Unlocking the Hidden Treasures: Enhancing Recommendations with Unlabeled Data

Yuhan Zhao
Harbin Engineering University
Hong Kong Baptist University
Harbin, China
asa9ao@hrbeu.edu.cn

Rui Chen*
Harbin Engineering University
Harbin, China
ruichen@hrbeu.edu.cn

Qilong Han*
Harbin Engineering University
Harbin, China
hanqilong@hrbeu.edu.cn

Hongtao Song
Harbin Engineering University
Harbin, China
songhongtao@hrbeu.edu.cn

Li Chen
Hong Kong Baptist University
Hong Kong, China
lichen@comp.hkbu.edu.hk

ABSTRACT

Collaborative filtering (CF) stands as a cornerstone in recommender systems, yet effectively leveraging the massive unlabeled data presents a significant challenge. Current research focuses on addressing the challenge of unlabeled data by extracting a subset that closely approximates negative samples. Regrettably, the remaining data are overlooked, failing to fully integrate this valuable information into the construction of user preferences. To address this gap, we introduce a novel positive-neutral-negative (PNN) learning paradigm. PNN introduces a neutral class, encompassing intricate items that are challenging to categorize directly as positive or negative samples. By training a model based on this triple-wise partial ranking, PNN offers a promising solution to learning complex user preferences. Through theoretical analysis, we connect PNN to one-way partial AUC (OPAUC) to validate its efficacy. Implementing the PNN paradigm is, however, technically challenging because: (1) it is difficult to classify unlabeled data into neutral or negative in the absence of supervised signals; (2) there does not exist any loss function that can handle set-level triple-wise ranking relationships. To address these challenges, we propose a semi-supervised learning method coupled with a user-aware attention model for knowledge acquisition and classification refinement. Additionally, a novel loss function with a two-step centroid ranking approach enables handling set-level rankings. Extensive experiments on four real-world datasets demonstrate that, when combined with PNN, a wide range of representative CF models can consistently and significantly boost their performance. Even with a simple matrix factorization, PNN can achieve comparable performance to sophisticated graph neutral networks. Our code is publicly available at https://github.com/Asa9aoTK/PNN-RecBole.

CCS CONCEPTS

• **Information systems → Recommender systems**;

KEYWORDS

Collaborative filtering, loss function, positive-neutral-negative learning

ACM Reference Format:
Yuhan Zhao, Rui Chen, Qilong Han, Hongtao Song, and Li Chen. 2024. Unlocking the Hidden Treasures: Enhancing Recommendations with Unlabeled Data. In *18th ACM Conference on Recommender Systems (RecSys '24), October 14–18, 2024, Bari, Italy*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3640457.3688149

1 INTRODUCTION

Collaborative filtering(CF), as the most fundamental technique in recommender systems [12, 15, 22], aims to capture users' preferences and make top-K recommendations based on their historical interactions [8, 10, 29]. In real-world scenarios, these interactions are typically in the form of implicit feedback (e.g., clicks or purchases), rather than explicit ratings [28]. In these scenarios, user interactions act as positive samples, reflecting user preferences. However, the numerous unobserved items, lack explicit user feedback and remain unlabeled, thereby hindering effective utilization.

Most existing works [2, 3, 5, 18, 36] make an intuitive assumption that unlabeled data can directly provide negative signals. Yet, recent works reveal a nuanced reality: *unlabeled data and negative samples harbor an inevitable disparity.* For instance, unlabeled data may contain items that users potentially like or items with user uncertainty. Acknowledging this phenomenon, current research endeavors to address the challenge of unlabeled data by attempting to extract a subset closely approximating negative samples, thereby bridging this gap. Techniques like negative sampling [4, 6, 39], data filtering [15, 28], and weight adjustment [14] are employed for this purpose. Regrettably, under this approach, the remaining data are overlooked, failing to fully integrate this valuable information into the construction of user preferences. This limitation hampers these methods from fully harnessing the potential of unlabeled data.

In this paper, we pose a fundamental question: *Is it feasible to fully unearth and leverage the information within massive unlabeled data?* An intuitive approach might be to initially follow the existing methods by filtering out some negative samples and treating

[*] Corresponding authors.

RecSys '24, October 14–18, 2024, Bari, Italy

ACM ISBN 979-8-4007-0505-2/24/10
https://doi.org/10.1145/3640457.3688149

the remaining items as positive instances. However, such an approach is ill-suited for recommender systems. Recommendations fundamentally deal with ranking problems rather than binary classification. We cannot guarantee that the remaining items exclusively belong to the positive class. Moreover, the challenge intensifies as hard negative samples are difficult to distinguish from positive samples. Misclassifying hard negative samples as positive instances can severely compromise effectiveness.

Our answer is a novel generic positive-neutral-negative (PNN) learning paradigm. Given user $u \in U$, different from categorizing all items into either positive Σ^u_{pos} or negative Σ^u_{neg}, our insight is to introduce a third "neutral" class Σ^u_{neu}, which encompasses a plethora of intricate items that are challenging to directly categorize as either positive or negative samples. In what follows, we slightly abuse the notation to let Σ^u_* also denote the set of items with class label $*$. With this new class, our goal is to train a model based on partial rankings $\Sigma^u_{pos} >_u \Sigma^u_{neu} >_u \Sigma^u_{neg}$(i.e., u prefers items Σ^u_{pos} over Σ^u_{neu} over Σ^u_{neg}). This way allows complex and challenging unlabeled data to intuitively participate in user preference construction. Furthermore, we establish a theoretical connection between PNN and one-way partial AUC (OPAUC) to validate PNN's efficacy in recommendation performance.

While the conceptual framework of the PNN paradigm effectively addresses the aforementioned issue, providing a concrete implementation is technically challenging for at least two reasons:

- **Sparse supervised signals.** Reliable signals primarily stem from user interactions, which tend to be sparse compared to the pool of unlabeled data. This scarcity impedes the effective identification of neutral and negative items within massive unlabeled data.
- **Constrained positive-negative loss functions.** Traditional loss functions categorize all items into either positive or negative classes, failing to accommodate the nuanced nature of neutral items. Even if these latent neutral items are unearthed, they are inevitably pigeonholed into positive or negative classifications.

To address the former challenge, inspired by pseudo-labelling techniques [19, 30], we design a novel semi-supervised learning method to leverage the BPR loss to acquire sufficient prior knowledge. This knowledge enables us to discern intricate signals concealed within unlabeled data. Then, we propose a user-aware attention model to indicate the readiness to classify unlabeled data. This model uses classification performance on observed items to decide how to change semi-supervised learning to our loss function. For the second problem, we propose a novel loss function that deals with set-level ranking relationships involving three classes. Though preference ranking may not satisfy the transitive property [1, 16, 24], theoretically we can still *losslessly* decompose a triple-wise ranking into three pairwise rankings. Since $\Sigma^u_{pos} >_u \Sigma^u_{neg}$ can be learned from the BPR loss during the knowledge acquisition phase, we introduce a novel loss function to tackle $\Sigma^u_{pos} >_u \Sigma^u_{neu}$ and $\Sigma^u_{neu} >_u \Sigma^u_{neg}$. To support set-level rankings, we propose an innovative two-step centroid ranking approach. In the first step, we advocate representing each set of items within the three classes by its centroid, thereby transforming intricate set-level relationships into more manageable item-level counterparts. Subsequently, we propose a clamp mechanism meticulously to maintain adaptive margins between distinct classes. This mechanism guarantees that every item in a class ranks higher or lower than any item in another class. We summarize our main contributions as follows:

- We introduce the novel PNN learning paradigm, a pioneering approach to fully exploit the wealth of information within massive unlabeled data in CF. By incorporating a third neutral class and leveraging set-level ranking relationships, PNN addresses the complexity of unlabeled data. Furthermore, through mathematical analysis, we establish the relationship between PNN and OPAUC, demonstrating PNN's ability to optimize various indicators of the recommendation system.
- We propose a concrete implementation of the PNN learning paradigm which can be seamlessly integrated with multiple mainstream CF models. It features a semi-supervised learning method with a user-aware attention model to reliably classify unlabeled data and a two-step centroid ranking approach to accommodate set-level rankings.
- We perform extensive experiments on four public datasets to demonstrate that, when combined with PNN, a wide variety of mainstream CF models can consistently and substantially boost their performance, confirming the value of PNN.

