistic topic modeling, where the probabilistic topic modeling allows the interoperability on item content information.
- **CDL** [38] is a hierarchical Bayesian deep learning model, which jointly performs deep representation learning on item content information and collaborative filtering for the feedback.

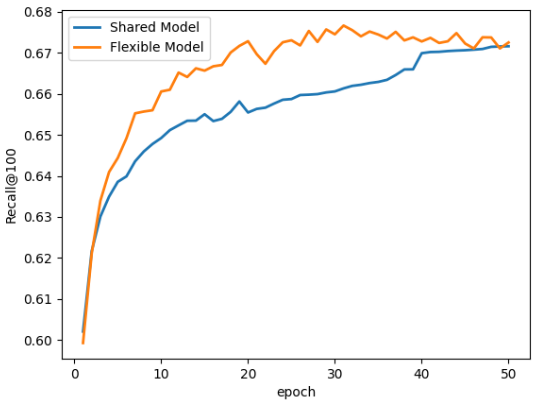


FIGURE 3. Comparison of Convergence Graphs for training up to epoch 50 on Recall@100 for the CiteULike dataset.

V. RESULTS

We evaluate our proposed model in many ways. First, we compare the performance of our proposed model to the previous State-Of-The-Art (SOTA) model. Second, we justify our use of Gumbel-Softmax through the empirical results. Third, we use the CiteULike dataset and present the full performance comparison with the strong baselines. Forth, we show how our model performs in warm-start setting. Finally, we show that our multiple-choice learning scheme with Gumbel-Softmax fully uses the two modules not only focusing only on one of the two.

A. COMPARISON AGAINST SOTA MODELS

We use offline test Recall@K as our evaluation metric and report the average Recall@K to evaluate the proposed model, where K is set to 20, 50, 100, and 150. Table 1 provides the performance comparison between our proposed model and other base models including the previous SOTA model (shared model). We also report the results from None-shared model which is separate loosely coupled model. Two sets of experiments were conducted under two settings; one from the standard setting, and the other from the challenging setting. For standard setting, we have 4:1 (train:test) split. For challenging setting, we have 3:1 (train:test) split. These two sets are reported in Table 1a and Table 1b. We also found that the previous SOTA model can be further tuned for achieving higher performance than the performance from the original paper [18], and we report the higher performance we obtained. In Table 1a, we report the results under standard setting. Specifically, CiteULike dataset contains 3,396 cold-start items out of 16,980 total items, and MLIMDb dataset contains 229 cold-start items out of 1,146 total items. The experimental results show that our proposed model outperforms all other models, including the state-of-the-art model shared (+attention) model, in all evaluation metrics. In Table1b, we report the results under challenging setting. Specifically, CiteULike dataset contains 4,245 cold-start items out of 16,980 total items, and MLIMDb dataset contains 286 cold-start items out of 1,146 total

TABLE 1. Performance comparison using R@20, R@50, R@100, and R@150 on cold-start items in each dataset. The best performance is in bold font. The previously reported SOTA results have been further improved in our experimentation for stringent comparison.

(a) Standard setting: 4:1 split

Method	CiteULike				MLIMDb			
	R@20	R@50	R@100	R@150	R@20	R@50	R@100	R@150
Flexible Model	**32.9**	**53.7**	**67.5**	**75.0**	**19.8**	**37.8**	**55.2**	**76.2**
Shared Model [18] (+attention)	31.5	50.0	67.1	69.3	19.5	33.5	52.7	71.0
Shared Model [18] (-attention)	30.2	51.4	65.7	72.3	18.9	31.9	46.3	63.2
None-shared Model	28.6	47.2	61.7	69.0	18.3	30.5	45.8	62.3
SimPDO [49]	27.9	44.6	59.2	67.2	17.5	31.5	50.6	67.7

(b) Challenging setting: 3:1 split.

