Rethinking Multi-Interest Learning for Candidate Matching in Recommender Systems

Yueqi Xie
HKUST
Hong Kong
yxieay@connect.ust.hk

Jingqi Gao
Upstage
Hong Kong
mrgao.ary@gmail.com

Peilin Zhou
HKUST(gz)
China
zhoupalin@gmail.com

Qichen Ye
Peking University
China
yeeeqichen@pku.edu.cn

Yining Hua
MIT
USA
ninghua@mit.edu

Jae Boum Kim
HKUST
Hong Kong
jbkim@cse.ust.hk

Fangzhao Wu
MSRA
China
wufangzhao@gmail.com

Sunghun Kim
HKUST
Hong Kong
hunkim@cse.ust.hk

ABSTRACT

Existing research efforts for multi-interest candidate matching in recommender systems mainly focus on improving model architecture or incorporating additional information, neglecting the importance of training schemes. This work revisits the training framework and uncovers two major problems hindering the expressiveness of learned multi-interest representations. First, the current training objective (i.e., uniformly sampled softmax) fails to effectively train discriminative representations in a multi-interest learning scenario due to the severe increase in easy negative samples. Second, a *routing collapse* problem is observed where each learned interest may collapse to express information only from a single item, resulting in information loss. To address these issues, we propose the REMI framework, consisting of an **I**nterest-aware **H**ard **N**egative mining strategy (IHN) and a **R**outing **R**egularization (RR) method. IHN emphasizes interest-aware hard negatives by proposing an ideal sampling distribution and developing a Monte-Carlo strategy for efficient approximation. RR prevents *routing collapse* by introducing a novel regularization term on the item-to-interest routing matrices. These two components enhance the learned multi-interest representations from both the optimization objective and the composition information. REMI is a general framework that can be readily applied to various existing multi-interest candidate matching methods. Experiments on three real-world datasets show our method can significantly improve state-of-the-art methods with easy implementation and negligible computational overhead. The source code is available at https://github.com/Tokkiu/REMI.

CCS CONCEPTS

- **Information systems → Recommender systems**.

KEYWORDS

Multi-interest Learning, Candidate Matching, Recommender Systems

## 1 INTRODUCTION

Recommender Systems (RS) play a crucial role in various online services [8, 15], such as E-Commerce platforms [20, 26, 48, 52, 57]. They are typically comprised of two stages: candidate *matching* and *ranking* [2, 3]. The candidate matching stage aims to efficiently retrieve thousands of items from the large item corpus via accurately modeling user interests, laying the foundation for fine ranking [3, 13, 31]. Due to its fundamental importance and specific requirements such as high efficiency, candidate matching has attracted increasing research interest [8, 27].

Recently, multi-interest learning-based methods [2, 26] have shown great potential in improving matching performance. These methods explicitly generate users' diverse interest representations from their behavior sequences, breaking the representation bottleneck of using a single generic user embedding. MIND [26] first captures the user's multiple interests through dynamic routing with Capsule Network [39]. Afterward, ComiRec [2] takes diversity into consideration and additionally leverages multi-head attention to encode users' diverse interests. Several recent works [3, 4] have further enabled awareness of periodicity, interactivity, and user profile.

(a)

(b)
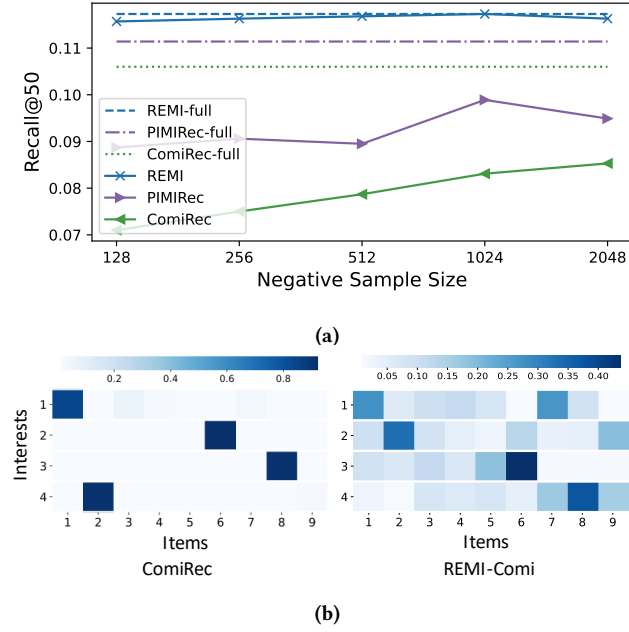
Figure 1: (a): Performance of different methods with practical negative sampling sizes, compared with full softmax (shown as the dashed lines, impractical in real-world scenarios) on Amazon Book. Multi-interest models, i.e., ComiRec and PIMIRec, suffer from dramatical degradation when using uniformly sampled softmax. Training with our REMI framework alleviates the problem and further improves performance. (b): Visualization of an example of item-to-interest routing weights of ComiRec and REMI-enhanced version. REMI helps avoid *routing collapse*.

Despite various model architectures and information explored in multi-interest learning, few efforts are devoted to the training scheme. Previous works generally follow a similar scheme, as depicted in Figure 2 (Left). The user behavior sequence is first encoded into item embeddings and then routed to multiple interest concept representations. To train these multi-interest representations, a common practice is to use the positive target item for each user to select the interest representation closest to the true label. This interest representation is then activated as the user representation and trained similarly to general candidate matching, where a uniformly or log-uniformly sampled softmax [8, 21] is applied to efficiently train the model on a large corpus.

In this work, we **RE**visit the currently used **M**ulti-**I**nterest learning framework and uncover two major problems. First, we raise doubts about the effectiveness of uniformly sampled softmax in multi-interest scenarios. While uniformly sampled softmax has been shown to be effective in training general recommendation systems [46], it falls short in multi-interest recommendation systems, such as ComiRec [2] and PIMIRec [4]. As illustrated in Figure 1 (a), it has significantly worse performance compared to full softmax within a reasonable sample size range (e.g., below a thousand). This is due to the *selected-interest-focused training scheme* in multi-interest learning that the training focuses on a specific

representation of one of the user's multiple interests instead of a general representation of the user. A specific interest concept representation is more susceptible to encountering "easy negatives" compared to an overall preference representation. For example, as shown in Figure 2, consider a user whose historical sequence includes food, electronics, and bags, and the concept related to "bag" is activated for training based on the target item. If we apply uniform sampling, most negatives are easily distinguishable from the representation of the "bag" concept and are therefore "easy negatives", even though some of them may be informative with respect to the overall preference of the user. A lack of informative hard negatives has been proven to impact training [6, 14]. Therefore, uniformly sampled softmax is incapable of training expressive multi-interest representations.

Second, multi-interest models rely on item-to-interest routing to determine the items that form an interest. This routing module is usually a multi-head attention [2] or capsule network [26]. Although the multi-head attention-based routing strategy exhibits higher efficiency and comparable effectiveness than capsule networks [2, 4], we observe that it suffers from *routing collapse*. This means that each interest representation may only be made up of one most relevant item instead of the intended all relevant items. For instance, the representation of interest in "bags" would consist of only one specific bag, not all related bags in the user sequence. Figure 1 (b) shows an example visualization of normalized routing logits with ComiRec-SA [2]. This can result in information loss in the interest representation from the beginning of its formation, hindering the ability to learn expressive multi-interest representations.

To solve these issues, we propose the REMI framework with Interest-aware Hard Negative mining (IHN) and Routing Regularization (RR). First, to alleviate the easy negative problem, we design an ideal distribution that prioritizes, to an adjustable extent, the hard negative items with higher scores related to the currently chosen interest concept (Figure 2, Right). However, direct sampling from this distribution leads to high complexity in computing all the relevance scores and per-user sampling. Therefore, we further propose to leverage the Monte-Carlo importance sampling technique [25, 37] to approximate the above ideal distribution with a few lines of extra code and negligible computation overhead. For the *Routing Collapse* problem, we offer a simple and effective routing regularization strategy on the variance of item-to-interest routing weights for each interest. The two modules improve the expressiveness of the multi-interest representation from two aspects: the optimization objective of training and the composition of information. As demonstrated in Figure 1, the proposed solution not only enable the effectiveness of sampled softmax within a reasonable sampling size but also further improve the achievable performance. Also, the visualization verifies the effectiveness of REMI in solving the *routing collapse* problem.

