Multi-Tower Multi-Interest Recommendation with User Representation Repel

Tianyu Xiong
Huawei Research
Beijing, China
tianyuxiong@gmail.com

Xiaohan Yu
Huawei Research
Beijing, China
yuxh97@gmail.com

ABSTRACT

In the era of information overload, the value of recommender systems has been profoundly recognized in academia and industry alike. Multi-interest sequential recommendation, in particular, is a subfield that has been receiving increasing attention in recent years. By generating multiple user representations, multi-interest learning models demonstrate superior expressiveness than single user representation models, both theoretically and empirically. Despite major advancements in the field, three major issues continue to plague the performance and adoptability of multi-interest learning methods, the difference between training and deployment objectives, the inability to access item information, and the difficulty of industrial adoption due to its single-tower architecture. We address these challenges by proposing a novel multi-tower multi-interest framework with user representation repel. Experimental results across multiple large-scale industrial datasets proved the effectiveness and generalizability of our proposed framework.

CCS CONCEPTS

• **Information systems → Probabilistic retrieval models**.

KEYWORDS

Recommender System, Candidate Matching, Multi-Interest Learning, Sequential Recommendation, Metric Learning

ACM Reference Format:
Tianyu Xiong and Xiaohan Yu. 2023. Multi-Tower Multi-Interest Recommendation with User Representation Repel. In *Proceedings of The 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (ACM/SIGIR).* ACM, New York, NY, USA, 9 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Sequential recommendation systems play a vital role in addressing the issue of information overload for users, particularly in domains like e-commerce, social media, and music streaming. They are instrumental in optimizing key business metrics such as click-through rates (CTR). These systems organize items based on when users interact with them, focusing on mining sequential patterns to predict the next item of interest to users. Many existing methods combine user preferences and item characteristics to make accurate predictions. As a result, research in sequential recommendation primarily revolves around enhancing the quality of how users and items are represented.

Due to the practical significance of sequential recommendation systems, various approaches have been proposed, yielding promising results. For instance, GRU4Rec [1] was the pioneering work that applied Recurrent Neural Networks (RNN) to model sequential information for recommendations. Kang and McAuley [2] introduced an attention-based method to capture complex, dynamic patterns in sequences. More recently, some approaches, like PinSage [3], have harnessed Graph Neural Network (GNN) techniques to derive user and item representations for downstream tasks. Nevertheless, it's worth noting that most prior studies have focused on creating a single, comprehensive representation of a user's behavior sequence, which may not effectively capture a user's diverse interests. Few studies in the literature have attempted to address the challenge of representing a user's multiple interests adequately within a single vector.

In recent times, there has been a notable rise in the adoption of multi-interest learning-based approaches [4, 5], demonstrating significant potential in enhancing matching performance. These methods explicitly address the challenge of representing users' varied interests by deriving diverse interest representations from their behavioral sequences, effectively overcoming the limitations associated with a single, generic user embedding. One such method, MIND [4], achieves this by initially capturing a user's multiple interests through dynamic routing, employing a Capsule Network [6]. Subsequently, ComiRec [5] takes the concept of diversity into account and extends the approach by utilizing multi-head attention mechanisms to encode the diverse interests of users. Recent research efforts [7, 8] have gone even further, incorporating considerations of periodicity, interactivity, and user profiles into the modeling process. Additionally, REMI[9] has introduced an interest-aware hard negative mining strategy and used routing regularization to address the problem of routing collapse. These advancements reflect the growing recognition of the importance of accommodating users' multiple interests and improving the overall performance of matching algorithms in various applications.

Despite the various model architectures and information explored in the realm of multi-interest learning, the current paradigm for multi-interest learning frames the challenge of candidate matching as an extreme multiclass classification problem. In this prevailing approach, the user's behavioral sequence is initially transformed into a sequence of item embeddings, which are subsequently translated into multiple interest representations.

ACM/SIGIR, July 14-18, 2024, Washington, DC, USA

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

When these multi-interest representations are deployed, they individually retrieve sets of items using the k-nearest neighbors (kNN) algorithm. However, during the training phase to create these multi-interest representations, a label-aware attention mechanism or argmax selection process is employed to identify the representation that is closest to the positive target. Once this representation is selected, it becomes the active user representation and is trained in a manner similar to general candidate matching. This typically involves applying a softmax function, either uniformly or according to a log-uniform distribution [10, 11], to efficiently train the model on a large dataset. Nevertheless, it's important to acknowledge that there are two significant drawbacks associated with this approach. Firstly, the practice of selecting the most suitable representation using the target item during training creates a disconnect between the objectives of the training and deployment phases, leading to overly simplistic training. Secondly, framing candidate matching as a multiclass classification problem reduces each candidate item to a binary label (0 or 1), which fundamentally limits the utilization of valuable item-side information and semantic knowledge.

To tackle these issues, we introduce a novel approach called Multi-Tower Multi-Interest Learning (MTMI) as an alternative paradigm for multi-interest learning. Our approach not only addresses the aforementioned issues but also facilitates the seamless adaptation of multi-interest learning to two-tower candidate matching models, which are widely employed in the industry today.

MTMI draws inspiration from the Deep Structured Semantic Models (DSSM)[12] approach, where user and item information are independently processed to produce user embeddings and item embeddings. Unlike the common practice of employing a single-user tower and a single-item tower (the two-tower design), MTMI allows for the incorporation of multiple user towers, each generating a distinct user representation. We use an Inverse Distance Weighted Loss to assess the weighted distance between the generated user representations and the target item representation. Fig. 1 illustrates a motivating example of our MTMI framework.

We conducted extensive experiments, comparing MTMI-based models to MIND-based models under basic settings and applying state-of-the-art techniques. These experiments utilized three real-world, large-scale public recommendation datasets. The results demonstrate that MTMI-based methods significantly outperform MIND-based methods in basic settings. While they may not surpass the state-of-the-art in multi-interest learning, they pave the way for future advancements and improvements in this field.