Method	CiteULike				MLIMDb			
	R@20	R@50	R@100	R@150	R@20	R@50	R@100	R@150
Flexible Model	**25.9**	46.6	**61.3**	**69.3**	**13.8**	**34.7**	**53.8**	**72.5**
Shared Model [18] (+attention)	23.9	**48.3**	60.6	67.3	12.8	30.2	45.9	67.4
Shared Model [18] (-attention)	23.2	41.7	54.6	61.6	12.5	29.5	45.1	64.5
Non-shared Model	20.8	36.2	48.6	55.6	11.4	27.9	42.5	63.9
SimPDO [49]	22.5	41.0	53.2	61.2	11.9	28.7	45.7	66.0

items. Although the overall performance decreases in general as we have fewer samples in training set, our proposed model still outperforms other models with less performance degradation. In MLIMDb dataset, previous SOTA model drops significantly, while our model holds well even in the challenging setting. This demonstrates the robustness of our model in various cold-start settings.

The Strengths of our Model Through Additional Experiment Results Analysis: We provide results to demonstrate the distinctions of our model compared to the SOTA model during the learning process. In Figure 3, we present Recall@100 results as graphs during the training of the Shared model (+attention) [18] and our model on the CiteULike dataset. This allows us to confirm that our model reaches peak performance faster than the Shared model. Table 2 demonstrates the robustness of our model. We conducted experiments on both the CiteULike dataset and the MLIMDb dataset, setting the learning rates at 0.01, 0.001, 0.0005, 0.0001, and 0.00001, respectively. The experimental results showed that our model outperformed the Shared model (+attention) at all learning rates. Consequently, we have fixed the learning rate at 0.001 in the CiteULike dataset, and 0.0001 in the MLIMDb dataset, which yielded the best results. Table 3 compares the performance based on changes in item feature dimensions. We applied tf-idf to the bag-of-words from the item feature and compared dimensions of 5696, 1303, 546, and 263. These dimensions correspond to using only words that appear more than once, twice, thrice, and four times, respectively. When conducting the experiments, we changed the dimensions of the MLPs with three layers of the tightly-coupled user encoder and item encoder to [500,300,100] for 546 item feature dimension, [200,150,100] for 263 item feature dimension. The 5696, 1303 item dimensions maintain three layers of MLPs with [1000, 500, 100].

Case Study: Movie Recommendation: Our model outperforms the previous SOTA model on MLIMDb dataset. Here, we conduct a case study for further analysis. In Figure 4 we compared the performance based on feature categories of movies. Specifically, we select a single category from the IMDb dataset, and use it as smaller features for movies. This way, we can study how each category contribute to the cold-start predictions. 'All' refers to using all categories as feature values as in our results in Table 1a, 1b. 'Directors', 'Writers', 'Actors', and 'Genres' refer to using each category as a standalone feature value. Each individual category dimension is as follows: director is 197, writer is 340, actor is 727, and genre is 23. When conducting the experiments, we changed the dimensions of the MLPs with three layers of the tightly-coupled user encoder and item encoder to [300, 200, 100] for writer and actor categories, [100, 80, 60] for director category, and [20, 20, 20] for genre category. In the Shared (+attention) models using tightly-coupled method where feature similarity is important, using directors as a single feature resulted in the best performance. This indicates that the directors have a significant influence on movie recommendation. The next best performance was achieved by using writer and then actor as standalone features in the shared model, indicating their significant influence on movie recommendations. Using each of the three categories as standalone features yielded higher performance compared to using all four categories together as features, with improvements of 5.8, 3.5, and 1.3 respectively. However, when using genre as a single feature, the performance is slightly lower than when using all categories as features. From these observations, we can infer that the reason for the lower performance when using all categories as features compared to using director, writer, and actor individually is due to the low similarity of genres. This can have an impact on recommending cold start items. Movie genres tend to be more divisive than other categories, meaning that users may interact more with movies of their preferred genres. If movies are recommended solely based on genre similarity, then movies from the genres with more interactions will be recommended more often, potentially leading to a problem where genres with fewer interactions are ignored and diverse

TABLE 2. Comparison of results based on Learning Rates.

Learning Rate	CiteULike		MLIMDb	
	Flexible Model	Shared Model (+attention) [18]	Flexible Model	Shared Model (+attention) [18]
0.01	65.7	64.8	52.1	44.1
0.001	67.5	67.1	52.7	44.8
0.0005	67.3	66.6	53.1	45.3
0.0001	65.0	64.8	53.8	45.9
0.00001	60.5	57.0	49.8	42.5

TABLE 3. Performance comparison based on item feature dimension variations in the MLIMDb dataset.