We conduct extensive experiments on three real-world, large-scale public recommendation datasets. The results show that REMI significantly boosts the performance of existing multi-interest candidate matching methods when applied to them. In addition, in-depth analyses of two main components of our framework demonstrate their effectiveness in solving the targeted problems. Our contributions are summarized as follows:
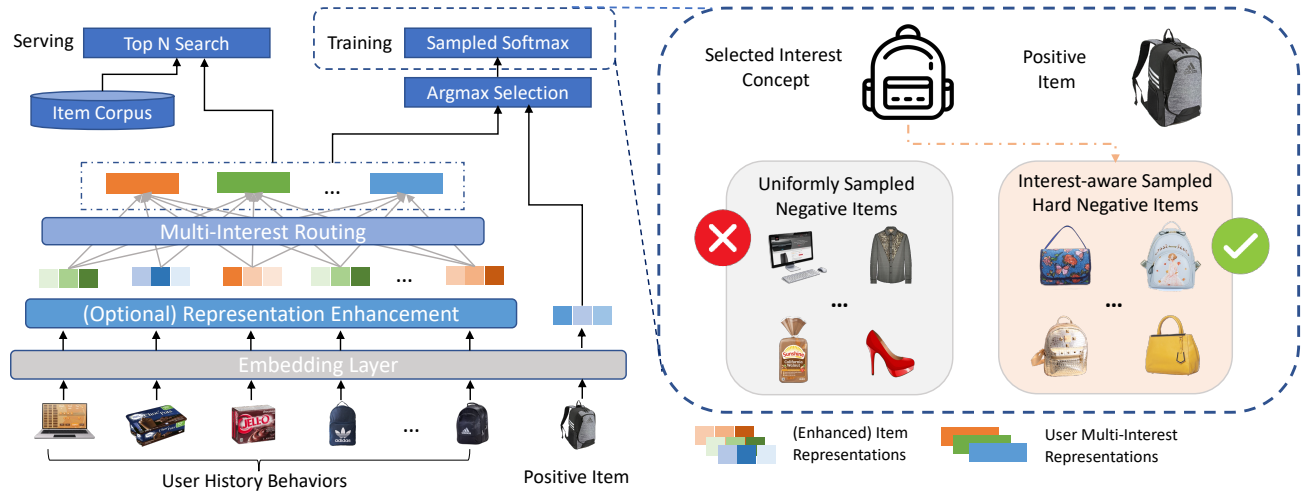
Figure 2: Demonstration of the existing framework (Left) and illustration of different sampling strategies (Right). Although some items in the uniformly sampled negative items (e.g., bread, computer) can be informative hard negative samples when training a general user representation for the sequence consisting of food, electronic devices, and bags, they become unrelated easy negative samples when training the selected interest representation related to "bag" in the existing multi-interest training scheme. The interest-aware hard negative sampling (IHN) alleviates this problem by prioritizing hard negative samples related to the current-selected interest with an adjustable degree, which benefits the learning of discriminative user interest representation.

- We revisit the training scheme of multi-interest learning and uncover two major problems regarding the current negative sampling strategy and routing collapse, which significantly hinder the performance of multi-interest models.
- We propose a simple and effective interest-aware hard negative mining strategy and routing regularization to solve the identified issues and improve the expressiveness of the multi-interest representation from two aspects: the optimization objective of training and the composition of information.
- We conduct extensive experiments which demonstrate that REMI significantly improves the existing multi-interest candidate matching solutions with a few lines of extra code and negligible computation overhead.

2 RELATED WORK

2.1 Candidate Matching

Candidate matching is an essential step in large-scale industrial RS, as it efficiently filters a subset from a large item pool for subsequent fine ranking [3, 13, 31]. Due to its high demand for efficiency, candidate matching models typically utilize lightweight architecture and avoid candidate awareness during user modeling. In the early stage, Collaborative Filtering (CF) based solutions [23, 40] introduce learnable matching between user and candidates and Neural Collaborative Filtering (NCF) [16] enhances the classical CF with multi-layer perceptrons. Afterward, the two-tower DNN structures [8, 19] become highly popular with high computation efficiency, which avoids early interaction between user modeling and candidate modeling. In addition, tree-based structures and graph-based structures have also been explored for deep candidate matching [28, 58, 59]. For example, PDNP [28] builds a retrieval

architecture based on a 2-hop graph, which enables online retrieval with low latency and computation cost. These solutions generally model the user's preference in one vector, potentially limiting the representation ability considering the multi-interest nature of users.

2.2 Multi-Interest Learning

Recent research has found that modeling users' interests as a single vector may be insufficient to accurately capture the complex interaction patterns of users [2, 26]. As a result, multi-interest learning is gradually gaining more attention in both the matching stage and ranking stage of RS. Targeting the matching stage, MIND [26] first adopts a dynamic routing mechanism to aggregate users' historical behaviors into multiple interest capsules [39]. Afterward, ComiRec [2] further investigates multi-head attention-based multi-interest routing for capturing the user's diverse interests and introduces diversity controllable methods. Under a similar framework, PIMIRec [4] and UMI [3] incorporate time information, interactivity, and user profile. Re4 [53] further takes the backward flow into account to regularize the process. Despite improved architectures and enriched information, these works generally follow the training of the general training scheme of candidate matching, where we identify several problems in our work. For the ranking stage, DMIN [47], DemiNet [45], and MGNM [42] explore more architectures and enable candidate-aware Click-through Rate Prediction through a multi-head attention mechanism. Note that in the experimental section, DMIN, DemiNet, and MGNM are not chosen as baselines due to the inapplicable computational complexity for the matching stage.

2.3 Negative Sampling in Recommender Systems

Negative sampling is crucial in RS because it enables the selection of a small set of negative examples instead of using all the non-target items. This reduces the computational cost and speeds up training while still enabling the model to learn meaningful representations. In the early stages, Bayesian Personalized Ranking [36] uniformly samples negative items from items that users had not interacted with. Later, several studies [17, 43] give popular items higher probabilities of being sampled as negative items to suppress popularity bias. However, popularity-based solutions cannot effectively help solve the problem of increasing easy negative samples in multi-interest learning. MNS [49] mixes batch and uniformly sampled negatives to mitigate the selection bias but not targets hard negative samples. Some enhanced methods [10, 29, 34, 56] leverage additional information to improve negative sampling. Specifically, these methods produce informative negative samples for RS in certain application scenarios with available information such as connections in social networks [34, 44, 50, 55], user locations [29, 32, 51], item category information [56], and detailed user feedback [9, 10, 30]. However, these solutions are only applicable in certain scenarios and cannot produce hard negative samples at scale based on current user interest representations. Finally, the most related hard negative sampling strategies are those that produce negative samples according to the current user representation [5, 6, 11, 35, 54]. For example, the work [54] uses a dynamic rejection sampling strategy according to item ranking, the work [11] leverages score-based memory update and variance-based sampling, and a recent work [6] generates hard negative samples for sequential recommendation using the Next Negative Item (NNI) Sampler with Pre-selection and Post-selection. Generally, these solutions allow dynamic sampling by developing complex schemes to explicitly perform context-aware sampling. Our method, in contrast, simply re-weights the loss generated with uniform sampling and effectively addresses the problem in multi-interest learning with negligible cost.

3 PRELIMINARY

3.1 Problem Formulation

Suppose we have a set of users denoted as $\mathcal{U}$ and a large item corpus denoted as $\mathcal{I}$. For each user $u \in \mathcal{U}$, we have a historical behavior sequence $\mathbf{s}^{(u)} = (i_1^{(u)}, i_2^{(u)}, \cdots, i_n^{(u)})$ sorted by time, where n is the maximum sequence length and $i_t^{(u)}$ represents the t-th item in the user behavior sequence. The candidate matching stage in RS aims to efficiently retrieve a subset of items the user is likely to interact with from the huge item corpus $\mathcal{I}$.

3.2 Existing Framework

3.2.1 Model Architecture. Current solutions [2, 26] provide several strategies to form the multi-interest extraction network to get the representation matrices of the multiple interests. Since we focus on the training scheme, we simply abstract the process with two steps 1) item representation (including item embedding and optional representation enhancement) and 2) multi-interest routing (Figure 2).

Let d denote the hidden dimension of the representation, n denote the max sequence length, and K denote the total number of interests, we write step 1) as:

$$\mathbf{H} = \mathcal{F}(s^{(u)}), \tag{1}$$

where $\mathcal{F}$ denotes the model function to encode (and enhance) the items in the user behavior sequence, and $\mathbf{H} \in \mathbb{R}^{d \times n}$ denotes the encoded (and enhanced) item representations. And the learned item embedding matrix for the items in the corpus is denoted as $\mathbf{E} = [\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_{|\mathcal{I}|}] \in \mathbb{R}^{|\mathcal{I}| \times d}$.