In summary, our contributions can be distilled into three key aspects:

- **Problem Reevaluation**: We reevaluated the prevailing multi-interest learning paradigm, identifying three significant issues regarding the disparity between training and deployment objectives, the constraints on accessing item information, and the difficulty for industrial adoption.
- **MTMI Paradigm Introduction**: We introduced a groundbreaking Multi-Tower Multi-Interest learning paradigm (MTMI) designed to address these identified problems effectively. MTMI also enhances the applicability of Multi-Interest Learning, particularly for two-tower candidate generation systems commonly used in the industry.
- **Empirical Validation**: We conducted comprehensive experiments, revealing that MTMI yields substantial improvements over existing multi-interest learning methods, under the most basic settings. These findings underscore the practical value and promise of MTMI in the field of recommendation systems.

2 RELATED WORK

2.1 Candidate Matching

In the context of large-scale industrial recommender systems, candidate matching holds significant importance as it effectively narrows down a selection from a vast pool of items for subsequent refined ranking processes [8, 13, 14]. Given the imperative for efficiency, candidate matching models typically employ lightweight architectures and often do not incorporate candidate awareness during user modeling. In earlier stages, solutions based on Collaborative Filtering (CF) [15, 16] introduced learnable mechanisms for matching users with candidates, while Neural network-based Collaborative Filtering (NCF) [17] enhanced traditional CF with multi-layer perceptrons. Subsequently, the adoption of two-tower Deep Neural Network (DNN) structures [10, 12] gained popularity due to their computational efficiency, avoiding early interactions between user and candidate modeling. Furthermore, there have been investigations into tree-based and graph-based structures for deep candidate matching [18–20]. For instance, PDNP [18] devised a retrieval architecture based on a 2-hop graph, facilitating online retrieval with minimal latency and computational costs. However, these approaches typically represent user preferences as a single vector, potentially limiting their ability to capture the multi-interest nature of users.

2.2 Sequential Recommendation

The issue of sequential recommendation stands as a pivotal challenge within the realm of recommender systems, and it has garnered substantial attention in recent research endeavors. Several noteworthy contributions have focused on addressing this challenge. For instance, FPMC [21] encapsulates both a conventional Markov chain and the standard matrix factorization model to handle sequential basket data. HRM [22] extends the FPMC framework by implementing a two-layer structure that facilitates the construction of a hybrid representation encompassing users and items based on the most recent transaction. In a pioneering fashion, GRU4Rec [23] introduces an RNN-based approach to comprehensively model entire user sessions, thereby enhancing the precision of recommendations. DREAM [24], founded on Recurrent Neural Networks (RNN), leverages dynamic user representations to unveil evolving user interests. Fossil [25] seamlessly integrates similarity-based techniques with Markov Chains to enable personalized sequential predictions, particularly suited for sparse and long-tailed datasets. TransRec [26] adopts an approach that embeds items into a vector space, representing users as vectors that operate on item sequences, thereby enabling large-scale sequential prediction. RUM [27], on the other hand, combines a memory-augmented neural network with insights from collaborative filtering to facilitate recommendation. SASRec [28] employs a self-attention-based sequential model, adept at capturing long-term semantics, and employs an attention

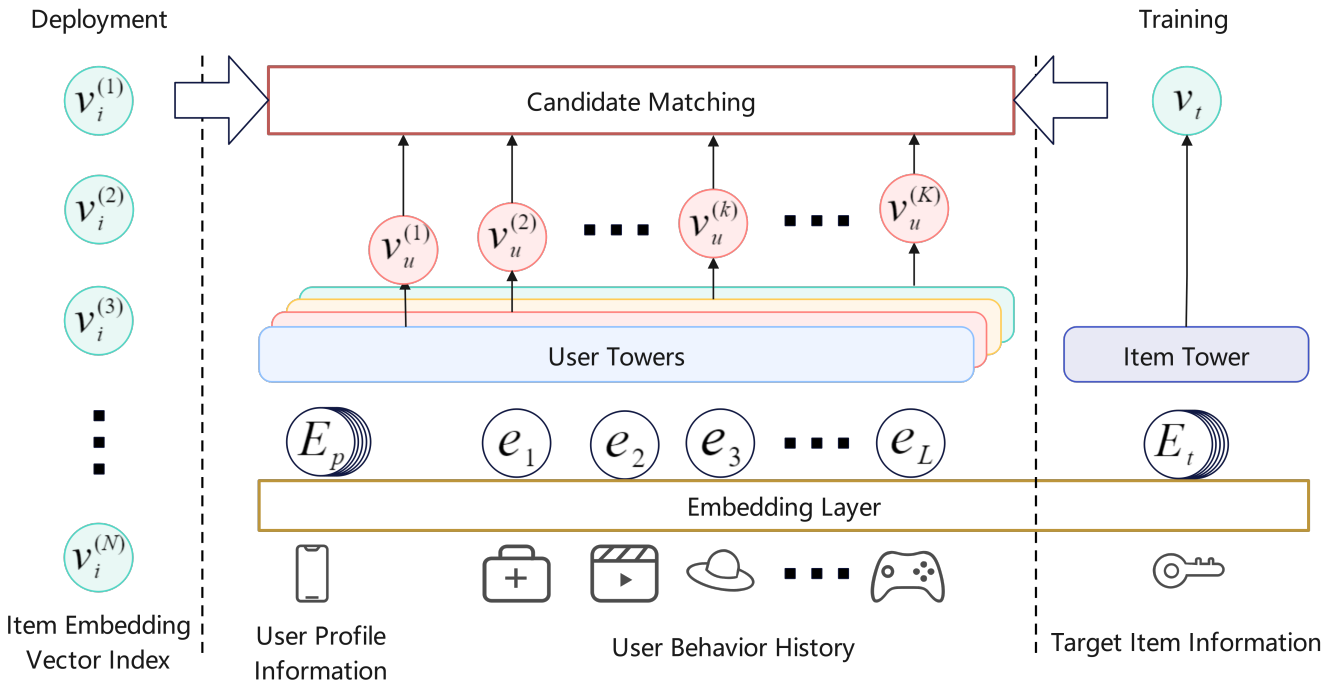

Figure 1: The MTMI candidate matching system using a multi-tower architecture with embedding layers. Item features(right), are use to compute a item representation(Right). User behaviors and user profiles are mapped to embedding vectors, which are then used to compute multiple user interst representations. User interest representations and item representation are used to compute similarity scores and match candidates to users.

