Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.059262

ARTICLE

Improving Robustness for Tag Recommendation via Self-Paced Adversarial Metric Learning

Zhengshun Fei[1,*], Jianxin Chen[1], Gui Chen[2] and Xinjian Xiang[1,*]

[1]School of Automation and Electrical Engineering, Zhejiang University of Science and Technology, Hangzhou, 310023, China
[2]Bingwu (Ningbo) Intelligent Equipment Co., Ltd., Ningbo, 315600, China

*Corresponding Authors: Zhengshun Fei. Email: zsfei@zju.edu.cn; Xinjian Xiang. Email: 188002@zust.edu.cn

Received: 01 October 2024; Accepted: 09 December 2024; Published: 06 March 2025

ABSTRACT: Tag recommendation systems can significantly improve the accuracy of information retrieval by recommending relevant tag sets that align with user preferences and resource characteristics. However, metric learning methods often suffer from high sensitivity, leading to unstable recommendation results when facing adversarial samples generated through malicious user behavior. Adversarial training is considered to be an effective method for improving the robustness of tag recommendation systems and addressing adversarial samples. However, it still faces the challenge of overfitting. Although curriculum learning-based adversarial training somewhat mitigates this issue, challenges still exist, such as the lack of a quantitative standard for attack intensity and catastrophic forgetting. To address these challenges, we propose a Self-Paced Adversarial Metric Learning (SPAML) method. First, we employ a metric learning model to capture the deep distance relationships between normal samples. Then, we incorporate a self-paced adversarial training model, which dynamically adjusts the weights of adversarial samples, allowing the model to progressively learn from simpler to more complex adversarial samples. Finally, we jointly optimize the metric learning loss and self-paced adversarial training loss in an adversarial manner, enhancing the robustness and performance of tag recommendation tasks. Extensive experiments on the MovieLens and LastFm datasets demonstrate that SPAML achieves F1@3 and NDCG@3 scores of 22% and 32.7% on the MovieLens dataset, and 19.4% and 29% on the LastFm dataset, respectively, outperforming the most competitive baselines. Specifically, F1@3 improves by 4.7% and 6.8%, and NDCG@3 improves by 5.0% and 6.9%, respectively.

KEYWORDS: Tag recommendation; metric learning; adversarial training; self-paced adversarial training; robustness

1 Introduction

The explosive growth of internet data has made the challenge of information overload increasingly pronounced. As a result, it has become more difficult for users to efficiently access relevant information. Recommendation systems [1,2] analyze user preferences and behaviors to provide personalized information filtering. They have become essential technologies in fields such as e-commerce, social media, and online streaming. Traditional recommendation systems mainly focus on modeling the two-dimensional relationship between users and items, utilizing collaborative filtering (CF) [3] methods to predict user interest in items to deliver personalized recommendations. To explore more complex recommendation scenarios and meet practical needs, tag recommendation systems [4,5] introduce tag data into the interactions between users and items. This creates a three-dimensional relationship among users, items, and tags. By leveraging tags to represent diverse types of content, such as products, audio, video, and images. These systems better

capture the intrinsic connections between content and user preferences. This enables more personalized and efficient recommendations, especially in more complex recommendation scenarios.

Currently, most tag recommendation systems rely on tensor factorization to process high-dimensional interaction data. This technique decomposes the complex relationships among users, items, and tags into a product of multiple low-dimensional matrices, capturing latent associations and enabling personalized tag predictions. Examples of such methods include pairwise interaction tensor factorization (PITF) [6], nonlinear tensor factorization (NITF) [7], and attention-based neural tag recommendation (ABNT) [8]. In recent years, various deep learning techniques have been combined to more effectively learn and process the relationships between different entities. Graph neural network (GNN) has been successfully integrated with tensor factorization methods to extract higher-order collaborative signals among users, items, and tags. Notable examples include metapath and multi-interest aggregated graph neural network (M2GNN) [9] and attention learning tag-aware recommendation (TRAL) [10]. Tag recommendation also effectively captures diverse data structures by leveraging content features such as text, code, and sentiment information, enabling more accurate tag predictions. Representative methods include sentiment analysis matrix factorization (SAMF) [11], retrieval augmented cross-modal tag recommendation (RACM) [12], and Code-mixed representation learning for tag recommendation (CDR4Tag) [13]. Despite the success of tensor factorization in tag recommendation, its reliance on inner product for recommendation presents inherent limitations. While the inner product reduces computational complexity, it fails to satisfy the triangle inequality [14]. This leads to an inability to accurately measure the true distance among users, items, and tags. For instance, two similar users may be mapped to distant locations in the inner product space, resulting in suboptimal recommendation accuracy.