Then, step 2) can be formulated as calculating the item-to-interest routing matrix $\mathbf{A} \in \mathbb{R}^{n \times K}$ followed by obtaining the final multi-interest representations $\mathbf{V}_u \in \mathbb{R}^{d \times K}$:

$$\mathbf{A} = \mathcal{G}(\mathbf{H}), \tag{2}$$

$$\mathbf{V}_u = \mathbf{H}\mathbf{A}, \tag{3}$$

where $\mathcal{G}$ is the function to get the routing matrix from the (enhanced) item representations.

3.2.2 Training Scheme. During training, an *argmax* operator is then used to select the most relevant user embedding vector $\mathbf{v}_u$ according to the positive item i^+:

$$
\begin{aligned}
k &= \operatorname{argmax}\left(\mathbf{V}_u^\top \mathbf{e}_{i^+}\right), \\
\mathbf{v}_u &= \mathbf{V}_u[:, k],
\end{aligned}
\tag{4}
$$

where $\mathbf{e}_{i^+}$ is the embedding of the target item i^+, and $\mathbf{v}_u \in \mathbb{R}^{d \times 1}$ is the selected interest concept representation. Then, the problem can be modeled as a classification task. The likelihood of the user u, with the chosen interest k, interacting with the target (positive) item i^+ can be represented with a softmax function:

$$P(i^+|u) = \frac{e^{\mathbf{v}_u^\top \mathbf{e}_{i^+}}}{\sum_{i \in \mathcal{I}} e^{\mathbf{v}_u^\top \mathbf{e}_i}} = \frac{e^{\mathbf{v}_u^\top \mathbf{e}_{i^+}}}{e^{\mathbf{v}_u^\top \mathbf{e}_{i^+}} + \sum_{i^- \in \mathcal{I} \setminus \{i^+\}} e^{\mathbf{v}_u^\top \mathbf{e}_{i^-}}}. \tag{5}$$

The loss function of the model is to minimize the negative log-likelihood of all the positive pairs (u, i^+) in the training dataset $\mathcal{D}$:

$$\mathcal{L}_{\mathcal{SM}}(\theta) = \frac{1}{|\mathcal{D}|} \sum_{(u,i^+) \in \mathcal{D}} -logP(i^+|u). \tag{6}$$

Due to the computation complexity of the denominator with huge amounts of negative items in Equation 5, multi-interest learning typically follows the common practice in general large-corpus candidate matching tasks to use a generic sampled softmax technique [8, 21] with log-uniform sampling or uniform sampling to train the model. We detailly discuss the sampled softmax strategy in Section 3.3.

3.2.3 Serving Scheme. For serving, the multi-interest representations $\mathbf{V}_u$ are obtained for each user u. Then each interest embedding retrieves top-N items from $\mathcal{I}$ based on the inner production using Faiss [22]. Afterward, the final score $f(u, i)$ for the item i in the retrieved set for user u is calculated as:

$$f(u, i) = \max_{1 \le k \le K} (v_u^{k\top} \mathbf{e}_i), \tag{7}$$

where $v_u^k \in \mathbb{R}^{d \times 1}$ is the k-th interest representation. Finally, the top-N scored items are recommended to the user u.

3.3 Sampled Softmax Loss

This section revisits the sampled softmax strategy in a multi-interest learning context. Sampled softmax function reduces computation complexity by only considering the positive class and sampled negative classes. A smaller negative sample size indicates higher training efficiency. A practical sampling size would be within thousands. Note that in prior works [2, 4], the negative sampling size is claimed to be 10 in the paper, but their official codes[1] [2] show that the sample size is set to $10 * batchsize$ per batch, and the negative samples are shared in the batch. Therefore, the negative sample size is commonly set to 1280 to pair with one positive training sample in practice in the prior studies.

There are multiple variants of sampled softmax [8, 21], and the prior studies utilize the version implemented in TensorFlow [1]:

$$\mathcal{L}_{SSM}(\theta) = \frac{1}{|\mathcal{D}|} \sum_{(u,i^+)\in\mathcal{D}} -\log \frac{e^{\mathbf{v}_u^\top \mathbf{e}_{i^+} - \log Q(i^+)}}{e^{\mathbf{v}_u^\top \mathbf{e}_{i^+} - \log Q(i^+)} + \sum_{i=1}^{L} e^{\mathbf{v}_u^\top \mathbf{e}_{i^-} - \log Q(i^-)}}, \tag{8}$$

where L is the negative sampling size, and i^- is sampled from the distribution Q. $\log Q$ is the correction term in sampled softmax and can be simply removed with uniform sampling or the strategy in [21]. In prior works [2–4, 53], Q is often generic sampling distributions such as log-uniform and uniform sampling, which are shown to perform relatively well in general RS learning [46]. Specifically, using log-uniform sampling on the item set sorted by popularity gives the popular items a higher probability of being selected as negative samples.

However, in multi-interest scenarios, such a solution does not perform well due to the *selected-interest-focused training scheme* in multi-interest learning. As discussed in section 1, uniformly or log-uniformly sampled items are more likely to be easy negatives for the selected specific interest from multiple interests compared with general user representation, reducing the training effectiveness of the uniformly sampled softmax function.

4 METHODOLOGY

REMI is a general training framework which can be readily applied to various multi-interest learning methods [2–4] without architecture modification and boost their performance. In this section, we detailly introduce two main components of REMI, interest-aware negative mining (IHN) and routing regularization (RR), which solve the problem of the increase of easy negatives and routing collapse in multi-interest learning, respectively. Then we summarize the overall training objective.

4.1 Interest-aware Hard Negative Mining (IHN) for Multi-Interest Learning

To alleviate the problem caused by the severe increase of easy negatives in multi-interest scenarios with generic sampled softmax, we first design an ideal sampling distribution, followed by a strategy to approximate the ideal sampling with only a few lines of extra codes and negligible computation overhead.

Ideal Sampling Distribution Design. We follow two principles in designing the ideal distribution:

[1]https://github.com/THUDM/ComiRec
[2]https://github.com/ChenGaoDe/PIMI_Rec

- *The informative negative samples are those the selected interest concept is likely to be misclassified to interact with.*
- *The degree of hardness should be adjustable since the optimal effective hardness of negative samples depends on the sample size and dataset.*

Therefore, we propose a negative sampling distribution q_β, which is subjective to the chosen interest concept $\mathbf{v}_u$, and assign the hard negative items with higher probabilities of being sampled. Inspired by the sampling strategy in contrastive learning [38], we propose q_β as follows:

$$q_\beta(i^-) \propto e^{(\beta \mathbf{v}_u^\top \mathbf{e}_{i^-})}, \tag{9}$$

with concentration parameter $\beta \geq 0$. The inner product measures the similarity between the interest and item embeddings. A larger inner product indicates a greater similarity and a harder example. β controls the hardness level for optimal training effectiveness. Then the objective in the sampled softmax in Equation 8 can be re-written as:

$$\mathcal{L}_{SSM}(\theta) = \frac{1}{|\mathcal{D}|} \sum_{(u,i^+)\in\mathcal{D}} -\log \frac{e^{\mathbf{v}_u^\top \mathbf{e}_{i^+}}}{e^{\mathbf{v}_u^\top \mathbf{e}_{i^+}} + L\mathbb{E}_{i^-\sim q_\beta}\left[e^{\mathbf{v}_u^\top \mathbf{e}_{i^-}}\right]}, \tag{10}$$

where the correction term is omitted for simplicity. However, directly sampling from q_β also suffers from great computation complexity to calculate all the $e^{\beta \mathbf{v}_u^\top \mathbf{e}_{i^-}}$ for huge amounts of items. In fact, its computation complexity is equivalent to full softmax. As a result, our next research question becomes: Is it possible to approximate $\mathbb{E}_{i^-\sim q_\beta}\left[e^{\mathbf{v}_u^\top \mathbf{e}_{i^-}}\right]$ without explicitly sampling from q_β?