mechanism to generate predictions based on relatively limited user actions. In contrast, DIN [29] introduces a local activation unit designed to adaptively learn user interest representations based on past behaviors pertaining to specific advertisements. Finally, SDM [14] encodes behavior sequences through a multi-head self-attention module, allowing for the capture of multiple types of user interests, while a long-short-term gated fusion module is employed to incorporate long-term preferences. These prior works collectively contribute to the body of research addressing the challenges posed by sequential recommendation in recommender systems.

2.3 Multi-interest Learning

Recent studies have highlighted the inadequacy of representing users' interests as a single vector, leading to increased interest in multi-interest learning for both the matching and ranking stages of recommender systems. In the matching stage, MIND [4] introduced a dynamic routing mechanism to aggregate users' historical behaviors into multiple interest capsules, followed by ComiRec [5], which explored multi-head attention-based multi-interest routing for a more diverse user interest representation. PIMIRec[7] and UMI [8] extended these concepts by incorporating time information, interactivity, and user profiles, while Re4 [30] considered backward flow regularization. Furthermore, REMI[9] has presented a strategy for mining hard negatives that takes into account user interests and has employed routing regularization as a means to mitigate the issue of routing collapse.

2.4 Deep Metric Learning

The main objective of Metric Learning is to measure the similarity of samples using an optimally learned distance metric[31, 32]. In recent years, Deep Metric Learning, which empowers metric learning with deep architectures has been gaining traction due to its superior expressibility of nonlinearities. It has been widely adopted in the state-of-the-art (SOTA) face recognition and information retrieval models[33]. [34] first proposed a contrastive loss for face recognition, which separates dissimilar samples by a given margin. The triplet loss, being one of the foundations of this article, is first proposed in [35], which minimizes the distance between the anchor and the positive sample simultaneously maximizing the distance between the anchor and negative samples. [36] proposed quadruplet loss to minimize intraclass variation, and [37, 38] improves sample effectiveness by making full use of samples in a batch, The learning objective was improved by taking the overall structure of the dataset into account as in [39, 40]. Despite these innovations. The triplet loss and contrastive loss remain to be relevant due to their simplicity and computational efficiency.

3 METHOD

3.1 Problem Formulation

The Candidate Matching phase of the industrial recommendation system is designed to select a subset of items from a vast item pool (denoted as $\mathcal{I}$) containing billions of items for each user u

Table 1: Notation

Notation	Description
u	an user
i	an item
e	the embedding of a feature
t	a target item
v_u	a user representation
v_i	a item representation
v_t	a target item representation
E_p	the matrix of a user's profile embeddings
E_t	the matrix of a target item embeddings
E_u	the matrix of a user's behavior embeddings
$\mathcal{U}$	the set of users
$\mathcal{I}$	the set of items
d	the dimension of an embedding
K	the number of user representations
N	the number of candidate items

belonging to the set of users $\mathcal{U}$. This subset should consist of only a few thousand items, and each item should be relevant to the user's interests. To accomplish this objective, historical data generated by the RS is gathered to construct a matching model. More specifically, each instance can be represented as a tuple (E_u, E_p, E_t), where E_u represents the embedding matrix of items that user u has interacted with (referred to as user behavior), E_p is the embedding matrix that encompasses the user's basic profiles (e.g., gender and age), and E_t is the embedding matrix that encompasses the features of the target item (e.g., item ID, category ID, brand ID, seller ID, title, etc.).

The main objective of single-tower Multi-Interest Learning algorithms is to learn the mapping f_{user} that maps raw feature embeddings into user representation vectors.

$$V_u = f_{user}(E_p, E_u) \tag{1}$$

which minimize

$$\mathcal{L}_{\mathcal{SSM}}\left(\Upsilon(v_t, V_u, V_u)\right)$$

Given the user representation matrix V_u

$$V_u = \left(v_u^{(1)}, v_u^{(2)}, v_u^{(3)}, ..., v_u^{(K)}\right) \in R^{d \times K}$$

and the label aware attention Υ

$$\Upsilon(v_t, V_u, V_u) = V_u Softmax(Pow(V_u^T v_t, p))$$

For a detailed description of the notation used in our formulation, please refer to Table 1.

3.2 Embedding Layer & Pooling Layer

The embedding layer in the Multi-Tower Multi-Interest (MTMI) framework plays a crucial role in capturing the essence of user behavior and item features. This layer acts as a bridge between raw user interactions and item attributes, transforming them into dense, continuous vectors that can be easily processed by subsequent layers of the model. The embedding layer takes user behavior sequences, user profiles, and item features as input and represents them as numerical embeddings in a continuous vector space.