2 PNN MEETS OPAUC

In this section, we embark on a review of traditional methods and shed light on the PNN paradigm, leveraging the OPAUC metric.

The goal of CF is to learn the model's parameters Θ to recommend the top-ranked items. In the implicit feedback setting, for a user u, $\mathcal{I}^+_u$ denotes the positive class constituting by observed items while the remainder, $\mathcal{I}^{un}_u = \mathcal{I} \setminus \mathcal{I}^+_u$, represents unlabeled data. We demarcate true negative items as $\mathcal{I}^-_u$. Traditional loss functions can be formulated as follows:

$$\min_{\Theta} \sum_{u \in U} \sum_{i \in \mathcal{I}^+_u} E_{j \sim P_{ns}(j|\Theta)} [\mathcal{L}(r(u, i \mid \Theta), r(u, j \mid \Theta))], \tag{1}$$

where $P_{ns}(j \mid \Theta)$ denotes the negative sampling probability that a negative item is drawn. Departing from conventional paradigms, we introduce the PNN learning framework, which stratifies unlabeled data into neutral and negative classes, employing set-level rankings for parameter optimization:

$$\max_{\Theta} \sum_{u \in U} \Sigma^u_{pos} >_u \Sigma^u_{neu} >_u \Sigma^u_{neg}. \tag{2}$$

As a novel paradigm, our objective is to theoretically dissect the impact of PNN on recommendation performance. Recent research reveals the correlation between OPAUC and top-K evaluation metrics [20], motivating our investigation into PNN efficacy via this metric. OPAUC is expressed as:

$$\widehat{OPAUC}(\Theta, \gamma, \delta) = \frac{1}{|U|} \sum_{u \in U} \int_{\gamma}^{\delta} TPR_{u,\theta} \left[FPR^{-1}_{u,\theta}(s) \right] \mathrm{d}s, \tag{3}$$

where TPR and FPR represent true positive rates and false positive rates, respectively. OPAUC only cares about the performance within a given false positive rate (FPR) range $[\gamma, \delta]$. In line with prior works [20], we consider the special case of OPAUC with $\gamma = 0$, yielding the non-parametric estimator:

$$\widehat{OPAUC}(\delta) = \frac{1}{|U|} \sum_{u \in U} \frac{1}{|\mathcal{I}_u^+|} \frac{1}{|\mathcal{I}_u^-|} \sum_{i \in I_u^+} \sum_{j \in S_{I_u^-}[1, |\mathcal{I}_u^-| \cdot \delta]} \mathbb{I}\left(r_{ui} > r_{uj}\right), \tag{4}$$

where $I(\cdot)$ is an indicator function and $S_{I_u^-}\,[1, n_- \cdot \delta]$ denotes the subset of the top-ranked negative items(subsequently abbreviated as $S_{I_u^-}$). However, as previously mentioned, given that unlabeled data harbors signals beyond mere negativity, a chasm emerges between $\mathcal{I}_u^-$ and $\mathcal{I}_u^{un}$. Our optimization endeavor cannot directly procure top-ranked negative items. Thus, we invoke further refinement:

$$\widehat{OPAUC}(\delta) = |K| \sum_{i \in I_u^+} \sum_{i^n \in S_{I_u^{un}}} \mathbb{I}\left(r_{ui} > r_{uj}\right) \mathbb{I}(i^n \in S_{I_u^-}). \tag{5}$$

$|K|$ denotes the constant unrelated to derivation. And for brevity, we've omitted the user. Since direct sampling from $\mathcal{I}_u^-$ is unfeasible, we initially sample from unlabeled data and subsequently utilize an indicator function to identify top-ranked negative samples. Then, for these samples, it is evident that $r_{ui^n} > r_{uj}$. Consequently, we can rewrite the equation as

$$\widehat{OPAUC}(\delta) = |K| \sum_{i \in I_u^+} \sum_{i^n \in I_u^{un}} \sum_{j \in I_u^{un}} \mathbb{I}\left(r_{ui} > r_{ui^n} > r_{uj}\right) \mathbb{I}(i^n \in S_{I_u^-}) \tag{6}$$

Obviously, by expressing Eq. 6 in set notation and introducing symbols for positive, neutral, and negative classes within PNN, we arrive at Eq. 2.

It becomes evident that neutral samples and the set-level rankings introduced by PNN can directly optimize the OPAUC metric, thereby enhancing various evaluation metrics of recommender systems. Simultaneously, through Eq. 6, it's apparent that the concrete implement PNN needs to grapple with two key challenges. First, $\mathbb{I}(i^n \in S_{I_u^-})$ implies the necessity to devise a reliable scheme to distinguish neutral samples and negative ones. Second, $\mathbb{I}\left(r_{ui} > r_{ui^n} > r_{uj}\right)$ necessitates an approach beyond conventional positive-negative learning paradigms.

3 METHODOLOGY

Driven by the aforementioned limitations, we propose a novel generic positive-neutral-negative (PNN) learning paradigm. In the following sections, we introduce a concrete implementation as a loss function $\mathcal{L}$, which allows it to be seamlessly integrated with many mainstream CF models. The implementation boils down to two key components: a classifier that reliably distinguishes between neutral and negative items and a ranking mechanism that can effectively handle set-level triple-wise ranking relationships. The workflow is illustrated in Figure 1. Note that these steps can be implemented by different methods and thus the overall paradigm is generic.

3.1 Semi-Supervised Learning

The key challenge of the classification is a lack of supervised signals that are essential to separate neutral and negative items. Inspired by pseudo-labelling techniques [19, 30], we design a novel semi-supervised learning method that adequately leverages the knowledge acquired from the BPR loss as the decision foundation. Given user $u \in U$, we formulate the loss function as follows:

$$\mathcal{L}_u = (1 - \lambda)\mathcal{L}_{\text{BPR}} + \lambda\mathcal{L}_{\text{PNN}}, \tag{7}$$

where $\mathcal{L}_{\text{BPR}}$ represents the BPR loss, $\mathcal{L}_{\text{PNN}}$ denotes the PNN loss that factors in the influence of both the classification and ranking tasks, and λ is a key parameter that allows us to adaptively adjust the weights between the BPR loss and the PNN loss.

At the beginning of the training process, we mainly rely on the BPR loss to train the model to acquire prior knowledge about user preferences via

$$\mathcal{L}_{\text{BPR}} = -\ln[\sigma(s(\mathbf{e}_u, \mathbf{e}_i) - s(\mathbf{e}_u, \mathbf{e}_j))], \tag{8}$$

where $s(\cdot, \cdot)$ denotes the score function (e.g., inner product) used to calculate similarities, $\sigma(\cdot)$ denotes the sigmoid function, and $\mathbf{e}_u$, $\mathbf{e}_i$, and $\mathbf{e}_j$ are user/item embeddings. Here i is a positive sample with which the user interacted, and j is a negative sample selected from unlabeled data.