	Flexible Model	Shared Model (+attention) [18]
5696	48.8	46.3
1303	55.2	52.7
546	54.8	50.4
263	54.2	49.1

recommendations are not made. Table 4 also support this analysis. Table 4 shows the movies recommended through the tightly coupled method: shared (+attention) model and our Flexible Two-Tower Model from a selected user in MLIMDb dataset. Based on the user's past watched movies, the directors, writers, and actors who appear in 'Cosi', 'Reality Bites', and 'He Walked by Night' movies that the user has interacted with at least once or twice. The user has interacted with 43 Dramas, 38 Comedies, 19 Crime, 11 Action, and 11 Adventure films. In addition, the user has interacted with one Film-Noir and two Thriller movies. However, a similarity-based recommendation model only recommends movies that belong to the genre with which the user has interacted the most, which are the 'Cosi' movies in this case. On the other hand, our model utilizes a loosely-coupled method that reflects the user-item interaction relationship, which allows it to recommend movies from genres that may not have high similarity but are still relevant to the user's past interactions, such as the movies 'Reality Bites', and 'He Walked by Night'. Furthermore, in Figure 4, our model shows significantly better performance than the shared model. When using all four movie features, the model performs 0.6% better than when using only the directors as feature, 1.1% better than when using only the writers as feature, 3.% better than when using only the actors as feature, and 6.5% better than when using only the genres as feature. This demonstrates the flexibility and effectiveness of our model, which overcomes the limitations of similarity-based recommendations by incorporating user-item interactions and considers various movie features to provide more personalized recommendations.

B. GUMBEL-SOFTMAX AS GATING MECHANISM

Our model performs the choices by stochastically selecting between the tightlyvcoupled and loosely coupled user encoders. We compare two the two gating mechanism, Softmax and Gumbel-Softmax. In the Table 5 we report the result in R@100 for two datasets each under two settings. The experiments were conducted from CiteULike and MLIMDb into different subsets based on the cold-start ratio: standard and challenging. The experimental results reveal the effectiveness of hard gating using the Gumbel-Softmax, where Gumbel-Softmax always achieves higher performance than Softmax. We believe the regular Softmax achieves smoothed representations, and thus underperforms compared to the Gumbel-Softmax. Although there was a slight performance difference depending on the selection method, both methods outperformed the state-of-the-art models in Table 1. This confirms the strength of our flexible model, and demonstrates its practical value for cold-start recommendation systems.

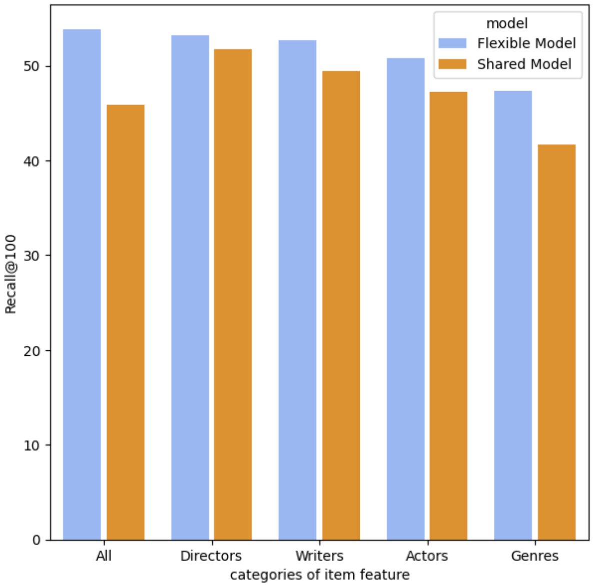


FIGURE 4. Performance comparison between models using item features on given category.

C. COMPARISON WITH STRONG BASELINE MODELS

As we want to validate how our proposed model performs compared to the state-of-the-art model [18], we use the same evaluation metric in [18] which is in R@100. In Table 6, the methods with * denote that the results are directly taken from [18].[2] Both shared model with and without

[2] The results of Shared model are reproduced by our experimentation with fine-tuning on the original code, which exhibit improved performance than the results reported in [18].