Importance Sampling-based Approximation. To approximate the ideal distribution at a low cost, we resort to the Monte-Carlo importance sampling techniques [25, 37] to sample from a simple uniform distribution p as follows:

$$\mathbb{E}_{i^-\sim q_\beta}\left[e^{\mathbf{v}_u^\top \mathbf{e}_{i^-}}\right] = \mathbb{E}_{i^-\sim p}\left[e^{\mathbf{v}_u^\top \mathbf{e}_{i^-}} q_\beta/p\right] = \mathbb{E}_{i^-\sim p}\left[e^{(\beta+1)\mathbf{v}_u^\top \mathbf{e}_{i^-}}/Z_\beta\right], \tag{11}$$

where Z_β is the partition function that can be empirically estimated over p:

$$\widehat{Z}_\beta = \mathbb{E}_{i^-\sim p}\left[e^{\beta \mathbf{v}_u^\top \mathbf{e}_{i^-}}\right] = \frac{1}{L}\sum_{i=1}^{L} e^{\beta \mathbf{v}_u^\top \mathbf{e}_{i^-}}. \tag{12}$$

Since the approximation only re-weights the objective for each sample and keeps the sampling procedure a batch-wise uniform sampling, such a method only leads to a negligible computation overhead. In addition, the implementation only requires a few lines of extra code and can be readily incorporated into the existing training framework (pseudocode provided in Appendix A).

Analysis. We then discuss the influence of concentration parameter β starting from two extreme cases. When $\beta = 0$, our solution is equivalent to uniform sampling since all the items have the same probability to be sampled. When $\beta \to \infty$, the loss function tends to focus on the hardest negative samples for a certain user since the hardest samples dominate the calculation of expectation in Equation 11. Note that focusing on the hardest negative samples does not necessarily lead to better effectiveness since there are false negative samples in recommendation scenarios, and easy negatives are also important for training recommendation models [13]. Choosing a proper β between two extreme cases to improve the effectiveness

and efficiency of sampled softmax-based training. We include more empirical case studies on β in Section 5.6.2.

4.2 Routing Regularization for Routing Collapse

This section discusses the *Routing Collapse* problem and provides a simple yet effective regularization-based solution.

Routing Collapse. A key module in multi-interest learning is items-to-interests routing, which is typically achieved with multi-head attention or capsule networks [2, 26]. Multi-head attention-based multi-interest routing demonstrates higher efficiency and comparable performance compared with capsule networks [2, 4], and therefore they are widely applied as the base structure in recent work [2–4, 53]. However, we observe that after training for multiple epochs, the interests tend to over-focus on single items in the behavior sequence, as shown in Figure 1 (b). In this case, only a small portion of the items in the user history is considered, which impacts the expressiveness of multi-interest representations from the beginning of its composition information. The prior study [12] on attention mechanism also observes similar problems when solely applying attention without MLP. Such collapse can be viewed as the model falling to a locally optimal solution.

Routing Regularization. We then propose a simple and effective solution to avoid *Routing Collapse* in our scenarios. Unlike most regularization done in the representation space [41, 53], we find that the collapse was caused by the sparsity of item-to-interest routing matrix $\mathbf{A} \in \mathbb{R}^{n \times K}$, so we introduce the variance regularizer on the routing weights to eliminate sparsity and effectively address the problem. Specifically, we calculate the regularization term with the following equations:

$$\mathbf{C} = (\mathbf{A} - \bar{\mathbf{A}})^\top (\mathbf{A} - \bar{\mathbf{A}}),$$
$$\mathcal{L}_{reg} = \| \operatorname{diag}(\mathbf{C}) \|_F^2, \tag{13}$$

where $\bar{\mathbf{A}}$ is the mean of $\mathbf{A}$ along the first axis, $\mathbf{C} \in \mathbb{R}^{K \times K}$ represents the covariance matrices of routing weights for different interests, $\operatorname{diag}(\mathbf{C})$ represents the extraction and construction of a diagonal matrix, and $\| \cdot \|_F$ denotes the Frobenius norm of matrices. Overall, our training objective can be formed as

$$\mathcal{L} = \mathcal{L}_{SSM} + \lambda \mathcal{L}_{reg}, \tag{14}$$

where λ is the hyper-parameter that balances two losses. The inference procedure is the same as prior works [2, 53], as introduced in Section 3.2.3.

5 EXPERIMENTS

We conduct experiments on three large-scale real-world datasets to answer the following research questions:

- **RQ1:** Can REMI help achieve state-of-the-art performance on candidate matching?
- **RQ2:** Can REMI be combined with different multi-interest learning models and boost their performance?
- **RQ3:** Can the Interest-aware Hard Negative mining strategy (IHN) solve the problems regarding sampled softmax for effective and efficient training?
- **RQ4:** Can Routing Regularization (RR) solve the problem regarding routing collapse?

Table 1: Statistics of datasets.

Dataset	# users	# items	# interactions
Amazon Books	603,668	367,982	8,898,041
Gowalla	65,506	174,605	2,061,264
RetailRocket	33,708	81,635	356,840

- **RQ5:** What is the effect of different components and hyper-parameters in REMI?

5.1 Experimental Settings

5.1.1 Dataset. We select three large-scale public datasets to evaluate the effectiveness of REMI:

- **Amazon** [33][3]. A dataset that consists of product views from the widely used Amazon platform. We choose the largest subset Book with various types of books for evaluation. The maximum sequence length is set to 20.
- **Gowalla** [7]. A typical checking-in dataset built from a location-based social networking website. The maximum sequence length is set to 40.
- **RetailRocket**[4]. An E-Commerce dataset that contains multiple types of user behaviors over a 4-month period. We only use the view events. The maximum sequence length is set to 20.

The dataset preprocessing follows the prior study [2]. We remove all items and users that occur less than five times in these datasets. All the interactions are regarded as implicit feedback. The statistics of the three datasets after preprocessing are summarized in Table 1.

5.1.2 Training and Evaluation Setup. We follow the prior studies [2, 4] to partition the training, validation, and test sets with a ratio of 8:1:1 regarding unique users and train models using the entire sequence of the users in the training set. For evaluation, we use the first 80% of the user behavior sequence to infer the user embeddings and compute the metrics with the remaining 20% items in the sequence. More details can be found in [2, 4]. We adopt the widely used metrics, i.e., **Recall**, **Hit Rate**, and **NDCG** (Normalized Discounted Cumulative Gain)[5], to evaluate our proposed solution. The metrics are computed with the top 20/50 matched candidates.

5.1.3 Baseline Models. We use two types of baselines for performance comparison: general candidate matching and state-of-the-art multi-interest models. The baselines are introduced as follows:

- **Most Popular.** A basic algorithm that recommends the most popular items to users.
- **GRU4Rec** [18]. An RNN-based model that captures sequential patterns.
- **YouTube DNN** [8]. A two-tower DNN model that pools behavior embeddings followed by MLP layers to get the final user representation.
- **MIND** [26]. The first multi-interest framework that captures diverse interests with capsule networks.

---

[3]http://jmcauley.ucsd.edu/data/amazon/
[4]https://www.kaggle.com/retailrocket/ecommerce-dataset
[5]The calculation of NDCG follows the officially revised code of ComiRec: https://github.com/THUDM/ComiRec, which is in line with the paper's description of the calculation of NDCG but does not align with the results in the original paper.

- **ComiRec-SA** [2] An advanced multi-interest framework that allows diversity control and further introduces multi-head attention to model multiple interests. We use the ComiRec-SA version, which demonstrates comparable performance with the DR version with a stabler and faster training.
- **Re4** [53]. An advanced multi-interest framework that takes the backward flow into account to regularize the process. Note that the sampling size is set same as other methods to ensure fairness.
- **UMI_{HN}** [3]. A variant of UMI. UMI proposes to use user profiles when generating users' multiple interests and proposes an HN strategy to enhance training. Since the HN is shown to be the most effective part of the method and the user profile is not available in the datasets, we use **UMI_{HN}** as our baseline.
- **PIMIRec** [4]. A model that considers time information and interactivity among items when modeling user interests.

5.1.4 Implementation Details. We implement our work and baseline methods with PyTorch 1.8 in Python 3.7. We build REMI on **ComiRec-SA** by default for the best efficiency, except for Section 5.3. All parameters are set as follows if not otherwise noted: following [2], the number of dimensions d for embeddings is set to 64, the batch size is set to 128, and the maximum number of training iterations is set to 1 million for all the models. As discussed in Section 3.3, the batch-wise shared negative training sample size is set to 128 * 10 as prior works for fair comparison in the main study. It is worth mentioning that REMI can get comparable performance with a smaller sample size, which further enables efficient training in practice. We discuss this in Section 5.4. In the overall comparisons (Table 2, 3), we report the best performance with the interest number $K \in \{2, 4, 6, 8\}$ for baseline multi-interest models and $K = 4$ for REMI. In other analysis studies, $K = 4$ is used by default. We use the Adam optimzer [24] for training. For MIND and Re4, we grid search the optimal learning rates from $\{1e-3, 3e-3, 5e-3\}$ and weight decay from $\{1e-6, 1e-5\}$. We use $lr = 1e-3$ for REMI and other baselines. Other hyperparameters for the baselines are set according to the original paper. For the hyperparameters for REMI, we search β in $\{0.1, 1, 4, 10\}$ for different conditions and set λ to $1e2$.