We use dynamic negative sampling (DNS) [36], a hard negative sampling method, to select an item j with probability proportional to $p_j \propto s(\mathbf{e}_u, \mathbf{e}_j)$. The intuition of DNS is to select negative items that are "closer" to the user. Such negative samples can yield greater gradients [4, 17], thereby accelerating the training process. This choice is important because we expect the model to quickly acquire valuable knowledge from the BPR loss and then start to optimize $\mathcal{L}_{\text{PNN}}$. However, this strategy needs to calculate the scores of all unlabeled data to assign probabilities. The time complexity is prohibitive. To balance efficiency and effectiveness, we propose to select the negative sample by batch style via

$$\mathbf{e}_j = \underset{n \in batch \setminus i}{\arg\max}\, s(\mathbf{e}_u, \mathbf{e}_n). \tag{9}$$

Diverging from the approach of selecting negative samples from the entire item set, we restrict the scope to a single batch. Within this batch, we directly identify the highest-score sample as the negative sample, excluding the current positive sample. The parameter λ is designed to indicate the readiness to make use of the PNN loss. A conventional method is to set λ as a hyperparameter or learn a function of the number of completed training epochs [30]. However, such methods are not aligned with our objective. More specifically, once the current model has acquired sufficient knowledge for the classification task, we want to reduce the weight of the BPR loss and start to focus on optimizing the PNN loss. Conversely, when the classification task is arduous, we may want to increase the weight of the BPR loss.

Similarly, how to design λ is also a challenging task due to a lack of supervised signals. Inspired by previous works [6, 15], we propose a user-aware attention model to indirectly assess the classification performance based on the user's positive items sets Σ_{pos}^u. Intuitively, if the model can correctly identify these items as positive samples, it suggests that the model has adequately learned user preferences from the BPR loss. Following this intuition, we devise λ as follows:

$$\lambda = \sigma[\mathbf{W}_1(\sum_{i \in \Sigma_{pos}^u} \alpha_i^{attr} \mathbf{e}_i)], \tag{10}$$

where $\mathbf{W}_1 \in \mathbb{R}_+^{1 \times d}$ is a trainable matrix, and the sigmoid function $\sigma(\cdot)$ is used to map values to $(0, 1)$. Here α_i^{attr} represents the

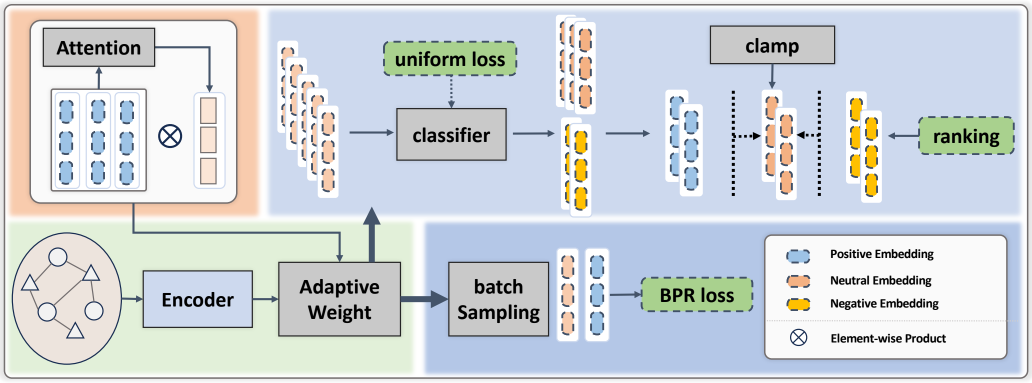


Figure 1: The overall workflow of our PNN learning paradigm.

aggregation weight, which can be computed via

$$\alpha_i^{attr} = \frac{\exp\left(\beta_i^{attr}\right)}{\sum_{j\in\Sigma_{pos}^u}\exp\left(\beta_j^{attr}\right)}, \qquad (11)$$
$$\beta_i^{attr} = \mathbf{e}_u^T \tanh(\mathbf{W}_2\mathbf{e}_i + \mathbf{b}),$$

where $\mathbf{W}_2 \in \mathbb{R}_+^{d\times d}$ and $\mathbf{b} \in \mathbb{R}^{d\times 1}$ are trainable parameters, and the user embedding $\mathbf{e}_u$ is used as a query vector. In this way, the attention mechanism can serve as an indirect evaluation metric for classification performance.

When our model possesses sufficient knowledge, the positive items should exhibit a high degree of similarity to the user's preferences. As a result, α_i^{attr} will receive a higher value. Consequently, the weight λ will also increase, prioritizing the significance of $\mathcal{L}_{\text{PNN}}$ in the overall loss function.

3.2 PNN Loss Function

After acquiring sufficient knowledge from the BPR loss to warm-start the PNN loss, we can now focus on classifying unlabeled data into either neutral or negative and effectively handling set-level triple-wise ranking relationships.

3.2.1 Classification Task. Following the convention of implicit CF, for a given user $u \in U$, the positive class Σ_{pos}^u consists of all her interacted items. Let $|\Sigma_{pos}^u| = N_u$. The classification between neutral and negative is achieved by the PNN loss function.

Intuitively, the items with the lowest similarities to u are more likely to be truly negative. Driven by this intuition, we take a greedy approach to iteratively select N_u negative items. Since the knowledge gained from the semi-supervised learning process could still be limited, our design principle is to put only truly reliable negative items into Σ_{neg}^u. For this reason, we limit the size of Σ_{neg}^u to N_u to reduce the risk of introducing false negative samples. Note that N_u is much smaller than the number of unlabeled data. Formally, in each iteration, we update Σ_{neg}^u as follows:

$$i^{neg} = \mathop{\arg\min}_{i\in\mathcal{E}} s(\mathbf{e}_u, \mathbf{e}_i),$$
$$\mathcal{E}^u = \mathcal{E}^u - i^{neg}, \qquad (12)$$
$$\Sigma_{neg}^u = \Sigma_{neg}^u \cup \{i^{neg}\},$$

where $\mathcal{E}$ is initialized to the set of u's unlabeled items $\mathcal{E}^u$, and Σ_{neg}^u is initialized to $\varnothing$. Consequently, the neutral class consists of all remaining unlabeled data, namely $\mathcal{E}^u - \Sigma_{neg}^u$.

Recalling that the PNN loss function is optimized on top of the knowledge acquired from the BPR loss. Since the BPR loss is designed to set positive items apart from unlabeled data , it may lead to clusters of unlabeled data with similar scores, which is detrimental to our classification goal. Considering the scenario where both i_1 and i_2 receive identical scores of 2 points, in such a case, discerning which is more likely to represent a negative sample becomes challenging. To address this problem, it is natural to impose a constraint that unlabeled data are sufficiently distant from each other. Specifically, we introduce the following loss factor:

$$\mathcal{L}_{\text{uniform}} = \log \mathop{\mathbb{E}}_{i,i'\in\mathcal{E}^u} e^{-2\|\mathbf{e}_i - \mathbf{e}_{i'}\|^2}. \qquad (13)$$

By utilizing the Euclidean distance, we encourage unlabeled data to push each other further apart, leading to a more uniform distribution [27]. As a result, we can more reliably distinguish between neutral and negative items.