TABLE 4. Comparison example of cold-start movies recommendation between the Shared (+attention) model and the proposed model.

main features of interaction movies (10↑)	Interaction cold start item list	Recommendation of tightly-coupled method	Recommendation of Flexible Two-Tower Model
1. 'Drama' 2. 'Comedy' 3. 'Romance' 4. 'Crime' 5. 'Action' 6. 'Adventure'	• 'Reality bites' • 'Cosi' • 'He Walked by Night'	• 'Reality bites'	• 'Reality bites' • 'Cosi' • 'He Walked by Night'

Cold start movies	directors	writers	actors	genres
'Reality bites'	Ben Stiller	Helen Childress	Winona Ryder, Ethan Hawke, Janeane Garofalo, Steve Zahn	Comedy, Drama, Romance
'Cosi'	Mark Joffe	Louis Nowra	Ben Mendelsohn, Barry Otto, Toni Collette, Rachel Griffiths	Comedy, Drama, Music
'He Walked by Night'	Alfred L. Werker, Anthony Mann	John C. Higgins, Crane Wilbur, Harry Essex	Richard Basehart, Scott Brady, Roy Roberts, Whit Bissell	Crime, Film-Noir, Thriller

TABLE 5. Performance comparison between Softmax and Gumbel-Softmax in our proposed model. Gumbel-Softmax always performs better than Softmax.

(a) R@100 of item cold-start under standard setting

Gating Mechanism	CiteULike	MLIMDb
Gumbel-Softmax	**67.5**	**55.2**
Softmax	67.4	53.8

(b) R@100 of item cold-start under challenging setting

Gating Mechanism	CiteULike	MLIMDb
Gumbel-Softmax	**61.3**	**53.1**
Softmax	60.5	52.1

TABLE 6. Performance comparison of cold-start recommendation in R@100 using the CiteULike dataset. Results with * denote that the results are directly taken from [18]. The previously reported SOTA results have been further improved in our experimentation for stringent comparison.

Method	Test Recall (%)
Flexible Model	**67.5**
Shared Model [18] (+attention)	67.1
Shared Model [18] (-attention)	65.7
DN-WMF* (DropoutNet, retrained) [20]	65.2
DN-WMF* (DropoutNet) [20]	63.6
ACCM* [19]	63.1
DN-CDL* (DropoutNet) [20]	62.9
None-Shared Model	61.7
DeepMusic* [36]	60.1
SimPDO [49]	59.2
CTR* [37]	58.9
CDL* [38]	57.3

attention mechanism, only focuses on the side information of the items, and have been proven to be effective for new item recommendations. By comparing the Shared model with and without attention mechanism, we also observe that the attention mechanism is effective. The Shared [18] and SimPDO [49] models can be also be viewed as our model with loosely-coupled model never been activated. Non-Shared model has been also included to the baseline, which can be also viewed as our model with tightly-coupled model never been activated. From the table above, our results show an improvement of 0.4% over the best performing Shared model (+attention), and even greater improvements over other strong base models. This is meaningful as the results we achieve is not a smoothed results of the two, but based on selecting better representations at each interactions through our proposed scheme. It is also worth noting that the train-test split has been fixed across different models for fair comparison, and thus 0.4% improvement is not minimal. Overall, our proposed model achieves strong performance in cold-start item recommendation, as evidenced by its superior performance compared to other base models.

D. PERFORMANCE IN WARM-START SETTINGS

To make the RS practical, the model should have competitive performance not only for the cold-start, but also in warm-start settings. Here, we compare the performance of our model to the shared model with attention mechanism and the none-shared model, where we only compare the

TABLE 7. Comparison of average recall values of R@100 with warm-start items.

Method	Test Recall(CiteULike) (%)	Test Recall(MLIMDB) (%)
Flexible Model	**74.8**	**61.9**
Shared Model [18] (+attention)	73.5	58.0
None-Shared Model	61.8	52.3
SimPDO [49]	67.0	57.9

TABLE 8. The ratio of selection between tight and loose for each dataset. Results are from standard split.

	tightly-coupled	loosely-coupled
CiteULike	54.1	45.8
MLIMDb	50.1	49.9

performance in R@100. As shown in Table 7, our model achieves improvements of 1.3% and 3.9% over the shared model with attention, as well as improvements of 13.% and 9.6% over the non-shared model. Comparing to SimPDO, we achieve improvement of 6.2% and 4.% for the CiteULike and MLIMDb datasets, respectively. We show how our flexible model that stochastically selects the tightly-coupled or loosely-coupled achieves better performance than the models using each mode separately in every case: cold and warm.