5.2 Overall Performance (RQ1)

We summarize the overall performance in Table 2. Based on the result, we can make the following observations:

First, the personalized deep learning-based models perform much better than MostPopular, indicating the importance of modeling user history behavior in RS. In addition, vanilla multi-interest methods (i.e., MIND and ComiRec) do not always outperform the general learning models like YouTube DNN and GRU4Rec. This is in line with some prior studies on candidate matching [3], which may be attributed to the unsuitable training scheme.

Also, the enhanced multi-interest models (Re4 and UMI) help to improve the multi-interest model compared with basic ComiRec and MIND, demonstrating the effectiveness of including backward flow and better training procedures. Note that the advanced training strategy in UMI, despite a similar name to our IHN strategy, is from a different aspect. We present a more detailed analysis

and compatibility experiments in Section 5.3. We also observe that PIMIRec has the best performance among the baselines, in which the interactivity module demonstrates the most significant performance improvement [4]. It also, to some extent, alleviates the collapse problem through the early interactions among different items. Moreover, in our experiment, we notice that Re4 and PIMIRec require a long training time due to the utilization of complex backward flow and graph structure, leading to efficiency concerns in these solutions.

Finally, REMI, when built on the simple ComiRec-SA, significantly outperforms the state-of-the-art multi-interest models and general models on all three datasets regarding all metrics. Through solving two key questions of the multi-interest learning process, the REMI framework remarkably improves the capacity of existing multi-interest models without architecture modification or extra computation burden. These results demonstrate the importance of the identified problems and the effectiveness of our solutions.

5.3 Enhancement Study (RQ2)

As a general training framework with simple implementation, REMI can be applied to the existing multi-interest models to further boost their performance. In this section, we include a comparison of the original ComiRec-SA [2], PIMIRec [4], UMI_{HN} [3] with their REMI-enhanced versions. The results are shown in Table 3.
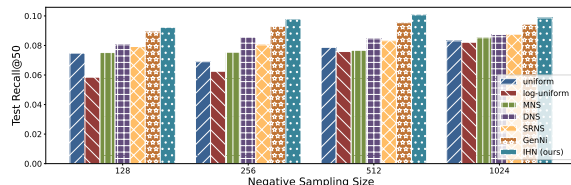
ComiRec represents the basic multi-interest framework with no additional information or enhancement. We observe improvements from **27.95% to 69.81%** on this simple model on the three datasets. These improvements reflect the importance of the identified problems and the effectiveness of the proposed solutions in basic multi-interest learning. They also suggest the potential for the application of REMI in large-scale industry datasets since it helps achieve strong performance even with simple and light architecture. PIMIRec represents an advanced multi-interest model that integrates additional information (i.e., periodicity) and incorporates complex architecture such as GNN to tackle the evolutional user interests. By explicitly regularizing the routing process and incorporating effective IHN training techniques, our solution can further enhance the performance of PIMI. We also combine REMI with UMI_{HN}. UMI_{HN} improves the training by applying a candidate-aware calculation of scores instead of selecting one interest for all candidates as Equation 4. However, it still attaches the same importance to the uniformly-sampled negative items and also leads to lower efficiency, while the IHN in REMI approximates an interest-aware hard negative sampling with negligible cost. The results show that REMI is compatible with their optimization and leads to further performance improvement.

5.4 Effectiveness of Interest-aware Hard Negative Mining (RQ3)

In this section, we first present a quantitative analysis of the efficiency and effectiveness of the proposed IHN (Interest-aware Hard Negative Mining) method for multi-interest model training. Afterward, we present several case studies to substantiate whether the IHN method assigns higher weights to those interest-aware hard negatives.

Table 2: Performance comparison of different methods on three datasets. The bold and underlined numbers represent the best and second-best results, respectively. The last column shows the relative improvement of REMI over the best baseline methods. We conduct paired t-test of REMI and the best baseline, and the improvement of REMI is significant with $p \leq 0.01$ for all the settings.

Dataset	Metric	Pop	GRU4Rec	Youtube DNN	MIND	ComiRec	Re4	UMI_{HN}	PIMIRec	REMI	Improv.
Amazon Books	R@20	0.0158	0.0441	0.0467	0.0420	0.0557	0.0597	<u>0.0690</u>	0.0682	**0.0826**	+19.7%
	HR@20	0.0345	0.1004	0.1043	0.0986	0.1142	0.1240	<u>0.1423</u>	0.1411	**0.1650**	+16.0%
	ND@20	0.0143	0.0378	0.0391	0.0357	0.0446	0.0476	<u>0.0527</u>	0.0526	**0.0623**	+18.2%
	R@50	0.0281	0.0706	0.0722	0.0687	0.0863	0.0690	0.1053	<u>0.1056</u>	**0.1189**	+12.6%
	HR@50	0.0602	0.1553	0.1607	0.1533	0.1796	0.1975	0.2059	<u>0.2062</u>	**0.2298**	+11.4%
	ND@50	0.0193	0.0443	0.0457	0.0433	0.0511	0.0576	<u>0.0587</u>	0.0583	**0.0657**	+11.9%
Gowalla	R@20	0.0231	0.0900	0.0864	0.0901	0.0805	0.0843	0.0961	<u>0.1193</u>	**0.1317**	+10.4%
	HR@20	0.1121	0.3359	0.3211	0.3129	0.2901	0.3104	0.3314	<u>0.3843</u>	**0.4145**	+7.9%
	ND@20	0.0483	0.1433	0.1384	0.1331	0.1210	0.1287	0.1391	<u>0.1603</u>	**0.1774**	+10.7%
	R@50	0.0365	0.1458	0.1388	0.1456	0.1320	0.1396	0.1642	<u>0.1951</u>	**0.2062**	+5.7%
	HR@50	0.1582	0.4577	0.4390	0.4442	0.4086	0.4224	0.4719	<u>0.5207</u>	**0.5455**	+4.8%
	ND@50	0.0569	0.1494	0.1434	0.1424	0.1310	0.1410	0.1505	<u>0.1660</u>	**0.1785**	+7.5%
Retail Rocket	R@20	0.0129	0.0827	0.1050	0.1171	0.1304	0.1397	0.1519	<u>0.1828</u>	**0.2109**	+15.3%
	HR@20	0.0252	0.1376	0.1711	0.1883	0.1904	0.2103	0.2364	<u>0.2764</u>	**0.3144**	+13.7%
	ND@20	0.0098	0.0517	0.0641	0.0698	0.0689	0.0785	0.0875	<u>0.1025</u>	**0.1170**	+14.1%
	R@50	0.0244	0.1371	0.1608	0.1899	0.1922	0.2194	0.2423	<u>0.2811</u>	**0.3214**	+14.3%
	HR@50	0.0462	0.2132	0.2518	0.2927	0.2895	0.3174	0.3574	<u>0.3969</u>	**0.4544**	+14.5%
	ND@50	0.0139	0.0593	0.0701	0.0795	0.0786	0.0884	0.0974	<u>0.1093</u>	**0.1274**	+16.6%

Table 3: Generalization experiments. Performance of existing multi-interest models and their REMI-enhanced version.