3.2.2 Ranking Task. Next, we discuss how to handle set-level triple-wise ranking relationships so that our solution can be seamlessly integrated with mainstream CF models. It is important to realize that preference ranking may not satisfy the transitive property [1, 16, 24]. Hence we need to decompose the set-level triple-wise ranking relationship $\Sigma_{pos}^u >_u \Sigma_{neu}^u >_u \Sigma_{neg}^u$ into three pairwise ranking relationships to avoid information loss: $\Sigma_{pos}^u >_u \Sigma_{neg}^u$, $\Sigma_{pos}^u >_u \Sigma_{neu}^u$, and $\Sigma_{neu}^u >_u \Sigma_{neg}^u$. While the BPR loss operates at the item level, it can still well enforce $\Sigma_{pos}^u >_u \Sigma_{neg}^u$. So the PNN loss should take into consideration the latter two pairwise rankings.

Once we have decomposed pairwise rankings, we move to address the challenge due to the set-level formulation. To this end, we propose a two-step centroid ranking approach to convert set-level relationships into item-level relationships while maintaining the desirable property of set-level rankings. In the first step, we propose to represent the set of items in each of the three classes by its centroid, namely $\mathbf{e}_{i^{neu}} = \mathop{\mathbb{E}}_{i\in\Sigma_{neu}^u} \mathbf{e}_i$, $\mathbf{e}_{i^+} = \mathop{\mathbb{E}}_{i\in\Sigma_{pos}^u} \mathbf{e}_i$, and $\mathbf{e}_{i^-} = \mathop{\mathbb{E}}_{i\in\Sigma_{neg}^u} \mathbf{e}_i$. This natural idea also gains another benefit of mitigating the impact of class imbalance: the size of Σ_{neu}^u is much larger than that of Σ_{pos}^u

or Σ^u_{neg}. Then we can capture the two pairwise rankings via

$$\begin{aligned}\mathcal{L}_{\text{rank}} = &- \ln[\sigma(s(\mathbf{e}_u, \mathbf{e}_{i^+}) - s(\mathbf{e}_u, \mathbf{e}_{i^{neu}}))] \\ &- \ln[\sigma(s(\mathbf{e}_u, \mathbf{e}_{i^{neu}}) - s(\mathbf{e}_u, \mathbf{e}_{i^-}))].\end{aligned} \tag{14}$$

Here we construct $\mathcal{L}_{\text{rank}}$ in a format similar to the BPR loss. This is because it guarantees that the magnitudes of $\mathcal{L}_{\text{BPR}}$ and $\mathcal{L}_{\text{rank}}$ remain comparable and that the learning process would not be dominated by one of them [21, 32].

Unfortunately, the above method cannot guarantee that every item in a class ranks higher than any item in another class. Take positive and neutral as examples, an ideal ranking need to satisfy $\forall i \in \Sigma^u_{pos} >_u \forall j \in \Sigma^u_{neu}$. It is not hard to understand because Eq. 14 can only enforce the ranking relationships on the centroids. Some of the classes may overlap with each other. Therefore, we want to maintain larger margins between different classes so that the set-level property can be respected. Intuitively, we can set a fixed margin between classes as usual to avoid overlap. However, we have no prior knowledge to determine the positive-neutral-negative reasonable margin, artificially setting the margin will introduce inductive bias. Inspired by previous works [7, 9, 33, 35], in the second step, we propose a novel clamp mechanism to adaptively generate the margin. Our insight is to utilize two clamp embeddings to tightly constrain the neutral classes in the embedding space. Since the neutral class occupies a position between the positive and negative classes, clamping the neutral class creates a distinct margin with the other two classes. As a result, this approach enables better separation and avoids overlap among the different classes. Figure 2 vividly shows this process.

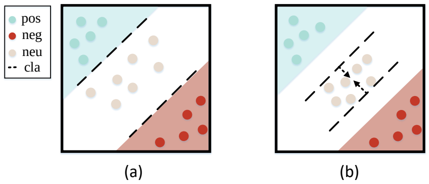


Figure 2: The overall process of the clamp mechanism. pos, neu, and neg, respectively represent the positive, neutral, and negative class, and cla denotes clamp embedding.

Our clamp embedding requires the presence of one embedding between positive and neutral, as well as another embedding between neutral and negative. For the sake of brevity, we will focus solely on the positive-related embedding. First, we generate noise $\Delta_{pos} \in \mathbb{R}^d$ as follows. We adopt a uniform distribution on the interval $[0, 0.1]$ to decide the magnitude. A uniform distribution introduces a certain amount of randomness, which is also beneficial to improve the model's robustness. We fix the direction of Δ_{pos} towards the positive. Take the direction towards the positive class as an example. We can generate the clamp embedding as

$$\begin{gathered}\Delta_{pos} = \bar{\Delta}_{pos} \odot \frac{(\mathbf{e}_{i^+})}{||\,(\mathbf{e}_{i^+})\,||}, \bar{\Delta}_{pos} \in \mathbb{R}^d \sim U(0, 0.1), \\ \mathbf{e}^{pos}_{clamp} = \mathbf{e}_{i^{neu}} + \Delta,\end{gathered} \tag{15}$$

where $\odot$ is the element-wise product, and $f(x) = \frac{x}{||x||}$ is the sign function to determine the direction vector of the noise. $\mathbf{e}^{neg}_{clamp}$ can be obtained in a similar way.

Table 1: The statistics of the datasets used in the experiments.

Dataset	Users	Items	Interactions	Sparsity
ML-1M	6,041	3,707	1,000,209	95.53%
Yelp	31,669	38,049	1,561,406	99.87%
Gowalla	29,859	40,982	1,027,370	99.92%
Foursquare	1,084	38,334	91,024	99.78%

Note that we want to enlarge the margin by tightly constraining the neutral classes in the embedding space. For this reason, we minimize the distance between the two clamp embeddings, compressing the neutral class similar to how a clamp operates. which leads to the following:

$$\mathcal{L}_{\text{constrain}} = \mathbb{E}\|(\mathbf{e}^{pos}_{clamp} - \mathbf{e}^{neg}_{clamp})\|^2. \tag{16}$$

It is worth noting that Eq. 16 can be simplified, providing an alternative interpretation of the clamp mechanism. However, to ensure readers' understanding, we retain the original form. Now we are ready to present the PNN loss $\mathcal{L}_{\text{PNN}}$, which accounts for both classifying unlabeled data and tackling set-level triple-wise rankings:

$$\mathcal{L}_{\text{PNN}} = \alpha \mathcal{L}_{\text{constrain}} + \beta \mathcal{L}_{\text{uniform}} + \mathcal{L}_{\text{rank}}, \tag{17}$$

where α and β are hyperparameters to control the relative weights of $\mathcal{L}_{\text{constrain}}$ and $\mathcal{L}_{\text{uniform}}$, respectively.

3.3 Mini-Batch Training

In this section, we explain how PNN can be easily integrated with different mainstream CF models. Ideally, we want PNN to use all observed and unobserved items of a given user in each training epoch. In this case, the final loss function $\mathcal{L}$ can be expressed as $\mathcal{L} = \sum_{u \in U} \mathcal{L}_u$. However, for efficiency reasons, mainstream CF models generally adopt mini-batch training, where a mini-batch consists of user-item interactions in the form of (u, i). Thus, accessing all user interaction data in a mini-batch is impossible.