E. SELECTION RATIO

Our study showed better performance than using a single tightly coupled or loosely coupled approach alone. Therefore, we provide important insights into the effects of flexibility. To verify the effect of flexibility, we conducted experiments to confirm whether our model uses both tightly-coupled and loosely-coupled approaches simply count the number of selected indicator vectors in Equation 8. Table 8 shows the average tower selection ratio according to the epoch. In the CiteULike dataset, the average selection probabilties of *tightly coupled* and *loosely coupled* methods are 54.1% and 45.8%, respectively. In the MLIMDb dataset, the average selection probabilities of *tightly coupled* and *loosely coupled* methods are 50.1% and 49.9%, respectively. Based on the results, it can be observed that both *tightly* and *loosely* coupled encoders are fully used in both data sets. We can observe an increased impact of *loosely coupled* method compared to the CiteULike dataset. Through the changes in influence of method, we can explain one of the reasons why the performance of our flexible method is significantly better than using either the tightly-coupled or loosely-coupled methods alone in the MLIMDb dataset compared to the CiteULike dataset in the Table 8. Furthermore, we can verify the effectiveness of the flexible method through the actual examples shown in Table 4, and in the Figure 4 showing the experimental results by categories of movie features. This result allowed us to compare the performance by category, demonstrating how our *Flexible Two-Tower* model can overcome the limitations of recommendation systems that only focus on similarity.

VI. DISCUSSION AND FUTURE WORK

Recommendation is important not only for accurately suggesting items that users want, but also for providing diverse recommendations. For cold-start item predictions, recommendations rely heavily on feature similarity as there is little to no interaction information available between the user and the item. However, relying solely on feature similarity for recommending cold start items can hinder diverse recommendations, as it may overlook the possibility that the user may prefer an item even if its similarity with their past interactions is low. Therefore, we propose a flexible model that can address such issues by incorporating the relationship between users and items. This is demonstrated by Table 8, which shows the flexibility of our approach, and various results that confirm its effectiveness. In particular, for the movie dataset, we observed that our model's flexibility in performance changes according to feature categories supplements similarity-based models. Despite the varying impact of different categories, we constructed the MLIMDb dataset with a similar structure to the original CiteULike dataset by setting each feature at the same level for the sake of fairness in the experiment. However, we expect to achieve better performance by further leveraging the information on the categories, such as having different weights on different categories. We leave this as our future research.

VII. CONCLUSION

One of the main challenges in RS is item cold-start problem, where the absence of previous interactions or ratings in new items makes it difficult to be predicted. To solve this problem, hybrid neural network models using side information of items as a feature have been favored in the literature. These models recommend items mainly based on the similarity of between the item features. However, focusing too much on item feature similarities can lead to missing capturing other signals for RS. We proposed a flexible model for better capturing the diverse aspects of users. Our proposed framework stochastically selects between the tightly coupled user encoder which focuses on capturing the item feature, and the loosely coupled user encoder which captures the patterns beyond item features. The effectiveness of this flexibility has been demonstrated through extensive experiments. The experimental results reveal that our proposed model achieve SOTA results for item cold-start recommendation. Moreover, in the warm-start settings, our proposed model achieves competitive performance outperforming the previous models. These two results reflect the practical application value of our proposed model.

REFERENCES

[1] H. Ko, S. Lee, Y. Park, and A. Choi, "A survey of recommendation systems: Recommendation models, techniques, and application fields," *Electronics*, vol. 11, no. 1, p. 141, Jan. 2022.

[2] X. Wang, K. Zhou, J.-R. Wen, and W. X. Zhao, "Towards unified conversational recommender systems via knowledge-enhanced prompt learning," in *Proc. 28th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, New York, NY, USA, Aug. 2022, pp. 1929–1937.

[3] P. Christmann, R. Saha Roy, and G. Weikum, "Explainable conversational question answering over heterogeneous sources via iterative graph neural networks," in *Proc. 46th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2023.