Dataset	Metric	ComiRec			PIMIRec			UMI_{HN}		
		Original	+REMI	+Δ	Original	+REMI	+Δ	Original	+REMI	+Δ
Books	R@20	0.0557	**0.0826**	+48.29%	0.0682	**0.0834**	+22.28%	0.0690	**0.0828**	+20.00%
	HR@20	0.1142	**0.1650**	+44.48%	0.1411	**0.1678**	+18.92%	0.1423	**0.1672**	+17.49%
	ND@20	0.0446	**0.0623**	+39.68%	0.0526	**0.0643**	+22.24%	0.0527	**0.0637**	+20.87%
	R@50	0.0863	**0.1189**	+37.77%	0.1056	**0.1204**	+14.01%	0.1053	**0.1206**	+14.52%
	HR@50	0.1796	**0.2298**	+27.95%	0.2062	**0.2312**	+12.12%	0.2059	**0.2335**	+13.40%
	ND@50	0.0511	**0.0657**	+28.57%	0.0583	**0.0673**	+15.43%	0.0587	**0.0675**	+14.99%
Gowalla	R@20	0.0805	**0.1317**	+63.60%	0.1193	**0.1327**	+11.23%	0.0961	**0.1328**	+38.18%
	HR@20	0.2901	**0.4145**	+42.88%	0.3843	**0.4231**	+10.09%	0.3314	**0.4194**	+26.55%
	ND@20	0.1210	**0.1774**	+46.61%	0.1603	**0.1785**	+11.35%	0.1391	**0.1772**	+27.39%
	R@50	0.1320	**0.2062**	+56.21%	0.1951	**0.2143**	+9.84%	0.1642	**0.2102**	+28.01%
	HR@50	0.4086	**0.5455**	+33.50%	0.5207	**0.5632**	+8.16%	0.4719	**0.5571**	+18.05%
	ND@50	0.1310	**0.1785**	+36.25%	0.1660	**0.1826**	+10.00%	0.1505	**0.1816**	+20.66%
Retail Rocket	R@20	0.1304	**0.2109**	+61.73%	0.1828	**0.2173**	+18.87%	0.1519	**0.2155**	+41.86%
	HR@20	0.1904	**0.3144**	+65.12%	0.2764	**0.3265**	+18.12%	0.2364	**0.3236**	+36.88%
	ND@20	0.0689	**0.1170**	+69.81%	0.1025	**0.1210**	+18.04%	0.0875	**0.1211**	+38.40%
	R@50	0.1922	**0.3214**	+67.22%	0.2811	**0.3217**	+14.43%	0.2423	**0.3161**	+30.45%
	HR@50	0.2895	**0.4544**	+56.96%	0.3969	**0.4587**	+15.57%	0.3574	**0.4464**	+24.90%
	ND@50	0.0786	**0.1274**	+62.08%	0.1093	**0.1298**	+18.75%	0.0974	**0.1259**	+29.26%

Figure 3: The effectiveness comparison of different negative sampling strategies on the Amazon Book dataset in Recall@50.

Figure 4: Case Study on Amazon Book. IHN assigns higher weights to the hard negatives according to the current interest, thus making the learning of user interest representations more discriminative.

For a fair comparison, we do not include routing regularization and compare IHN to the generic uniform sampling and log-uniform sampling, as well as the state-of-the-art sampling techniques, i.e., MNS [49], DNS [54], SRNS [11], and GenNi [6], when applied to ComiRec-SA. We observe similar results for three datasets, and due to space limitations, we present the results on the Amazon Book dataset. IHN consistently outperforms all baselines in multi-interest

learning, as demonstrated in Figure 3 with efficiency similar to uniform sampling. For sampling techniques that exhibit similar efficiency, such as uniform and log-uniform sampling, we observe that the performance gap between IHN and generic sampling increases as the negative sampling size decreases, indicating the potential of IHN in further improving training efficiency with a smaller negative sampling size without compromising performance. In addition, the commonly adopted log-uniform loss is found to be inferior to uniform sampling when dealing with large item candidate pools. This is due to the fact that the log-uniform distribution assigns significantly small sampling probabilities to unpopular items in such cases. Consequently, this exacerbates the popularity bias and leads to undertrained embeddings for unpopular items.

Next, we present case studies to validate whether **IHN indeed assigns higher weights to interest-aware hard negatives**. We randomly select several users from the Amazon Book Dataset. Since user interest representations are implicit and can not be directly visualized, we display the true positive items that can be used to imply the selected interest concept in the multi-interest learning framework, as demonstrated in Figure 2. We present the negative items with the highest and lowest weights assigned by the IHN

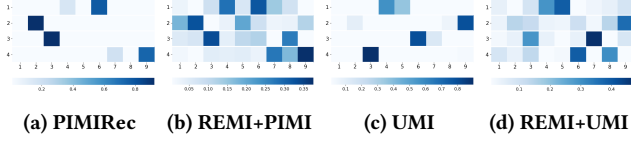(a) PIMIRec   (b) REMI+PIMI   (c) UMI   (d) REMI+UMI

Figure 5: Visualization of examples of routing matrices for PIMIRec, UMI, and their REMI-enhanced versions. REMI solves the routing collapse problem.

Table 4: Comparison of various regularization strategies. We show the metrics@50 on three datasets. The final column in each dataset block displays the HR@50 improvement.

Method	Books				Gowalla				RetailRocket			
	Recall	NDCG	HR	+Δ	Recall	NDCG	HR	+Δ	Recall	NDCG	HR	+Δ
No Reg.	0.0824	0.0464	0.1661	-	0.1271	0.1246	0.3891	-	0.1818	0.0761	0.2738	-
Rep. Reg. [41]	0.0814	0.0481	0.1712	+3.07%	0.1141	0.1261	0.3935	+1.13%	0.1895	0.0822	0.2939	+7.34%
Re4	0.1008	0.0562	0.1975	+18.90%	0.1182	0.1309	0.4060	+4.34%	0.2088	0.0884	0.3174	+15.92%
RR (Ours)	0.1078	0.0606	0.2135	+28.53%	0.1841	0.1626	0.5067	+30.22%	0.2804	0.1145	0.4028	+47.11%

Table 5: Ablation study of Interest-aware Hard Negative Sampling (IHN) and Routing Regularization(RR). We show the metrics@50 on three datasets. The final column in each dataset block displays the HR@50 improvement.

Settings		Books				Gowalla				RetailRocket			
IHN	RR	Recall	NDCG	HR	+Δ	Recall	NDCG	HR	+Δ	Recall	NDCG	HR	+Δ
✗	✗	0.0814	0.0481	0.1661	-	0.1271	0.1246	0.3891	-	0.1818	0.0761	0.2738	-
✓	✗	0.1008	0.0562	0.1975	+18.90%	0.1608	0.1525	0.4715	+21.17%	0.2559	0.0907	0.3307	+20.78%
✗	✓	0.1078	0.0606	0.2135	+28.53%	0.1841	0.1626	0.5067	+30.22%	0.2804	0.1145	0.4028	+47.11%
✓	✓	0.1189	0.0657	0.2298	+38.35%	0.2060	0.1783	0.5455	+40.19%	0.3124	0.1274	0.4544	+65.96%

Table 6: Study on the interest number K. We show the metrics@50 on three datasets.

K	Books			Gowalla			RetailRocket		
	HR	Recall	NDCG	HR	Recall	NDCG	HR	Recall	NDCG
2	0.2182	0.1129	0.0625	0.5503	0.2059	0.1786	0.4461	0.3181	0.1241
4	0.2298	0.1189	0.0657	0.5455	0.2062	0.1785	0.4544	0.3214	0.1274
6	0.2340	0.1216	0.0667	0.5515	0.2065	0.1790	0.4550	0.3210	0.1267
8	0.2360	0.1220	0.0673	0.5565	0.2106	0.1793	0.4470	0.3193	0.1254

method in Figure 4. The left figure shows that when the positive sample relates to manhood and faith, the highest-weighted items are hard negatives related to the heroic spirit, while the lowest-weighted items are entirely unrelated books, such as those about art, which do not provide useful information for training discriminative user embeddings. Similarly, in the right figure, when the positive sample pertains to politics and humanity, particularly feminism, the highest-weighted negative samples of IHN are also related to politics but not feminism, providing more granular supervision signals. In contrast, the lowest-weighted negative samples are unrelated, such as those about cats. These two examples demonstrate that the IHN method effectively assigns higher weights to more meaningful hard negatives that help train discriminative user interest representation while giving lower weights to entirely unrelated negative samples.

5.5 Effectiveness of Routing Regularization (RQ4)

This section verifies the effectiveness of Routing Regularization without adding IHN to ensure fairness with other methods. We compared it to other multi-interest regularization methods, including Re4 [53] and SINE [41]. SINE regulates multi-concepts in the representation space, while Re4 uses backward flow to regulate the training process. However, these methods fail to address the *routing collapse* issue and perform worse than our Routing Regularization, as shown in Table 4. Furthermore, besides the visualization in the introduction, Figure 1 (b), we also visualize the item-to-interest routing for state-of-the-art UMI [3] and PIMI [4] in Figure 5, with and without REMI. Some random samples show that REMI effectively avoids *routing collapse* for UMI and PIMI, where interests are no longer composed of one or two items but fuse all corresponding relevant items from the historical sequence.