3.3 Representation/Interest Extraction

In MTMI-basic, a simplified approach is adopted for the representation extraction module, referred to as "the towers." Instead of employing complex architectural components, the representation extraction module in this variant relies on straightforward attention fusion modules as illustrated in 3. [41, 42]

$$w = Softmax(proj_L(e_1||...||e_L)) \tag{2}$$

$$f = \sum_{i=1}^{L} w_i e_i \tag{3}$$

3.4 Multi-tower Architecture

The Multi-Tower Multi-Interest (MTMI) framework incorporates a unique multi-tower architecture designed to enhance the modeling of user interests and item characteristics. This architecture comprises multiple user towers and an item tower, all of which share a common embedding layer. This is demonstrated in Fig. 1. During the training phase, the multiple user towers independently generate user representations, while the item tower simultaneously produces a target item representation. These user representations are designed to exhibit repulsion from one another, reflecting the diversity of user interests within the system. At the same time, they are drawn towards the target item representation.

In the deployment phase, the user representations produced by the user tower individually gather a collection of nearby item representations, along with the associated distances, using the K-Nearest Neighbor algorithm. Subsequently, these sets of items are consolidated and arranged based on the estimated distances, resulting in a ranked order. An illustrative graph of the training phase and deployment phase can be found in Fig. 2.

3.5 Invert Distance Weighting Loss(IDWLoss)

Given cosine similarity $\mathcal{S}_c(A, B)$, cosine distance $\mathcal{D}_c(A, B)$ and inverted distance $\mathcal{ID}_c(A, B)$

$$\mathcal{S}_c(A, B) = \frac{A \cdot B}{||A||||B||} \tag{4}$$

$$\mathcal{D}_c(A, B) = 1 - \mathcal{S}_c(A, B) \tag{5}$$

$$\mathcal{ID}_c(A, B) = \frac{1}{\mathcal{D}_c(A, B)^{\alpha}} \tag{6}$$

Our novel IDW loss $\mathcal{L}_{\mathcal{ID}\mathcal{W}}$ can be formulated as

$$\begin{aligned}\mathcal{L}_{\mathcal{ID}\mathcal{W}}(V_u, v_+, V_-) = \\ \sum_{j}^{K} \mathcal{ID}_c\left(v_u^{(j)}, v_t\right) \mathcal{L}_{triplet}\left(v_u^{(j)}, v_{i+}, v_{i-}^{(j)}\right) \\ + \beta \sum_{j}^{K} \mathcal{L}_{triplet}\left(v_u^{(j)}, v_{i+}, v_{u-}^{(j)}\right)\end{aligned} \tag{7}$$

where $\mathcal{L}_{triplet}$ is a triplet loss commonly used in metric learning. [35, 43]

$$\mathcal{L}_{triplet}(v_u, v_+, v_-) = \max\left(0, \gamma - \mathcal{S}_c\left(v_u, v_+\right) + \mathcal{S}_c\left(v_u, v_-\right)\right) \tag{8}$$

v_{i+} is the representation of the positive item.

$$v_{i+} = v_t \tag{9}$$

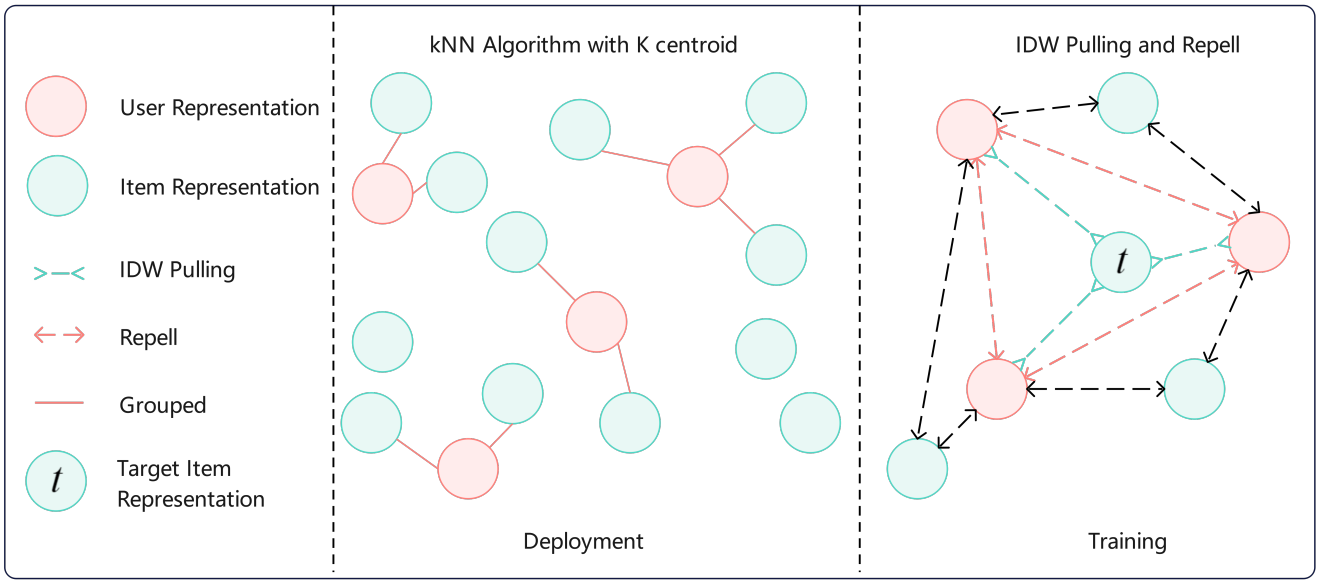

Figure 2: A visual representation of the Candidate Matching process, consisting of approximate kNN algorithm with K centroid for serving(Left), and IDW pulling and User Presentation Repel(Right)

Table 2: Statistics of Datasets

Dataset	Users	Goods	Interactions	Cutoff
MovieLens 100k	943	1,349	99,287	20
Retail Rocket	33,708	81,635	356,840	20
Amazon Books	603,668	367,982	8,898,041	20
Gowalla	976,779	1,708,530	85,384,110	40

v_{u-} represents user representations other than the currently selected one.

$$v_{u-}^{(j)} = \left[v_u^{(i)}; i \in \{1, ..., K\}, i! = j\right] \tag{10}$$

Negative items $i-$ are sampled from a multinomial distribution with 120 samples and N classes.