[4] D. Lin, J. Wang, and W. Li, "COLA: Improving conversational recommender systems by collaborative augmentation," in *Proc. AAAI Conf. Artif. Intell.*, Jun. 2023, vol. 37, no. 4, pp. 4462–4470.

[5] K. Zhou, H. Yu, W. X. Zhao, and J.-R. Wen, "Filter-enhanced MLP is all you need for sequential recommendation," in *Proc. ACM Web Conf.*, New York, NY, USA, Apr. 2022, pp. 2388–2399.

[6] H. Chen, Y. Lin, M. Pan, L. Wang, C.-C.-M. Yeh, X. Li, Y. Zheng, F. Wang, and H. Yang, "Denoising self-attentive sequential recommendation," in *Proc. 16th ACM Conf. Recommender Syst.*, New York, NY, USA, Sep. 2022, pp. 92–101.

[7] X. Li, A. Sun, M. Zhao, J. Yu, K. Zhu, D. Jin, M. Yu, and R. Yu, "Multi-intention oriented contrastive learning for sequential recommendation," in *Proc. 16th ACM Int. Conf. Web Search Data Mining*, New York, NY, USA, Feb. 2023, pp. 411–419.

[8] Y. Ma, X. Geng, and J. Wang, "A deep neural network with multiplex interactions for cold-start service recommendation," *IEEE Trans. Eng. Manag.*, vol. 68, no. 1, pp. 105–119, Feb. 2021.

[9] C.-Y. Tsai, Y.-F. Chiu, and Y.-J. Chen, "A two-stage neural network-based cold start item recommender," *Appl. Sci.*, vol. 11, no. 9, p. 4243, May 2021.

[10] S. A. Puthiya Parambath and S. Chawla, "Simple and effective neural-free soft-cluster embeddings for item cold-start recommendations," *Data Mining Knowl. Discovery*, vol. 34, no. 5, pp. 1560–1588, Sep. 2020.

[11] I. Fernández-Tobías, I. Cantador, P. Tomeo, V. W. Anelli, and T. Di Noia, "Addressing the user cold start with cross-domain collaborative filtering: Exploiting item metadata in matrix factorization," *User Model. User-Adapted Interact.*, vol. 29, no. 2, pp. 443–486, Apr. 2019.

[12] K. V. Rodpysh, S. J. Mirabedini, and T. Banirostam, "Employing singular value decomposition and similarity criteria for alleviating cold start and sparse data in context-aware recommender systems," *Electron. Commerce Res.*, vol. 23, no. 2, pp. 681–707, Jun. 2023.

[13] J. Jeevamol and V. G. Renumol, "An ontology-based hybrid e-learning content recommender system for alleviating the cold-start problem," *Educ. Inf. Technol.*, vol. 26, no. 4, pp. 4993–5022, Jul. 2021.

[14] G. Salha-Galvan, R. Hennequin, B. Chapus, V.-A. Tran, and M. Vazirgiannis, "Cold start similar artists ranking with gravity-inspired graph autoencoders," in *Proc. 15th ACM Conf. Recommender Syst.*, New York, NY, USA, Sep. 2021, pp. 443–452.

[15] T. Qi, F. Wu, C. Wu, and Y. Huang, "PP-Rec: News recommendation with personalized user interest and time-aware news popularity," in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics 11th Int. Joint Conf. Natural Lang. Process.*, 2021, pp. 5457–5467.

[16] T. Qian, Y. Liang, Q. Li, and H. Xiong, "Attribute graph neural networks for strict cold start recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 8, pp. 3597–3610, Aug. 2022.

[17] R. Nahta, Y. K. Meena, D. Gopalani, and G. S. Chauhan, "Embedding metadata using deep collaborative filtering to address the cold start problem for the rating prediction task," *Multimedia Tools Appl.*, vol. 80, no. 12, pp. 18553–18581, May 2021.

[18] R. Raziperchikolaei, G. Liang, and Y.-J. Chung, "Shared neural item representations for completely cold start problem," in *Proc. 15th ACM Conf. Recommender Syst.*, H. J. C. Pampín, M. A. Larson, M. C. Willemsen, J. A. Konstan, J. J. McAuley, J. Garcia-Gathright, B. Huurnink, and E. Oldridge, Eds. Amsterdam, The Netherlands, Sep. 2021, pp. 422–431.