To seamlessly integrate PNN, we propose to relax the ranking as $i >_u \Sigma^{\widetilde{u}}_{neu} >_u \Sigma^{\widetilde{u}}_{neg}$, where $\Sigma^{\widetilde{u}}_{neu}$ and $\Sigma^{\widetilde{u}}_{neg}$ denote the neutral class and negative class that can be built in mini-batch training, respectively. Note that $|\Sigma^{\widetilde{u}}_{neu}| \ll |\Sigma^u_{neu}|$ and $|\Sigma^{\widetilde{u}}_{neg}| \ll |\Sigma^u_{neg}|$. If a model can provide unlabeled data, we can directly utilize them to construct $\Sigma^{\widetilde{u}}_{neu}$ and $\Sigma^{\widetilde{u}}_{neg}$. However, when such data is not available, we design a scheme with reference to contrastive learning [31, 35]. We utilize items that were interacted with by other users, but not user u, in a mini-batch as unobserved items to form neutral and negative classes. Thus we have $\mathcal{L} = \sum_{(u,i) \in D} \mathcal{L}_u$. Similar to contrastive learning, during training, when i takes all the values in Σ^u_{pos}, we can get the following approximation:

$$\prod_{i \in \Sigma^u_{pos}} p(i >_u \Sigma^{\widetilde{u}}_{neu} >_u \Sigma^{\widetilde{u}}_{neg}) \approx p(\Sigma^u_{pos} >_u \Sigma^u_{neu} >_u \Sigma^u_{neg}). \tag{18}$$

This approximation enables PNN to be seamlessly applied to almost all mainstream CF models. Specifically, we just need to replace the original pairwise ranking loss function with $\mathcal{L} = \sum_{(u,i) \in D} \mathcal{L}_u$.

Table 2: Experimental results of different CF models with (w) or without (w/o) the PNN loss function. The best results are boldfaced, and the improvement is the average of all metrics.

Datasets	Metric	BPR-MF		LightGCN		NGCF		SGL	
		w/o	w	w/o	w	w/o	w	w/o	w
ML-1M	Recall@10	0.1635	**0.1840**	0.1730	**0.1971**	0.1614	**0.1880**	0.1769	**0.1942**
	Recall@20	0.2502	**0.2742**	0.2647	**0.2909**	0.2475	**0.2839**	0.2665	**0.2887**
	Hit@10	0.7419	**0.7753**	0.7603	**0.7997**	0.7407	**0.7803**	0.7699	**0.7929**
	Hit@20	0.8374	**0.8613**	0.8546	**0.8811**	0.8359	**0.8694**	0.8591	**0.8752**
	NDCG@10	0.2554	**0.2835**	0.2682	**0.2895**	0.2541	**0.2898**	0.2682	**0.2909**
	NDCG@20	0.2619	**0.2869**	0.2749	**0.2951**	0.2595	**0.2951**	0.2755	**0.2959**
	Improvement		**8.34%**		**7.90%**		**11.39%**		**6.47%**
Gowalla	Recall@10	0.0938	**0.1236**	0.1280	**0.1357**	0.1011	**0.1215**	0.1331	**0.1437**
	Recall@20	0.1389	**0.1787**	0.1849	**0.1972**	0.1490	**0.1813**	0.1897	**0.2085**
	Hit@10	0.1989	**0.2534**	0.2553	**0.2712**	0.2110	**0.2491**	0.2667	**0.2846**
	Hit@20	0.2763	**0.3366**	0.3432	**0.3635**	0.2900	**0.3414**	0.3519	**0.3797**
	NDCG@10	0.0674	**0.0896**	0.0917	**0.0981**	0.0722	**0.0875**	0.0966	**0.1044**
	NDCG@20	0.0804	**0.1052**	0.1079	**0.1157**	0.0859	**0.1046**	0.1128	**0.1229**
	Improvement		**28.90%**		**6.50%**		**20.09%**		**8.25%**
Yelp	Recall@10	0.0427	**0.0547**	0.0543	**0.0668**	0.0439	**0.0587**	0.0612	**0.0674**
	Recall@20	0.0717	**0.0873**	0.0884	**0.1079**	0.0738	**0.0946**	0.0990	**0.1084**
	Hit@10	0.1537	**0.1917**	0.1896	**0.2262**	0.1593	**0.2047**	0.2101	**0.2275**
	Hit@20	0.2398	**0.2819**	0.2847	**0.3317**	0.2460	**0.3014**	0.3098	**0.3334**
	NDCG@10	0.0334	**0.0435**	0.0432	**0.0539**	0.0345	**0.0469**	0.0487	**0.0540**
	NDCG@20	0.0432	**0.0543**	0.0546	**0.0676**	0.0447	**0.0588**	0.0614	**0.0677**
	Improvement		**24.68%**		**21.58%**		**30.07%**		**9.44%**
Foursquare	Recall@10	0.0262	**0.0332**	0.0369	**0.0376**	0.0325	**0.0356**	0.0358	**0.0366**
	Recall@20	0.0381	**0.0472**	0.0537	**0.0554**	0.0474	**0.0542**	0.0524	**0.0525**
	Hit@10	0.1717	**0.2041**	0.2364	**0.2364**	0.2151	**0.2225**	0.2281	**0.2299**
	Hit@20	0.2336	**0.2770**	0.3158	**0.3250**	0.2973	**0.3186**	**0.3213**	0.3130
	NDCG@10	0.0283	**0.0352**	0.0381	**0.0398**	0.0345	**0.0392**	0.0391	**0.0392**
	NDCG@20	0.0328	**0.0405**	0.0443	**0.0466**	0.0403	**0.0469**	**0.0455**	0.0449
	Improvement		**22.65%**		**2.93%**		**10.74%**		**-**

4 EXPERIMENTS

In this section, we conduct comprehensive experiments to answer the following key research questions:

- **RQ1:** How does integrating PNN help different mainstream CF models boost their performance? How does the performance of PNN compare to state-of-the-art baselines?
- **RQ2:** How does the neutral class affect model performance?
- **RQ3:** How do different components of PNN affect model performance?
- **RQ4:** How do different parameter settings affect model performance?
- **RQ5:** Is PNN efficient for practical use?

4.1 Experimental Setup

4.1.1 Datasets. We conduct experiments on four datasets widely used in the literature to evaluate the performance of PNN: (1) **MovieLens**: It contains user ratings on movies. We use 1M versions and treat all rating movies as interactive items. (2) **Yelp**: It contains user reviews of restaurants and bars. We use the transaction records after Jan. 1st, 2018. (3) **Gowalla**: This is a check-in dataset obtained from Gowalla, where users share their locations by checking-in. (4) **Foursquare**: This dataset contains check-ins in NYC and Tokyo collected for about 10 months. We split the historical interactions into training, validation, and test sets in a ratio of 8:1:1. Table 1 summarizes the statistics of the datasets.

4.1.2 Implementation Details. The experiments are conducted on the NVIDIA Ampere A100-40G GPU using PyTorch. To ensure reproducibility, we implement all methods by using the RecBole v1.1.1 framework [37]. We maintain a fixed size of 64 for the embeddings. The optimization of parameters is carried out using Adam with a default learning rate of 0.001 and a default mini-batch size of 2048. The L_2 regularization coefficient is set to 10^{-4}. We evaluate the performance of PNN by varying the value α within the range of $[0, 1]$ and the value β within the range of $[0, 1]$. We meticulously tune all hyper-parameters on the validation datasets and report the best performance for all basic models.

Table 3: The performance comparison between PNN and other state-of-the-art methods.