4 EXPERIMENTS

We perform extensive experiments on three real-world recommendation datasets in order to address the following research inquiries:

- **RQ1:** Can MTMI enhance common two-tower Candidate Matching models?
- **RQ2:** Can MTMI beat single tower MIL algorithms under the most basic settings?
- **RQ3:** What is the effect of IDWLoss and User Representation Repel?
- **RQ4:** What is the Optimal Hyperparameter setting for MTMI?

4.1 Experimental Settings

4.1.1 Dataset. We have opted for the utilization of three extensive public datasets to conduct an assessment of MTMI's efficacy:

- **Amazon**[44]: This dataset encompasses a diverse array of product views derived from the widely recognized Amazon platform. For our evaluation, we specifically select the largest subset focusing on books, characterized by various book types.
- **Gowalla** [45]: This dataset collects check-in data from Gowalla, a location-based social networking website.
- **RetailRocket**[46]: This dataset mirrors real-world e-commerce interactions and spans a four-month duration, capturing various user behaviors. For the purpose of our study, we exclusively consider the view events within this dataset.
- **MovieLens**[47, 48]: This dataset was acquired from the MovieLens website over a seven-month duration. Subsequently, a data cleaning process was undertaken, wherein users with fewer than 20 ratings or those lacking comprehensive demographic information were excluded from the dataset. We use the 100k version of this dataset for the hyperparameter scan.

To ensure consistency and comparability, we have implemented dataset preprocessing methodologies in alignment with a prior research study [4, 5, 9]. This process involves the removal of items and users with occurrences falling below a predefined threshold of 5, maximum user behavior sequence length set to 20. Moreover, all user interactions within these datasets are treated as implicit feedback. For a detailed summary of the statistical attributes of the three datasets post-preprocessing, we refer to Table 2.

4.1.2 Training and Evaluation Setup. In adherence to the methodologies established in prior studies [4, 5, 9], we perform the partitioning of our dataset into training, validation, and test sets, maintaining a ratio of 8:1:1 concerning distinct users. In the training

phase, models are trained to utilize the complete user behavior sequences from the training set. During the evaluation process, we employ the initial 80% of the user behavior sequence to infer user embeddings, subsequently calculating the designated metrics using the remaining 20% of items within the sequence. For further elaboration, additional comprehensive information can be obtained from references [4, 5, 9]. We employ widely recognized evaluation metrics, specifically Recall, Hit Rate, and NDCG (Normalized Discounted Cumulative Gain), to assess the effectiveness of our proposed solution. These metrics are computed based on the top 20/50 matched candidates.

4.1.3 Baseline Models. To demonstrate the effectiveness of our MTMI framework, we compare it with classic single-tower MIL models such as MIND and ComiRec, some of the general sequential recommendation algorithms, as well as the SOTA single-tower MIL model REMI.

- **Most Popular** An algorithm recommending the most popular item to its users.
- **YouTube DNN** An influential industrial Deep Neural Network (DNN) model that aggregates behavior. embeddings, subsequently employing Multi-Layer Perceptron (MLP) layers to derive user representation.
- **GRU4Rec** The first sequential recommendation model based on Recurrent Neural Networks (RNN) that captures sequential patterns.
- **Que2Search** An influential industrial DNN model that aggregates behavior embeddings, subsequently employing Simple Attention Fusion layers to derive user representations.
- **MIND** Pioneering the realm of multi-interest learning frameworks, it leverages capsule networks to capture diverse user interests.
- **ComiRec** An advanced multi-interest framework that permits control over diversity and introduces a multi-head attention mechanism for modeling various user interests. The multi-head attention version, "ComiRec-SA", became the standard backbone for Multi-Interest Learning frameworks for its simplicity and stability.
- **REMI** The SOTA single-tower multi-interest learning framework that effectively tackled issues related to the challenge of "easy negative" instances and the problem of "routing collapse." This successful intervention has led to significant enhancement in performance.

4.1.4 Implementation Details. Our model was constructed using Torch 2.0.1, and for the purpose of k-nearest neighbors (kNN) search, Faiss 1.7.3 was employed. To visualize the outcomes, Matplotlib and Wandb were utilized. The optimization of hyperparameters involved an exhaustive grid search encompassing alpha, beta, and the number of towers or interests (referred to as K). This meticulous exploration led to the identification of the most effective parameter set $\{\alpha = 5.5, \beta = 6, K = 8\}$, which subsequently became the default configuration for our experiments unless otherwise specified. Similarly, the embedding dimension was firmly set at 64, while the batch size was established at 128 to ensure efficient model training. The learning rate was defined as 1×10^{-3} to strike a balance between convergence speed and model accuracy. For addressing negative

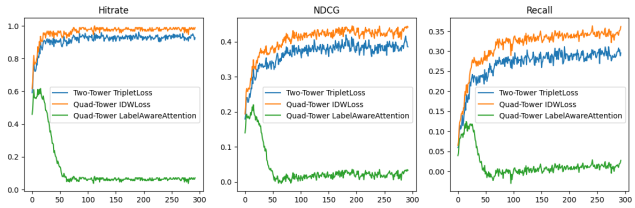

Figure 3: Study of the effect of IDWLoss. We use ML-100k dataset for these comparisons.

training requirements and maintaining consistency, the batch-wise shared negative training sample size was set at 128 * 10, and the number of negative samples was consistently maintained at 120 for all applicable experiments. These implementation details are pivotal for the reproducibility and robustness of our experimental procedures.