5.6 Ablation Study and Hyper-parameter Study (RQ5)

5.6.1 Ablation Studies. Here we present a detailed analysis of the effect of Interest-aware Hard Negative Mining (IHN) and Routing

Regularization (RR) (Table 5). Through solving two key problems in Multi-Interest Recommendation, IHN and RR independently improve performance remarkably. Combined, they boost the performance with a **38.35% to 65.96%** improvement in Hit Rate@50 on the three datasets. Noteworthy, these improvements come with negligible costs of computation and latency because the two proposed strategies only modify the training objective of the original ComiRec-SA model by re-weighting the loss and adding a variance-based regularization term.

5.6.2 Hyperparameters Studies. In this section, we thoroughly study three hyperparameters in REMI: the interest number K, concentration parameter β, and balance parameter λ.

The interest number K. First, we show the performance of different interest numbers K of REMI in Table 6. In the study, we fix the concentration parameter β and balance parameter λ to be 1 and 1e2, respectively. Generally, REMI demonstrates better performance with a larger K, indicating that REMI can leverage the strength of multi-interest encoding and generate more representative and diverse interests when the interest number grows. Furthermore, the optimal number of interests also varies depending on the dataset. For instance, in the case of RetailRocket, where the average user sequence length and the item corpus are relatively small, four interests results in the best performance.

The concentration parameter β. We investigate β in the range of 0.1 to 10 to enable its exponential coverage to cover a majority of values, as shown in Figure 6. In the study, we fix the balance parameter λ and interest number K to be 1e2 and 4, respectively. We observe that larger β does not necessarily lead to improvement. It aligns with the analysis in Section 4.1 that easy negatives are also important to supervise the training of RS, and the hardest samples might be false negatives. Moreover, the optimal choices of β for different datasets differ. Specifically, for Amazon Book and RetailRocket, the optimal β in the set is 4, while setting β as 1 produces the best performance for Gowalla. This can be attributed to the different characteristics of the dataset, such as the property of items. The result also demonstrates the importance of an adjustable β under different scenarios.
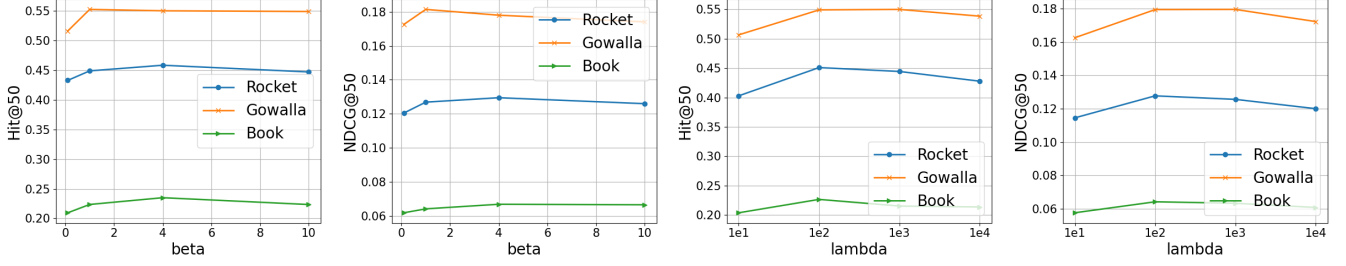
Figure 6: Study on balance parameter β and λ. We show Hit Rate@50 and NDCG@50 on three datasets.

The balance parameter λ. Figure 6 presents the Hit Rate@50 and NDCG@50 of REMI with different λ. In order to balance the loss between two differing magnitudes, we explore the effects of varying the magnitude of λ. In the study, we fix the interest number K and concentration parameter β to be 4 and 1, respectively. It shows that as λ increases, the performance first increases and then decreases. That aligns with our intuition that we need to properly select λ to avoid *routing collapse* while not influencing the training to force the routing to be average. We roughly tune it from $\{1e1, 1e2, 1e3, 1e4\}$ and fix λ to be $1e2$ for other studies.

6 CONCLUSIONS

This work revisits the existing training scheme of multi-interest learning and reveals the issues of increased easy negatives and routing collapse. To address these challenges, we propose REMI as a general training framework. REMI first mitigates the problem of easy negatives with an ideal interest-aware hard negative sampling distribution and an approximation method to achieve the goal at a negligible computational cost. REMI also incorporates a novel routing regularization to avoid routing collapse and further improve the modeling capacity of multi-interest models. Extensive experiments demonstrate that training with the REMI framework significantly boosts the performance of existing methods. We hope this training framework will further contribute to future research on promising multi-interest learning and highlight the importance of exploring training procedures in related fields.

A IMPLEMENTATION OF IHN

We provide PyTorch-style pseudocode in Algorithm 1 for our proposed IHN, compared with the original uniformly sampled softmax.

Algorithm 1 IHN.

```
 1: # pos : exp of inner product for the positive item
 2: # neg : exp of inner products for the sampled negative items
 3: # beta : concentration parameter
 4:
 5: # objective with uniform negative sampling
 6: Neg = neg.sum()
 7: loss_uniform = -log(pos / (pos + Neg))
 8:
 9: # objective with IHN
10: imp = (beta* log(neg)).exp()
11: Neg = (imp*neg).sum() / imp.mean()
12: loss_IHN = -log(pos / (pos + Neg))
```

REFERENCES

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. https://www.tensorflow.org/ Software available from tensorflow.org.
[2] Yukuo Cen, Jianwei Zhang, Xu Zou, Chang Zhou, Hongxia Yang, and Jie Tang. 2020. Controllable multi-interest framework for recommendation. In *SIGKDD*. 2942–2951.
[3] Zheng Chai, Zhihong Chen, Chenliang Li, Rong Xiao, Houyi Li, Jiawei Wu, Jingxu Chen, and Haihong Tang. 2022. User-Aware Multi-Interest Learning for Candidate Matching in Recommenders. In *SIGIR*. 1326–1335.

[4] Gaode Chen, Xinghua Zhang, Yanyan Zhao, Cong Xue, and Ji Xiang. 2021. Exploring Periodicity and Interactivity in Multi-Interest Framework for Sequential Recommendation. In *IJCAI*, Zhi-Hua Zhou (Ed.). ijcai.org, 1426–1433. https://doi.org/10.24963/ijcai.2021/197
[5] Jin Chen, Defu Lian, Binbin Jin, Kai Zheng, and Enhong Chen. 2022. Learning Recommenders for Implicit Feedback with Importance Resampling. In *WWW*. 1997–2005.
[6] Yongjun Chen, Jia Li, Zhiwei Liu, Nitish Shirish Keskar, Huan Wang, Julian McAuley, and Caiming Xiong. 2022. Generating Negative Samples for Sequential Recommendation. *arXiv preprint arXiv:2208.03645* (2022).
[7] Eunjoon Cho, Seth A Myers, and Jure Leskovec. 2011. Friendship and mobility: user movement in location-based social networks. In *SIGKDD*. 1082–1090.
[8] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *RecSys*. 191–198.
[9] Jingtao Ding, Fuli Feng, Xiangnan He, Guanghui Yu, Yong Li, and Depeng Jin. 2018. An improved sampler for bayesian personalized ranking by leveraging view data. In *Companion Proceedings of the The Web Conference 2018*. 13–14.
[10] Jingtao Ding, Yuhan Quan, Xiangnan He, Yong Li, and Depeng Jin. 2019. Reinforced Negative Sampling for Recommendation with Exposure Data.. In *IJCAI*. 2230–2236.
[11] Jingtao Ding, Yuhan Quan, Quanming Yao, Yong Li, and Depeng Jin. 2020. Simplify and robustify negative sampling for implicit collaborative filtering. *Advances in Neural Information Processing Systems* 33 (2020), 1094–1105.
[12] Yihe Dong, Jean-Baptiste Cordonnier, and Andreas Loukas. 2021. Attention is not all you need: Pure attention loses rank doubly exponentially with depth. In *ICML*. PMLR, 2793–2803.
[13] Carlos A Gomez-Uribe and Neil Hunt. 2015. The netflix recommender system: Algorithms, business value, and innovation. *TMIS* 6, 4 (2015), 1–19.
[14] Mihajlo Grbovic and Haibin Cheng. 2018. Real-time personalization using embeddings for search ranking at airbnb. In *SIGKDD*. 311–320.
[15] Ido Guy, Naama Zwerdling, Inbal Ronen, David Carmel, and Erel Uziel. 2010. Social media recommendation based on people and tags. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*. 194–201.
[16] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In *WWW*. ACM, 173–182.
[17] Xiangnan He, Hanwang Zhang, Min-Yen Kan, and Tat-Seng Chua. 2016. Fast matrix factorization for online recommendation with implicit feedback. In *SIGIR*. 549–558.