Dataset	Methods	HR@10	HR@20	Recall@10	Recall@20	NDCG@10	NDCG@20	MRR@10
ML-1M	**SRNS**	0.7611	0.8533	0.1746	0.2618	0.2639	0.2687	0.4661
	MixGCF	0.7601	0.8483	0.1739	0.2604	0.2657	0.2702	0.4680
	UIB	0.7346	0.8318	0.1559	0.2406	0.2453	0.2520	0.4279
	SimpleX	0.7586	0.8518	0.1709	0.2573	0.2626	0.2675	0.4607
	ANS	0.7675	0.8583	0.1759	0.2625	0.2624	0.2683	0.4640
	PNN	**0.7753**	**0.8613**	**0.1840**	**0.2742**	**0.2835**	**0.2869**	**0.4837**
Gowalla	**SRNS**	0.2295	0.3177	0.1171	0.1711	0.0814	0.0972	0.1042
	MixGCF	0.2401	0.3254	0.1189	0.1728	0.0853	0.1008	0.1145
	UIB	0.2275	0.3161	0.1140	0.1695	0.0794	0.0954	0.1029
	SimpleX	0.2192	0.3082	0.1074	0.1617	0.0736	0.0892	0.0949
	ANS	0.2467	0.3317	0.1206	0.1756	0.0869	0.1027	0.1180
	PNN	**0.2534**	**0.3366**	**0.1236**	**0.1787**	**0.0896**	**0.1052**	**0.1217**
Foursquare	**SRNS**	0.1948	0.2596	0.0314	0.0418	0.0351	0.0392	0.0840
	MixGCF	0.1994	0.2752	0.0316	0.0458	0.0351	0.0405	0.0863
	UIB	0.1662	0.2299	0.0265	0.0375	0.0280	0.0322	0.0717
	SimpleX	0.1801	0.2650	0.0267	0.0421	0.0267	0.0329	0.0641
	ANS	0.1958	0.2798	0.0303	0.0461	0.0321	0.0386	0.0821
	PNN	**0.2041**	**0.2770**	**0.0332**	**0.0472**	**0.0352**	**0.0405**	**0.0876**
Yelp	**SRNS**	0.1834	0.2779	0.0520	0.0862	0.0406	0.0522	0.0690
	MixGCF	0.1891	0.2781	0.0538	0.0871	0.0424	0.0536	0.4680
	UIB	0.1592	0.2504	0.0445	0.0756	0.0350	0.0455	0.0598
	SimpleX	0.1850	0.2780	0.0519	0.0861	0.0410	0.0524	0.0706
	ANS	0.1844	0.2777	0.0524	0.0862	0.0414	0.0527	0.0708
	PNN	**0.1917**	**0.2819**	**0.0547**	**0.0873**	**0.0435**	**0.0543**	**0.0739**

4.1.3 Evaluation Protocols. We evaluate performance employing standard metrics for top-K recommendations, encompassing Normalized Discounted Cumulative Gain (NDCG@K), Hit Rate (HR@K), Recall (Recall@K), and Mean Reciprocal Rank (MRR). Our findings are based on the average results obtained from five independent runs and we conduct statistical significance analysis by calculating p-values against the best-performing baseline.

4.1.4 Baseline. To validate the effectiveness of our solution, we integrate PNN with a wide range of mainstream CF models.

- **MF-BPR** [18], which is the most classic matrix factorization.
- **NGCF** [29], which employs a graph neural network to leverage high-order information.
- **LightGCN** [8], which eliminates the redundant components from NGCF to improve performance.
- **SGL** [31], which incorporates contrastive learning to improve the accuracy and robustness.

Furthermore, we compare PNN with several representative state-of-the-art methods. Following the conventional setting [4, 18, 26], we consider BPR-MF as the basic model of the following methods.

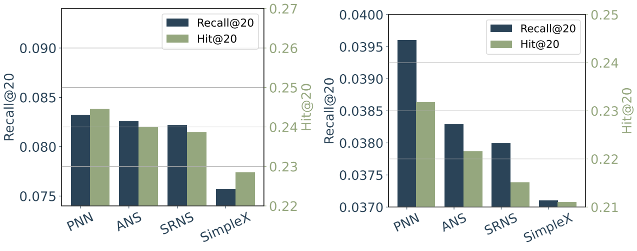
- **SimpleX** [15], which extracts a large volume of high-information samples from unlabeled data to serve as negative samples and uses cosine similarity for ranking.
- **UIB** [40], which introduces a personalized boundary to penalize samples crossing the threshold.
- **SRNS** [6], which selects hard samples from unlabeled data to serve as negative samples while incorporating a variance mechanism to filter out false negative samples.
- **MixGCF** [11], which leverages information from positive samples and graph neighborhood samples to synthesize negative samples based on unlabeled data.
- **ANS** [38], which proposes to generate negative samples based on fine-granular factors within unlabeled data.

4.2 RQ1: Overall Performance Comparison

We report the main experimental results in Table 2 and Table 3, where the reported improvements are the average of all metrics and are significant over the (best) baseline(s) under a two-sided t-test with $p < 0.05$.

- From Table 2, it can be observed that, with the help of PNN, all base models show significant performance improvements in almost all cases. The performance improvement of NGCF on Yelp is as high as 30.07%. Surprisingly, the experimental results indicate that, even with a simple BPR-MF model, PNN can achieve comparable performance to sophisticated GNN models. The results confirm the superiority and applicability of our proposed PNN solution.
- As shown in Table 3, while traditional methods achieve reasonably good performance, PNN consistently delivers the best performance across all datasets and all metrics. The encouraging results suggest exploring a reasonable way to unearth and leverage the information within unlabeled data.

- It is interesting to observe that SRNS achieves better performance in some datasets. This validates our assertion that unlabeled data and negative samples exhibit an inevitable disparity. While SRNS improves performance by filtering items that users potentially like, unfortunately, due to its limited utilization of this filtered samples, its performance still lags behind PNN.



(a) **Yelp** *(b)* **Foursquare**

Figure 3: The results of the study on neutral class.

4.3 RQ2: Study on Neutral Class

In this section, we will explore the role of neutral samples in PNN. Intuitively, we can label items that users potentially like or items with user uncertainty in the dataset and then evaluate the detection rate of false negative samples. However, a high detection rate does not directly reflect the distinction between our method and traditional approaches (such as SRNS, which can also detect false negative samples). Inspired by [39], we randomly select 10% of interactions from Yelp and Foursquare datasets as negative to simulate the false negative samples and use the constructed datasets to evaluate different methods. Following the principles of PNN, these samples should be categorized as the neutral class and participate in training processes. As can be observed in Figure 3, SimpleX performs worst because it relies on many negative samples, making it more vulnerable to noise. In contrast, PNN can effectively resist the impact of false negative noise. Despite SRNS's capability to filter false negative samples, its failure to effectively utilize this portion of the signal still results in suboptimal performance, reaffirming our motivation to fully leverage unlabeled data.