4.2 Enhancement to two tower models (RQ1)

We employed the MTMI scheme on three popular two-tower vector retrieval models, **YouTube DNN**, **GRU4Rec**, **Que2Search** to test if MTMI can effectively and consistently enhance existing two-tower models.

As illustrated in Table 3, the MTMI-enhanced models significantly outperform their vanilla counterparts.

4.3 Comparison with Basic MIL models (RQ2)

We also compared the performance of the MTMI-GRU model with basic MIL models. As illustrated in the able, the MTMI models also outperform basic MIL models.

4.4 Comparison with SOTA MIL models (RQ3)

Having comprehensively assessed MTMI in contrast to conventional single-tower models, we have additionally conducted a comparative analysis with the state-of-the-art (SOTA) single-tower multi-interest learning framework, REMI. Acknowledging that this comparison may not be entirely equitable, as REMI incorporates numerous advanced techniques and strategies specialized for single-tower MIL.

4.5 Abalation Study(RQ4)

To properly address the effectiveness of IDWLoss and User Representation Repel(URR). We conduct an ablation study for each of these methods.

The result of quadruple-user-tower IDWLoss(without representation repel) is compared with the single-user-tower triplet loss and quadruple-user-tower minimum loss. As we can see in Fig. 3, the IDWLoss empowered model outperforms the triplet loss model and the performance of the model that uses Label Aware Attention and Sampled Softmax quickly regresses due to information leakage and overfitting.

We also compared turning the URR on and off. As we can see in Fig. 4, the URR-enhanced model demonstrated significantly better performance.

Table 3: Comparing Performance of existing dual tower models and their MTMI enhanced version.

Dataset	Metric	Pop	GRU4Rec	**MTMI-GRU**	Youtube DNN	**MTMI-DNN**	Que2Search	**MTMI-Q2S**
Books	R@20	0.0158	0.0441	0.0478	0.0467	**0.049**	0.0453	0.0485
	HR@20	0.0345	0.1004	0.114	0.1043	**0.1155**	0.0969	0.0939
	ND@20	0.0143	0.0378	0.0363	0.0391	0.0394	0.038	**0.0419**
	R@50	0.0281	0.0706	**0.0877**	0.0722	0.0686	0.0702	0.0794
	HR@50	0.0602	0.1553	0.1607	0.1607	0.1594	0.154	**0.1801**
	ND@50	0.0193	0.0443	0.0542	0.0457	0.0404	0.045	**0.0549**
Gowalla	R@20	0.0231	0.09	0.0884	0.0864	**0.0959**	0.0809	0.0893
	HR@20	0.1121	0.3359	**0.3804**	0.3211	0.3642	0.3132	0.377
	ND@20	0.0483	0.1433	0.1582	0.1384	**0.1632**	0.1346	0.1629
	R@50	0.0365	0.1458	0.1617	0.1388	0.1603	0.1546	**0.17**
	HR@50	0.1582	0.4577	0.5496	0.439	0.4086	0.4791	**0.6471**
	ND@50	0.0569	0.1494	**0.1717**	0.1434	0.1544	0.1558	0.1683
Retail Rocket	R@20	0.0129	0.0827	0.0918	0.105	**0.1175**	0.0982	0.1168
	HR@20	0.0252	0.1376	0.1431	0.1711	**0.2297**	0.1216	0.1562
	ND@20	0.0098	0.0517	0.0669	0.0641	**0.0825**	0.0703	0.0678
	R@50	0.0244	0.1371	0.1628	0.1608	**0.1956**	0.1321	0.1641
	HR@50	0.0462	0.2132	0.2088	0.2518	**0.3075**	0.12	0.1408
	ND@50	0.0139	0.0593	0.0623	0.0701	**0.0785**	0.0647	0.0656

Table 4: Comparing the performance of basic MIL models with MTMI-GRU

Dataset	Metric	MIND	ComiRec	MTMI-DNN
Books	R@20	0.042	**0.0557**	0.049
	HR@20	0.0986	0.1142	**0.1155**
	ND@20	0.0357	**0.0446**	0.0394
	R@50	0.0687	**0.0863**	0.0686
	HR@50	0.1533	**0.1796**	0.1594
	ND@50	0.0433	**0.0511**	0.0404
Gowalla	R@20	0.0901	0.0805	**0.0959**
	HR@20	0.3129	0.2901	**0.3642**
	ND@20	0.1331	0.121	**0.1632**
	R@50	0.1456	0.132	**0.1603**
	HR@50	0.4442	**0.4086**	**0.4086**
	ND@50	0.1424	0.131	**0.1544**
Retail Rocket	R@20	0.1171	**0.1304**	0.1175
	HR@20	0.1883	0.1904	**0.2297**
	ND@20	0.0698	0.0689	**0.0825**
	R@50	0.1899	0.1922	**0.1956**
	HR@50	0.2927	0.2895	**0.3075**
	ND@50	**0.0795**	0.0786	0.0785

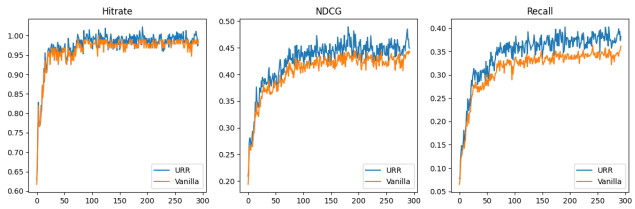

Figure 4: Study on the effect of URR. We use ML-100k dataset for these comparisons.