[19] S. Shi, M. Zhang, Y. Liu, and S. Ma, "Attention-based adaptive model to unify warm and cold starts recommendation," in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manage.*, New York, NY, USA, Oct. 2018, pp. 127–136.

[20] M. Volkovs, G. Yu, and T. Poutanen, "DropoutNet: Addressing cold start in recommender systems," in *Proc. Adv. Neural Inf. Process. Syst.*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Red Hook, NY, USA: Curran Associates, Inc., 2017, pp. 1–12.

[21] Y. Liu, S. Wang, M. S. Khan, and J. He, "A novel deep hybrid recommender system based on auto-encoder with neural collaborative filtering," *Big Data Mining Anal.*, vol. 1, no. 3, pp. 211–221, Sep. 2018.

[22] A. Guzman-Rivera, D. Batra, and P. Kohli, "Multiple choice learning: Learning to produce multiple structured outputs," in *Proc. 25th Int. Conf. Neural Inf. Process. Syst.* Red Hook, NY, USA: Curran Associates, 2012, pp. 1799–1807.

[23] J. K. Arthur, C. Zhou, E. A. Mantey, J. Osei-Kwakye, and Y. Chen, "A discriminative-based geometric deep learning model for cross domain recommender systems," *Appl. Sci.*, vol. 12, no. 10, p. 5202, May 2022.

[24] J. K. Arthur, C. Zhou, J. Osei-Kwakye, E. A. Mantey, and Y. Chen, "A heterogeneous couplings and persuasive user/item information model for next basket recommendation," *Eng. Appl. Artif. Intell.*, vol. 114, Sep. 2022, Art. no. 105132.

[25] D. K. Panda and S. Ray, "Approaches and algorithms to mitigate cold start problems in recommender systems: A systematic literature review," *J. Intell. Inf. Syst.*, vol. 59, no. 2, pp. 341–366, Oct. 2022.

[26] G. Xin, J. Qin, and J. Zheng, "A hybrid recommendation algorithm with co-embedded item attributes and ratings," in *Proc. 4th Int. Conf. Appl. Mach. Learn. (ICAML)*, Jul. 2022, pp. 1–7.

[27] S. Li, J. Kawale, and Y. Fu, "Deep collaborative filtering via marginalized denoising auto-encoder," in *Proc. 24th ACM Int. Conf. Inf. Knowl. Manage.*, New York, NY, USA, Oct. 2015, pp. 811–820.

[28] X. Dong, L. Yu, Z. Wu, Y. Sun, L. Yuan, and F. Zhang, "A hybrid collaborative filtering model with deep structure for recommender systems," in *Proc. AAAI Conf. Artif. Intell.*, Feb. 2017, vol. 31, no. 1, pp. 1–13.

[29] M. Kaminskas and F. Ricci, "Contextual music information retrieval and recommendation: State of the art and challenges," *Comput. Sci. Rev.*, vol. 6, nos. 2–3, pp. 89–119, May 2012.

[30] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with Gumbel-softmax," in *Proc. 5th Int. Conf. Learn. Represent.*, Toulon, France, Apr. 2017.

[31] C. Shen, G.-J. Qi, R. Jiang, Z. Jin, H. Yong, Y. Chen, and X.-S. Hua, "Sharp attention network via adaptive sampling for person re-identification," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 10, pp. 3016–3027, Oct. 2019.

[32] Y. Wang and J. M. Solomon, "PRNet: Self-supervised learning for partial-to-partial registration," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019.

[33] S. Yan, J. S. Smith, W. Lu, and B. Zhang, "Hierarchical multi-scale attention networks for action recognition," *Signal Process., Image Commun.*, vol. 61, pp. 73–84, Feb. 2018.

[34] P. Guo, C.-Y. Lee, and D. Ulbricht, "Learning to branch for multi-task learning," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 3854–3863.

[35] C. Wang and D. M. Blei, "Collaborative topic modeling for recommending scientific articles," in *Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, New York, NY, USA, Aug. 2011, pp. 448–456.