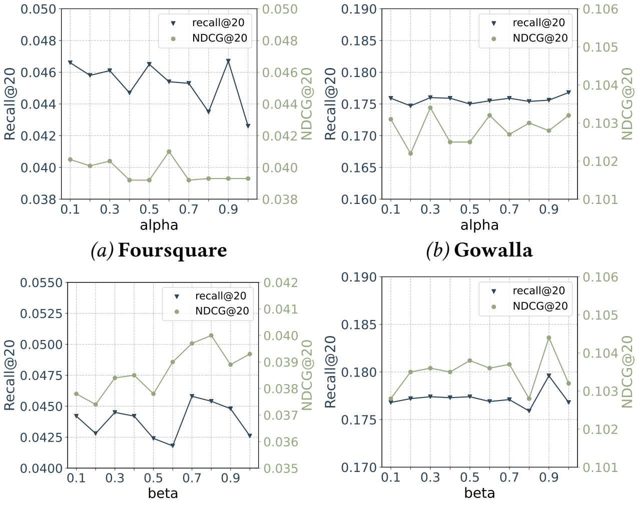
4.4 RQ3: Ablation Study

We analyze the effectiveness of different components via the following variants: (1) PNN without $\mathcal{L}_{\text{constrain}}$ (PNN w/o constrain), (2) PNN without $\mathcal{L}_{\text{uniform}}$ (PNN w/o uniform), (3) PNN without $\mathcal{L}_{\text{rank}}$ (PNN w/o rank), (4) PNN without adaptive weight λ (PNN w/o ada), and (5) PNN without semi-supervised learning (PNN w/o sem). The results are presented in Table 4. They suggest that all components positively contribute to model performance. We can observe that $\mathcal{L}_{\text{rank}}$ (corresponding to PNN w/o rank) and semi-supervised learning (corresponding to PNN w/o sem) significantly impact the performance. The former can be attributed to the fact that the ranking task has a direct impact on the top-K recommendation task. Without $\mathcal{L}_{\text{rank}}$, we cannot leverage the triple-wise ranking relationships. The latter confirms the challenge of the classification task in the absence of supervised signals and the effectiveness of the proposed semi-supervised learning method.

Table 4: Ablation Study.

Dataset	**Method**	Top-20		
		Hit Ratio	Recall	NDCG
Yelp	PNN w/o constrain	0.2705	0.0454	0.0389
	PNN w/o uniform	0.2705	0.0451	0.0380
	PNN w/o rank	0.0960	0.0152	0.0115
	PNN w/o ada	0.2761	0.0463	0.0399
	PNN w/o sem	0.1967	0.0319	0.0262
	PNN	**0.2770**	**0.0472**	**0.0405**
Gowalla	PNN w/o constrain	0.3078	0.1562	0.0908
	PNN w/o uniform	0.3140	0.1651	0.0951
	PNN w/o rank	0.0406	0.0150	0.0079
	PNN w/o ada	0.3008	0.1558	0.0894
	PNN w/o sem	0.1439	0.0620	0.0361
	PNN	**0.3366**	**0.1787**	**0.1052**

4.5 RQ4: Hyperparameter Study



(a) **Foursquare** *(b)* **Gowalla**

(c) **Foursquare** *(d)* **Gowalla**

Figure 4: Impact of hyperparameters on PNN.

We conduct experiments on Foursquare and Yelp to study the impact of different values of α and β and present the results in Figure 4. Recall that α is the parameter to balance the importance of $\mathcal{L}_{\text{constrain}}$ and β to balance the importance of $\mathcal{L}_{\text{uniform}}$. An overly small α value may lead to class overlap and cannot preserve the desirable set-level ranking property; an overly large α value may create large margins that push intra-class items to cluster. Similarly, overly large or small β values could undermine model performance. The experimental results are well aligned with our analysis. But overall PNN can obtain good performance under a relatively wide range of hyperparameters.

4.6 RQ5: Efficiency Analysis

Following the previous works [15, 25, 38], we compare the efficiency of PNN with other representative methods (BPR-MF, SimpleX, and ANS). Table 5 presents the average training time per epoch and the

Table 5: The efficiency results of different methods.

Dataset	Method	Training Time	# of Epoch
Yelp	BPR-MF	2.40s	82
	SimpleX	431.92s	36
	ANS	258.26s	16
	PNN	250.46s	15
Gowalla	BPR-MF	4.29s	47
	SimpleX	342.33s	29
	ANS	194.63s	20
	PNN	45.66s	27

number of epochs required to converge. All methods are relatively efficient. As expected, BPR-MF is the most efficient method due to its simplicity. SimpleX demonstrates the worst efficiency due to its requirement of a large number of negative samples and the subsequent filtering of uninformative negative samples. PNN is the second most efficient method. With its outstanding performance, we believe it is a desirable choice for practical use.

5 RELATED WORK

Effectively leveraging the massive unlabeled data is a significant challenge. Most existing works [2, 3, 5, 18, 36] make an intuitive assumption that unlabeled data can directly provide negative signals. BPR loss function [18, 23, 34] randomly selects unlabeled data as negative samples. ENMF [3] efficiently learns model parameters from the whole unlabeled data. Yet, recent works reveal a nuanced reality: unlabeled data and negative samples harbor an inevitable disparity. Acknowledging this phenomenon, current research attempts to extract a subset closely approximating negative samples. SimpleX [15] extracts a large volume of high-information samples from unlabeled data to serve as negative samples. Zhuo *et al.* [40] introduces the concept of user interest boundary (UIB) to penalize samples crossing a threshold in the ranking task. DNS [36] selects unlabeled data that are more similar to a user as negative samples. GDNS [39] develops a gain-aware function to select real negative samples. SRNS [6] selects negative samples while incorporating a variance mechanism to filter out false negative samples. MixGCF [11] integrates information from a graph structure and positive samples to enhance the hardness of negative items. DENS [13] disentangles relevant and irrelevant factors of samples to select appropriate negative samples. ANS [38] proposes to generate synthetic negative samples to improve implicit CF. Regrettably, under this approach, the remaining data are overlooked, we want to unlock these hidden treasures behind unlabeled data to improve recommendation performance.

6 CONCLUSION

In this work, our paper introduces an innovative positive-neutral-negative (PNN) learning paradigm, which changes the utilization of massive unlabeled data in CF for recommender systems. By introducing a third neutral class and leveraging set-level ranking relationships, PNN effectively addresses the complexity inherent in unlabeled data. We have provided a concrete implementation of PNN that seamlessly integrates with multiple mainstream CF models. This implementation includes a novel semi-supervised learning method with a user-aware attention model to classify unlabeled data reliably, as well as a two-step centroid ranking approach to handle set-level rankings. Through extensive experiments on four public datasets, we demonstrate that PNN significantly improves the performance of various mainstream CF recommendation models. Our work marks a significant advancement in recommender systems, promising improved user experiences and broader applicability in real-world scenarios.

ACKNOWLEDGMENTS

This work was supported by the Heilongjiang Key R&D Program of China under Grant No. GA23A915 and the National Natural Science Foundation of China under Grant No. 62072136, with partial support from the Hong Kong Baptist University IG-FNRA project (RC-FNRA-IG/21-22/SCI/01).

REFERENCES

[1] Walter E Block, II Barnett, and Others. 2012. Transitivity and The Money Pump. *Quarterly Journal of Austrian Economics* 15, 2 (2012).

[2] Chong Chen, Min Zhang, Chenyang Wang, Weizhi Ma, Minming Li, Yiqun Liu, and Shaoping Ma. 2019. An Efficient Adaptive Transfer Neural Network for Social-Aware Recommendation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 225–234.

[3] Chong Chen, Min Zhang, Yongfeng Zhang, Yiqun Liu, and Shaoping Ma. 2020. Efficient Neural Matrix Factorization without Sampling for Recommendation. *ACM Transactions on Information Systems (TOIS)* 38, 2 (2020), 1–28.

[4] Jin Chen, Defu Lian, Binbin Jin, Kai Zheng, and Enhong Chen. 2022. Learning Recommenders for Implicit Feedback with Importance Resampling. In *Proceedings of the ACM Web Conference*. 1997–2005.