Metric learning has been widely applied in areas including image classification [15] and person re-identification [16]. In tag recommendation, to address the limitations of tensor factorization, researchers have introduced metric learning methods based on the triangle inequality principle. Metric learning computes the distance differences between entities to quantify their proximity. This ensures that the distances between users, items, and tags accurately reflect their similarity. However, metric learning in geometric space struggles with flexibility, particularly when representing highly similar samples. This can lead to overly close positioning of similar items, failing to accurately reflect user preferences. To address this limitation, methods such as latent relational metric learning (LRML) [17], hyperbolic space metric learning (HyperML) [18], multimodal attentive metric learning (MAML) [19], and collaborative residual metric learning (CoRML) [20] have been proposed. These approaches optimize the distance metrics among users, items, and tags. This allows the model to handle similar items more flexibly and reduces the occurrence of recommendation errors. The fixed margin problem in metric learning limits the model's performance when handling complex data distributions. A small fixed margin struggles to capture intricate interactions, whereas a larger margin creates convergence difficulties. To address this issue, methods such as symmetric metric learning (SML) [21] and probabilistic metric learning with adaptive margin (PMLAM) [22] have been introduced. Metric learning excels at capturing the similarity between entities but is highly sensitive to noise. Even minor noise or perturbations can cause inaccuracies in distance calculations, affecting recommendation accuracy and limiting the robustness of metric learning in tag recommendation systems.

Metric learning models are highly vulnerable to adversarial attacks [23,24], where even minor input perturbations can lead to high-confidence mispredictions. To address this issue, various defense strategies have been proposed in recent years to improve model robustness. Adversarial training methods aim to improve model robustness by incorporating adversarial examples into the training process. These methods enhance resistance to adversarial attacks either by adding regularization terms to constrain parameter updates or by optimizing feature representations. However, a common issue in adversarial training arises

when high-intensity adversarial perturbations significantly alter a sample's features, pushing them across the model's decision boundary. This results in the model being unable to correctly classify normal samples and adversarial samples, leading to overfitting. To address this, researchers have proposed curriculum learning-based adversarial training methods [25], where the attack intensity is gradually increased from weak to strong. This progression helps prevent high-intensity samples from crossing the decision boundary prematurely. Notable examples include curriculum adversarial training (CAT) [26], dynamic adversarial training (DAT) [27], and friendly adversarial training (FAT) [28]. Despite its potential, curriculum learning faces two significant challenges: catastrophic forgetting and the lack of a quantitative standard for adversarial sample intensity. Catastrophic forgetting occurs when models trained with high-intensity attacks fail to retain the adversarial features learned from low-intensity attacks. Additionally, the lack of a standardized way to quantify adversarial sample intensity poses a challenge. It complicates the accurate measurement of their impact during model training and evaluation.

Self-paced learning (SPL) is a learning paradigm that simulates the human cognitive process, gradually mastering sample features from simple to complex. SPL has demonstrated success in fields including fault diagnosis [29] and image clustering [30]. Building on this foundation, we propose a novel tag recommendation method, SPAML, to address the limitations of curriculum learning and enhance model robustness. SPAML leverages metric learning to precisely model the relationships among users, items, and tags. In addition to the standard metric model, we introduce a self-paced adversarial training model that quantifies the difficulty of adversarial samples based on their loss function values and dynamically adjusts their weights during training. The core of our method lies in progressively incorporating adversarial samples during training. Samples with lower loss values are prioritized, while those with higher losses are gradually introduced, ensuring that the model consistently retains knowledge from low-intensity adversarial samples, thereby mitigating the issue of catastrophic forgetting. The adversarial process involves jointly optimizing the metric learning loss and the self-paced adversarial training loss. Furthermore, we propose two weighting strategies: hard weighting scheme and soft weighting scheme, leading to two model variants: SPAML-H and SPAML-S. Experimental results demonstrate that SPAML consistently outperforms the most competitive baselines in tag recommendation tasks, validating the effectiveness of our method. In summary, our key contributions are as follows:

- SPAML introduces a novel method by combining metric learning and self-paced adversarial training. Metric learning accurately captures the distance relationships among users, items, and tags, while self-paced adversarial training effectively addresses catastrophic forgetting and compensates for the lack of a quantitative standard for attack intensity in curriculum learning-based adversarial training.
- We designed a self-paced adversarial training model that dynamically adjusts the weight of adversarial samples during training. Unlike traditional fixed adversarial training strategies, SPAML employs both hard weighting scheme and soft weighting scheme to effectively prevent overfitting to adversarial samples, thereby improving recommendation accuracy in complex environments.
- Comprehensive experiments were conducted on the MovieLens and LastFm datasets to evaluate SPAML's performance in tag recommendation tasks. Additionally, ablation studies were performed to quantify the contribution of each component to SPAML's overall performance. The results demonstrate that self-paced adversarial training significantly enhances the model's adversarial robustness, particularly on the larger and more complex LastFm dataset.