[18] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based Recommendations with Recurrent Neural Networks. In *ICLR*, Yoshua Bengio and Yann LeCun (Eds.). http://arxiv.org/abs/1511.06939

[19] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry P. Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *CIKM*. ACM, 2333–2338.

[20] Hyunwoo Hwangbo, Yang Sok Kim, and Kyung Jin Cha. 2018. Recommendation system development for fashion retail e-commerce. *Electronic Commerce Research and Applications* 28 (2018), 94–101.

[21] Sébastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. 2015. On using very large target vocabulary for neural machine translation. *ACL*.

[22] Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data* 7, 3 (2019), 535–547.

[23] Santosh Kabbur, Xia Ning, and George Karypis. 2013. Fism: factored item similarity models for top-n recommender systems. In *SIGKDD*. 659–667.

[24] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[25] Tuen Kloek and Herman K Van Dijk. 1978. Bayesian estimates of equation system parameters: an application of integration by Monte Carlo. *Econometrica: Journal of the Econometric Society* (1978), 1–19.

[26] Chao Li, Zhiyuan Liu, Mengmeng Wu, Yuchi Xu, Huan Zhao, Pipei Huang, Guoliang Kang, Qiwei Chen, Wei Li, and Dik Lun Lee. 2019. Multi-interest network with dynamic routing for recommendation at Tmall. In *CIKM*. 2615–2623.

[27] Houyi Li, Zhihong Chen, Chenliang Li, Rong Xiao, Hongbo Deng, Peng Zhang, Yongchao Liu, and Haihong Tang. 2021. Path-based Deep Network for Candidate Item Matching in Recommenders. In *SIGIR*. 1493–1502.

[28] Houyi Li, Zhihong Chen, Chenliang Li, Rong Xiao, Hongbo Deng, Peng Zhang, Yongchao Liu, and Haihong Tang. 2021. Path-based Deep Network for Candidate Item Matching in Recommenders. In *SIGIR*. ACM, 1493–1502.

[29] Wei Liu, Zhi-Jie Wang, Bin Yao, and Jian Yin. 2019. Geo-ALM: POI Recommendation by Fusing Geographical Information and Adversarial Learning Mechanism.. In *IJCAI*, Vol. 7. 1807–1813.

[30] Babak Loni, Roberto Pagano, Martha Larson, and Alan Hanjalic. 2016. Bayesian personalized ranking with multi-channel user feedback. In *RecSys*. 361–364.

[31] Fuyu Lv, Taiwei Jin, Changlong Yu, Fei Sun, Quan Lin, Keping Yang, and Wilfred Ng. 2019. SDM: Sequential deep matching model for online large-scale recommender system. In *CIKM*. 2635–2643.

[32] Jarana Manotumruksa, Craig Macdonald, and Iadh Ounis. 2017. A personalised ranking framework with multiple sampling criteria for venue recommendation. In *CIKM*. 1469–1478.

[33] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *SIGIR*. 43–52.

[34] Dimitrios Rafailidis. 2019. Bayesian deep learning with trust and distrust in recommendation systems. In *WI*. IEEE, 18–25.

[35] Steffen Rendle and Christoph Freudenthaler. 2014. Improving pairwise learning for item recommendation from implicit feedback. In *WSDM*. 273–282.

[36] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618* (2012).

[37] Christian P Robert, George Casella, and George Casella. 1999. *Monte Carlo statistical methods*. Vol. 2. Springer.

[38] Joshua Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. 2021. Contrastive learning with hard negative samples. (2021).

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

[40] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *WWW*. 285–295.

[41] Qiaoyu Tan, Jianwei Zhang, Jiangchao Yao, Ninghao Liu, Jingren Zhou, Hongxia Yang, and Xia Hu. 2021. Sparse-interest network for sequential recommendation. In *WSDM*. 598–606.

[42] Yu Tian, Jianxin Chang, Yanan Niu, Yang Song, and Chenliang Li. 2022. When Multi-Level Meets Multi-Interest: A Multi-Grained Neural Model for Sequential Recommendation. In *SIGIR*. 1632–1641.

[43] Riku Togashi, Mayu Otani, and Shin'ichi Satoh. 2021. Alleviating cold-start problems in recommendation through pseudo-labelling over knowledge graph. In *WSDM*. 931–939.

[44] Xin Wang, Wei Lu, Martin Ester, Can Wang, and Chun Chen. 2016. Social recommendation with strong and weak ties. In *CIKM*. 5–14.

[45] Yule Wang, Qiang Luo, Yue Ding, Dong Wang, and Hongbo Deng. 2021. DemiNet: Dependency-Aware Multi-Interest Network with Self-Supervised Graph Learning for Click-Through Rate Prediction. *CoRR* abs/2109.12512 (2021).

[46] Jiancan Wu, Xiang Wang, Xingyu Gao, Jiawei Chen, Hongcheng Fu, Tianyu Qiu, and Xiangnan He. 2022. On the Effectiveness of Sampled Softmax Loss for Item Recommendation. *arXiv preprint arXiv:2201.02327* (2022).

[47] Zhibo Xiao, Luwei Yang, Wen Jiang, Yi Wei, Yi Hu, and Hao Wang. 2020. Deep Multi-Interest Network for Click-through Rate Prediction. In *CIKM*. ACM, 2265–2268.

[48] Yueqi Xie, Peilin Zhou, and Sunghun Kim. 2022. Decoupled Side Information Fusion for Sequential Recommendation. In *SIGIR*. 1611–1621.

[49] Ji Yang, Xinyang Yi, Derek Zhiyuan Cheng, Lichan Hong, Yang Li, Simon Xiaoming Wang, Taibai Xu, and Ed H Chi. 2020. Mixed negative sampling for learning two-tower neural networks in recommendations. In *Companion Proceedings of the Web Conference 2020*. 441–447.

[50] Junliang Yu, Hongzhi Yin, Min Gao, Xin Xia, Xiangliang Zhang, and Nguyen Quoc Viet Hung. 2021. Socially-aware self-supervised tri-training for recommendation. In *SIGKDD*. 2084–2092.

[51] Fajie Yuan, Joemon M Jose, Guibing Guo, Long Chen, Haitao Yu, and Rami S Alkhawaldeh. 2016. Joint geo-spatial preference and pairwise ranking for point-of-interest recommendation. In *ICTAI*. IEEE, 46–53.

[52] Peiyan Zhang, Jiayan Guo, Chaozhuo Li, Yueqi Xie, Jae Boum Kim, Yan Zhang, Xing Xie, Haohan Wang, and Sunghun Kim. 2023. Efficiently Leveraging Multi-Level User Intent for Session-Based Recommendation via Atten-Mixer Network. In *WSDM* (Singapore, Singapore) *(WSDM '23)*. 168–176.

[53] Shengyu Zhang, Lingxiao Yang, Dong Yao, Yujie Lu, Fuli Feng, Zhou Zhao, Tat-seng Chua, and Fei Wu. 2022. Re4: Learning to Re-contrast, Re-attend, Re-construct for Multi-interest Recommendation. In *WWW*. 2216–2226.

[54] Weinan Zhang, Tianqi Chen, Jun Wang, and Yong Yu. 2013. Optimizing top-n collaborative filtering via dynamic negative item sampling. In *SIGIR*. 785–788.

[55] Tong Zhao, Julian McAuley, and Irwin King. 2014. Leveraging social connections to improve personalized ranking for collaborative filtering. In *CIKM*. 261–270.

[56] Yu Zheng, Chen Gao, Liang Chen, Depeng Jin, and Yong Li. DGCN: Diversified Recommendation with Graph Convolutional Networks. In *WWW*. 401–412.

[57] Peilin Zhou, Jingqi Gao, Yueqi Xie, Qichen Ye, Yining Hua, and Sunghun Kim. 2022. Equivariant Contrastive Learning for Sequential Recommendation. *arXiv preprint arXiv:2211.05290* (2022).

[58] Han Zhu, Daqing Chang, Ziru Xu, Pengye Zhang, Xiang Li, Jie He, Han Li, Jian Xu, and Kun Gai. 2019. Joint Optimization of Tree-based Index and Deep Model for Recommender Systems. In *NeurIPS*. 3973–3982.

[59] Han Zhu, Xiang Li, Pengye Zhang, Guozheng Li, Jie He, Han Li, and Kun Gai. 2018. Learning Tree-based Deep Model for Recommender Systems. In *KDD*. ACM, 1079–1088.