[5] Jingtao Ding, Yuhan Quan, Xiangnan He, Yong Li, and Depeng Jin. 2019. Reinforced Negative Sampling for Recommendation with Exposure Data.. In *International Joint Conference on Artificial Intelligence*. 2230–2236.

[6] 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.

[7] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572* (2014).

[8] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 639–648.

[9] Xiangnan He, Zhankui He, Xiaoyu Du, and Tat-Seng Chua. 2018. Adversarial Personalized Ranking for Recommendation. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 355–364.

[10] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In *Proceedings of the ACM Web Conference*. 173–182.

[11] Tinglin Huang, Yuxiao Dong, Ming Ding, Zhen Yang, Wenzheng Feng, Xinyu Wang, and Jie Tang. 2021. MixGCF: An Improved Training Method for Graph Neural Network-based Recommender Systems. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 665–674.

[12] Yangqin Jiang, Chao Huang, and Lianghao Xia. 2023. Adaptive Graph Contrastive Learning for Recommendation. *arXiv preprint arXiv:2305.10837* (2023).

[13] Riwei Lai, Li Chen, Yuhan Zhao, Rui Chen, and Qilong Han. 2023. Disentangled Negative Sampling for Collaborative Filtering. In *Proceedings of the 16th ACM International Conference on Web Search and Data Mining*. 96–104.

[14] Guanyu Lin, Chen Gao, Yu Zheng, Yinfeng Li, Jianxin Chang, Yanan Niu, Yang Song, Kun Gai, Zhiheng Li, Depeng Jin, et al. 2024. Inverse Learning with Extremely Sparse Feedback for Recommendation. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*. 396–404.

[15] Kelong Mao, Jieming Zhu, Jinpeng Wang, Quanyu Dai, Zhenhua Dong, Xi Xiao, and Xiuqiang He. 2021. SimpleX: A Simple and Strong Baseline for Collaborative Filtering. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 1243–1252.

[16] Lucas Maystre and Matthias Grossglauser. 2017. Just Sort It! A Simple and Effective Approach to Active Preference Learning. In *Proceedings of the 34th International Conference on Machine Learning*, Vol. 70. PMLR, 2344–2353.

[17] Steffen Rendle and Christoph Freudenthaler. 2014. Improving Pairwise Learning for Item Recommendation from Implicit Feedback. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*. 273–282.

[18] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*. AUAI Press, 452–461.

[19] Mamshad Nayeem Rizve, Kevin Duarte, Yogesh S Rawat, and Mubarak Shah. 2021. In Defense of Pseudo-Labeling: An Uncertainty-aware Pseudo-label Selection Famework for Semi-Supervised Learning. *arXiv preprint arXiv:2101.06329* (2021).

[20] Wentao Shi, Jiawei Chen, Fuli Feng, Jizhi Zhang, Junkang Wu, Chongming Gao, and Xiangnan He. 2023. On the Theories behind Hard Negative Sampling for Recommendation. In *Proceedings of the ACM Web Conference*. 812–822.

[21] Trevor Standley, Amir Zamir, Dawn Chen, Leonidas Guibas, Jitendra Malik, and Silvio Savarese. 2020. Which Tasks Should be Learned Together in Multi-Task Learning?. In *International Conference on Machine Learning*. 9120–9132.

[22] Xiaoyuan Su and Taghi M Khoshgoftaar. 2009. A Survey of Collaborative Filtering Techniques. *Advances in Artificial Intelligence* 2009 (2009).

[23] Jianing Sun, Wei Guo, Dengcheng Zhang, Yingxue Zhang, Florence Regol, Yaochen Hu, Huifeng Guo, Ruiming Tang, Han Yuan, Xiuqiang He, et al. 2020. A Framework for Recommending Accurate and Diverse Items using Bayesian Graph Convolutional Neural Networks. In *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2030–2039.

[24] Amos Tversky. 1969. Intransitivity of Preferences. *Psychological review* 76, 1 (1969), 31.

[25] Chenyang Wang, Yuanqing Yu, Weizhi Ma, Min Zhang, Chong Chen, Yiqun Liu, and Shaoping Ma. 2022. Towards Representation Alignment and Uniformity in Collaborative Filtering. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1816–1825.

[26] Jun Wang, Lantao Yu, Weinan Zhang, Yu Gong, Yinghui Xu, Benyou Wang, Peng Zhang, and Dell Zhang. 2017. IRGAN: A Minimax Game for Unifying Generative and Discriminative Information Retrieval Models. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 515–524.

[27] Tongzhou Wang and Phillip Isola. 2020. Understanding Contrastive Representation Learning through Alignment and Uniformity on the Hypersphere. In *International Conference on Machine Learning*. PMLR, 9929–9939.

[28] Wenjie Wang, Fuli Feng, Xiangnan He, Liqiang Nie, and Tat-Seng Chua. 2021. Denoising Implicit Feedback for Recommendation. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. 373–381.

[29] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural Graph Collaborative Filtering. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 165–174.

[30] Yuchao Wang, Haochen Wang, Yujun Shen, Jingjing Fei, Wei Li, Guoqiang Jin, Liwei Wu, Rui Zhao, and Xinyi Le. 2022. Semi-Supervised Semantic Segmentation using Unreliable Pseudo-Labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4248–4257.

[31] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. 2021. Self-Supervised Graph Learning for Recommendation. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*. 726–735.

[32] Enneng Yang, Junwei Pan, Ximei Wang, Haibin Yu, Li Shen, Xihua Chen, Lei Xiao, Jie Jiang, and Guibing Guo. 2023. Adatask: A Task-Aware Adaptive Learning Rate Approach to Multi-Task Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 10745–10753.

[33] Junliang Yu, Xin Xia, Tong Chen, Lizhen Cui, Nguyen Quoc Viet Hung, and Hongzhi Yin. 2023. XSimGCL: Towards Extremely Simple Graph Contrastive Learning for Recommendation. *IEEE Transactions on Knowledge and Data Engineering* (2023).

[34] 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 *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2084–2092.

[35] Junliang Yu, Hongzhi Yin, Xin Xia, Tong Chen, Lizhen Cui, and Quoc Viet Hung Nguyen. 2022. Are Graph Augmentations Necessary? Simple Graph Contrastive Learning for Recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1294–1303.

[36] Weinan Zhang, Tianqi Chen, Jun Wang, and Yong Yu. 2013. Optimizing Top-N Collaborative Filtering via Dynamic Negative Item Sampling. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 785–788.

[37] Wayne Xin Zhao, Shanlei Mu, Yupeng Hou, Zihan Lin, Yushuo Chen, Xingyu Pan, Kaiyuan Li, Yujie Lu, Hui Wang, Changxin Tian, et al. 2021. Recbole: Towards a unified, comprehensive and efficient framework for recommendation algorithms. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 4653–4664.

[38] Yuhan Zhao, Rui Chen, Riwei Lai, Qilong Han, Hongtao Song, and Li Chen. 2023. Augmented Negative Sampling for Collaborative Filtering. In *Proceedings of the 17th ACM Conference on Recommender Systems*. 256–266.

[39] Qiannan Zhu, Haobo Zhang, Qing He, and Zhicheng Dou. 2022. A Gain-Tuning Dynamic Negative Sampler for Recommendation. In *Proceedings of the ACM Web Conference*. 277–285.

[40] Jianhuan Zhuo, Qiannan Zhu, Yinliang Yue, and Yuhong Zhao. 2022. Learning Explicit User Interest Boundary for Recommendation. In *Proceedings of the ACM Web Conference*. 193–202.