[36] A. van den Oord, S. Dieleman, and B. Schrauwen, "Deep content-based music recommendation," in *Proc. 26th Int. Conf. Neural Inf. Process. Syst. (NIPS)*, vol. 2. Lake Tahoe, NV, USA, 2013, pp. 2643–2651.

[37] H. Wang, B. Chen, and W.-J. Li, "Collaborative topic regression with social regularization for tag recommendation," in *Proc. 23rd Int. Joint Conf. Artif. Intell.*, 2013, pp. 2719–2725.

[38] H. Wang, N. Wang, and D.-Y. Yeung, "Collaborative deep learning for recommender systems," in *Proc. 21st ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, New York, NY, USA, Aug. 2015, pp. 1235–1244.

[39] Y. Wei, X. Wang, Q. Li, L. Nie, Y. Li, X. Li, and T.-S. Chua, "Contrastive learning for cold-start recommendation," in *Proc. 29th ACM Int. Conf. Multimedia*, New York, NY, USA, Oct. 2021, pp. 5382–5390.

[40] H. Lee, J. Im, S. Jang, H. Cho, and S. Chung, "MeLU: Meta-learned user preference estimator for cold-start recommendation," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, New York, NY, USA, Jul. 2019, pp. 1073–1082.

[41] R. Yu, Y. Gong, X. He, Y. Zhu, Q. Liu, W. Ou, and B. An, "Personalized adaptive meta learning for cold-start user preference prediction," in *Proc. AAAI Conf. Artif. Intell.*, May 2021, vol. 35, no. 12, pp. 10772–10780.

[42] N. Heidari, P. Moradi, and A. Koochari, "An attention-based deep learning method for solving the cold-start and sparsity issues of recommender systems," *Knowl.-Based Syst.*, vol. 256, Nov. 2022, Art. no. 109835.

[43] M. Mirhasani and R. Ravanmehr, "Alleviation of cold start in movie recommendation systems using sentiment analysis of multi-modal social networks," *J. Adv. Comput. Eng. Technol.*, vol. 6, no. 4, pp. 251–264, 2020.

[44] K. Vahidy Rodpysh, S. J. Mirabedini, and T. Banirostam, "Model-driven approach running route two-level SVD with context information and feature entities in recommender system," *Comput. Standards Interface*, vol. 82, Aug. 2022, Art. no. 103627.

[45] L. Gan, D. Nurbakova, L. Laporte, and S. Calabretto, "Enhancing recommendation diversity using determinantal point processes on knowledge graphs," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, New York, NY, USA, Jul. 2020, pp. 2001–2004.

[46] P. M. T. Do and T. T. S. Nguyen, "Semantic-enhanced neural collaborative filtering models in recommender systems," *Knowl.-Based Syst.*, vol. 257, Dec. 2022, Art. no. 109934.

[47] R. Shimizu, K. Asako, H. Ojima, S. Morinaga, M. Hamada, and T. Kuroda, "Balanced mini-batch training for imbalanced image data classification with neural network," in *Proc. 1st Int. Conf. Artif. Intell. Industries (AI4I)*, Sep. 2018, pp. 27–30.

[48] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie, "Class-balanced loss based on effective number of samples," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9260–9269.

[49] R. Raziperchikolaei and Y. J. Chung, "One-class recommendation systems with the Hinge pairwise distance loss and orthogonal representations," 2022, *arXiv:2208.14594*.

[50] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proc. 26th Int. Conf. World Wide Web*, 2017, pp. 173–182.

WON-MIN LEE received the B.S. degree in computer software engineering from Kunsan University, Kunsan, South Korea, in 2021. She is currently pursuing the M.S. degree in artificial intelligence with Chung-Ang University, Seoul, South Korea. Her research interests include recommendation systems and natural language processing.

YOON-SIK CHO received the B.S. degree in electrical engineering from Seoul National University, South Korea, in 2003, and the Ph.D. degree in electrical engineering from the University of Southern California, USA, in 2014. He was an Academic Mentor of the RIPS Program with the Institute for Pure and Applied Mathematics, University of California at Los Angeles, and a Postdoctoral Scholar with the Information Sciences Institute, University of Southern California. He is currently an Assistant Professor with the Department of AI, Chung-Ang University, South Korea. His research interests include large-scale data science, social network analysis, and cloud computing.