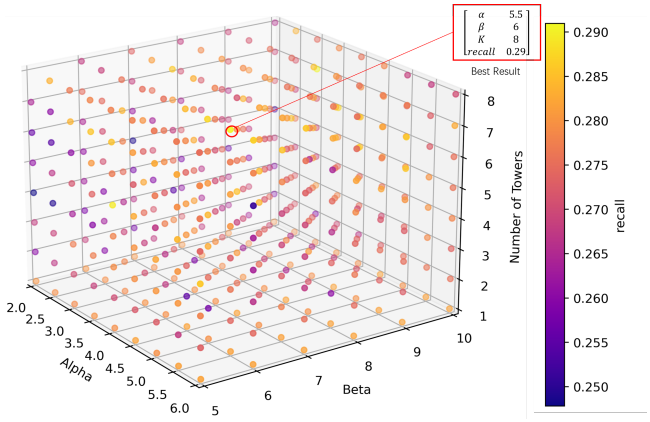

Figure 5: Hyperparameter Scan

4.6 Hyper-parameter Studies(RQ5)

Within this segment, an exhaustive examination of three pivotal hyperparameters in the context of the MTMI model is conducted, specifically addressing the number of interests denoted by K, the coefficient of distance scaling α, and the divergence parameter β. The variable K is selected from the set $2, 4, 6, 8$, α is ascertained from the interval $[0.5, 6]$, and β is chosen from the discrete series $1, 10, 100, 1000$. A comprehensive grid search methodology is employed to meticulously explore every conceivable permutation of these parameters. The entirety of this hyperparameter exploration

is carried out utilizing the Movielens-100k dataset, which is of a modest magnitude, thereby facilitating rapid iteration across successive experimental runs. The results of this exhaustive search can be found at 5, where the performance peaks at $K = 4, \alpha = 2, \beta = 1000$.

5 CONCLUSIONS AND FUTUREWORK

In conclusion, this paper has addressed three critical issues in the field of Multi-interest learning for Candidate Matching: the discrepancy between training and deployment objectives, the challenge of accessing item information, and the complexities of industrial deployment. To resolve these challenges, we introduced the innovative Multi-Tower Multi-Interest Learning framework, known as MTMI. MTMI aligns training and deployment objectives through a novel IDWLoss, offers the capability to access item information via a dedicated item tower, and facilitates the adaptation of industry-standard two-tower models for multi-interest learning. Comparative analyses between traditional Multi-Interest Learning methods and MTMI have revealed substantial improvements in performance.

Although MTMI has significantly enhanced the applicability of Multi-Interest Learning techniques and outperformed single-tower methods under basic settings, it has yet to surpass the state-of-the-art methods in Multi-Interest Learning. Future advancements and refinements hold the potential to elevate MTMI's performance to surpass state-of-the-art methods. There exist several potential avenues for enhancing MTMI, including the refinement of hard negative mining strategies, the development of more intricate and balanced loss functions, and the incorporation of effective inter-tower communication mechanisms. We eagerly anticipate witnessing the progress and innovations in this evolving field.

REFERENCES

[1] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based Recommendations with Recurrent Neural Networks, March 2016. arXiv:1511.06939 [cs].

[2] Wang-Cheng Kang and Julian McAuley. Self-Attentive Sequential Recommendation. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 197–206, November 2018. ISSN: 2374-8486.

[3] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L. Hamilton, and Jure Leskovec. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 974–983, 2018.

[4] Chao Li, Zhiyuan Liu, Mengmeng Wu, Yuchi Xu, Pipei Huang, Huan Zhao, Guoliang Kang, Qiwei Chen, Wei Li, and Dik Lun Lee. Multi-Interest Network with Dynamic Routing for Recommendation at Tmall, April 2019. Issue: arXiv:1904.08030 arXiv:1904.08030 [cs, stat].

[5] Yukuo Cen, Jianwei Zhang, Xu Zou, Chang Zhou, Hongxia Yang, and Jie Tang. Controllable Multi-Interest Framework for Recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, pages 2942–2951, New York, NY, USA, August 2020. Association for Computing Machinery.

[6] Sara Sabour, Nicholas Frosst, and Geoffrey E. Hinton. Dynamic routing between capsules. *Advances in neural information processing systems*, 30, 2017.

[7] Gaode Chen, Xinghua Zhang, Yanyan Zhao, Cong Xue, and Ji Xiang. Exploring periodicity and interactivity in multi-interest framework for sequential recommendation. *arXiv preprint arXiv:2106.04415*, 2021.

[8] Zheng Chai, Zhihong Chen, Chenliang Li, Rong Xiao, Houyi Li, Jiawei Wu, Jingxu Chen, and Haihong Tang. User-Aware Multi-Interest Learning for Candidate Matching in Recommenders. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, pages 1326–1335, New York, NY, USA, July 2022. Association for Computing Machinery.

[9] Yueqi Xie, Jingqi Gao, Peilin Zhou, Qichen Ye, Yining Hua, Jaeboum Kim, Fangzhao Wu, and Sunghun Kim. Rethinking Multi-Interest Learning for Candidate Matching in Recommender Systems, July 2023. arXiv:2302.14532 [cs].

[10] Paul Covington, Jay Adams, and Emre Sargin. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*, pages 191–198, 2016.

[11] Sébastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. On using very large target vocabulary for neural machine translation. *arXiv preprint arXiv:1412.2007*, 2014.

[12] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management - CIKM '13*, pages 2333–2338, San Francisco, California, USA, 2013. ACM Press.

[13] Carlos A. Gomez-Uribe and Neil Hunt. The Netflix Recommender System: Algorithms, Business Value, and Innovation. *ACM Transactions on Management Information Systems*, 6(4):1–19, January 2016.

[14] Fuyu Lv, Taiwei Jin, Changlong Yu, Fei Sun, Quan Lin, Keping Yang, and Wilfred Ng. SDM: Sequential Deep Matching Model for Online Large-scale Recommender System. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 2635–2643, Beijing China, November 2019. ACM.

[15] Santosh Kabbur, Xia Ning, and George Karypis. FISM: factored item similarity models for top-N recommender systems. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 659–667, Chicago Illinois USA, August 2013. ACM.

[16] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, pages 285–295, Hong Kong Hong Kong, April 2001. ACM.

[17] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*, pages 173–182, 2017.

[18] Houyi Li, Zhihong Chen, Chenliang Li, Rong Xiao, Hongbo Deng, Peng Zhang, Yongchao Liu, and Haihong Tang. Path-based Deep Network for Candidate Item Matching in Recommenders. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1493–1502, Virtual Event Canada, July 2021. ACM.

[19] Han Zhu, Daqing Chang, Ziru Xu, Pengye Zhang, Xiang Li, Jie He, Han Li, Jian Xu, and Kun Gai. Joint optimization of tree-based index and deep model for recommender systems. *Advances in Neural Information Processing Systems*, 32, 2019.

[20] Han Zhu, Xiang Li, Pengye Zhang, Guozheng Li, Jie He, Han Li, and Kun Gai. Learning Tree-based Deep Model for Recommender Systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1079–1088, London United Kingdom, July 2018. ACM.

[21] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 811–820, 2010.

[22] Pengfei Wang, Jiafeng Guo, Yanyan Lan, Jun Xu, Shengxian Wan, and Xueqi Cheng. Learning hierarchical representation model for nextbasket recommendation. In *Proceedings of the 38th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 403–412, 2015.

[23] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based Recommendations with Recurrent Neural Networks, March 2016. arXiv:1511.06939 [cs].

[24] Feng Yu, Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. A Dynamic Recurrent Model for Next Basket Recommendation. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 729–732, Pisa Italy, July 2016. ACM.

[25] Ruining He and Julian McAuley. Fusing similarity models with markov chains for sparse sequential recommendation. In *2016 IEEE 16th international conference on data mining (ICDM)*, pages 191–200. IEEE, 2016.

[26] Ruining He, Wang-Cheng Kang, and Julian McAuley. Translation-based Recommendation. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*, pages 161–169, Como Italy, August 2017. ACM.

[27] Xu Chen, Hongteng Xu, Yongfeng Zhang, Jiaxi Tang, Yixin Cao, Zheng Qin, and Hongyuan Zha. Sequential Recommendation with User Memory Networks. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pages 108–116, Marina Del Rey CA USA, February 2018. ACM.

[28] Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*, pages 197–206. IEEE, 2018.

[29] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. Deep Interest Network for Click-Through Rate Prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1059–1068, London United Kingdom, July 2018. ACM.

[30] Shengyu Zhang, Lingxiao Yang, Dong Yao, Yujie Lu, Fuli Feng, Zhou Zhao, Tatseng Chua, and Fei Wu. Re4: Learning to Re-contrast, Re-attend, Re-construct for Multi-interest Recommendation. In *Proceedings of the ACM Web Conference 2022*, pages 2216–2226, Virtual Event, Lyon France, April 2022. ACM.

[31] Mahmut Kaya and Hasan Şakir Bilge. Deep Metric Learning: A Survey. *Symmetry*, 11(9):1066, September 2019. Number: 9 Publisher: Multidisciplinary Digital

Publishing Institute.
[32] Deep Metric Learning: a (Long) Survey.
[33] Bartłomiej Twardowski, Paweł Zawistowski, and Szymon Zaborowski. Metric Learning for Session-based Recommendations, January 2021. arXiv:2101.02655 [cs].
[34] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A Simple Framework for Contrastive Learning of Visual Representations, June 2020. arXiv:2002.05709 [cs, stat].
[35] Joonseok Lee, Sami Abu-El-Haija, Balakrishnan Varadarajan, and Apostol Natsev. Collaborative deep metric learning for video understanding. In *Proceedings of the 24th ACM SIGKDD International conference on knowledge discovery & data mining*, pages 481–490, 2018.
[36] Weihua Chen, Xiaotang Chen, Jianguo Zhang, and Kaiqi Huang. Beyond triplet loss: a deep quadruplet network for person re-identification, April 2017. arXiv:1704.01719 [cs].
[37] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep Metric Learning via Lifted Structured Feature Embedding. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4004–4012, Las Vegas, NV, USA, June 2016. IEEE.
[38] Kihyuk Sohn. Improved Deep Metric Learning with Multi-class N-pair Loss Objective. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.
[39] Oren Rippel, Manohar Paluri, Piotr Dollar, and Lubomir Bourdev. Metric learning with adaptive density discrimination, 2016.
[40] Hyun Oh Song, Stefanie Jegelka, Vivek Rathod, and Kevin Murphy. Deep Metric Learning via Facility Location. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2206–2214, Honolulu, HI, July 2017. IEEE.
[41] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, \Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
[42] Yiqun Liu, Kaushik Rangadurai, Yunzhong He, Siddarth Malreddy, Xunlong Gui, Xiaoyi Liu, and Fedor Borisyuk. Que2Search: Fast and Accurate Query and Document Understanding for Search at Facebook. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 3376–3384, Virtual Event Singapore, August 2021. ACM.
[43] Mahmut Kaya and Hasan Şakir Bilge. Deep metric learning: A survey. *Symmetry*, 11(9):1066, 2019. Publisher: MDPI.
[44] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, pages 43–52, 2015.
[45] Eunjoon Cho, Seth A. Myers, and Jure Leskovec. Friendship and mobility: user movement in location-based social networks. In *Knowledge Discovery and Data Mining*, 2011.
[46] Roman Zykov, Noskov Artem, and Anokhin Alexander. Retailrocket recommender system dataset, 2022.
[47] F. Maxwell Harper and Joseph A. Konstan. The movielens datasets: History and context. *ACM Trans. Interact. Intell. Syst.*, 5(4), dec 2015.
[48] PRAJIT DATTA. Movielens 100k dataset, 2022.

Received 5 October 2023; revised 12 October 2023