2 Related Work

Tag recommendation systems leverage tensor factorization methods to predict tag lists by utilizing interaction data among users, items, and tags. PITF [6] adopts a pairwise interaction approach, learning from both users-tags and items-tags interaction, achieving strong recommendation performance. Building on PITF, NITF [7] extends the feature space's capacity by using gaussian radial basis functions, enhancing the model's ability to capture nonlinear features. In contrast, ABNT [8] integrates a multilayer perceptron with an attention mechanism, enabling nonlinear modeling of entities. Additionally, SAMF [11] utilizes generating topic distributions from user and item reviews, creating user and item feature matrices, and quantifying sentiment information in reviews, which are then integrated into the users-items rating matrix to address data sparsity and trustworthiness issues. RACM [12] enhances the representation of titles, descriptions, and code by retrieving information from external knowledge sources and applying a cross-modal, context-aware mechanism for fine-grained feature extraction, thereby enhancing cross-modal retrieval and tag recommendation performance. CDR4Tag [13] employs a dual interaction strategy through code mixing to incorporate the deep semantic associations between software objects and tags into a joint representation space, enriching the semantics of software objects. M2GNN [9] constructs a heterogeneous information network using graph neural network to capture the semantic relationships among users, items, and tags, and uses a hierarchical aggregation framework to filter out irrelevant tags and interests, solving the issue of data sparsity in cross-domain recommendation. TRAL [10] generates dense tag feature vectors for users and items, and employs an attention pooling layer to automatically assign feature weights, learning nonlinear high-order interaction features to improve recommendation accuracy. While current tag recommendation methods have achieved some success in improving recommendation performance, they primarily focus on the correlations among users, items, and tags. However, these methods perform less effectively when dealing with adversarial disturbances such as malicious user inputs and noise. Therefore, enhancing the robustness of tag recommendation systems is a key focus of our research.

Metric learning methods have been extensively researched and applied in recommendation systems, effectively capturing the similarity between different entities, which provide more accurate recommendations. Among these methods, collaborative metric learning (CML) [31] was the first to apply metric learning to recommendation systems, addressing the issue that matrix factorization did not satisfy the triangle inequality. It achieves this by mapping users and items into a low-dimensional metric space, where distances represent user preferences for items. LRML [17] pointed out that CML tended to cluster similar users and items into the same point, exacerbating geometric inflexibility and limiting model performance. To solve this, LRML generates latent relation vectors using a memory attention mechanism to improve flexibility when handling similar users and items. HyperML [18] adopts hyperbolic metric learning in the Mobius rotation space to better capture the hierarchical and complex structure of users-items relationships. MAML [19] leverages the multimodal features of items and uses an attention mechanism to estimate user attention on different aspects of the item, overcoming the inflexibility limitations of CML. Moreover, CML is constrained by the fixed-margin impact on performance, particularly in highly sparse recommendation scenarios. Assigning a learnable margin hyperparameter for each user and item can improve model performance, but at a high computational cost and with a risk of overfitting. SML [21] highlighted that CML's use of a fixed margin leads to user conflict problems, where it only considers users-items relationships and may drag negative sample items toward positive sample items, contradicting the basic assumptions of metric learning. SML addresses this by assigning different adaptive margins for each user and item, thereby learning distinct vector representations. PMLAM [22] parameterizes users and items using Gaussian distributions and generates adaptive margins for different training samples, modeling distances between users and items with

the Wasserstein distance. CoRML [20] models the users-items distance residuals to learn generalized users-items distance metrics, capturing user preferences based on interaction signals. Despite the improvements in the performance of metric learning models through enhanced geometric structures and flexible margin adjustments, the robustness of these models remains limited. Our work enhances metric learning model robustness through self-paced adversarial training, ensuring the accuracy of recommendation results.

Adversarial training [23,32,33] is a critical method for mitigating the inherent vulnerabilities of deep learning models. By introducing carefully constructed, imperceptible adversarial examples into the training process, this technique significantly improves the model's robustness. There are two main branches of adversarial training. The first introduces regularization terms into the objective function to constrain changes in model parameters, thereby enhancing robustness. For example, adversarial model perturbation (AMP) [34] minimizes loss under the worst norm constraint instead of directly minimizing empirical risk, encouraging the model to favor flatter local minima, thereby improving generalization ability. Fast adversarial training via law (FGSM-LAW) [35] combines Lipschitz regularization with automatic weight averaging to improve model robustness and prevent catastrophic overfitting. The second focuses on optimizing feature representation by adjusting the structure of the feature space to increase resistance to adversarial attacks. Feature separation and recalibration (FSR) [36] separates input feature maps into robust and non-robust features, recovering potentially useful information and significantly improving the robustness of adversarial training methods, while maintaining low computational cost. Maximum mean discrepancy adversarial autoencoder (MMD-AAE) [37] introduces maximum mean discrepancy to align distributions from different domains and matches adversarial autoencoder learning to an arbitrary prior distribution, enabling generalizable feature representation that can adapt to unseen target domains. Adversarial feature desensitization (AFD) [38] learns adversarially robust features from a domain adaptation perspective, making the learned features both predictive and robust to adversarial attacks. In addition, CAT [26] trains the model by gradually increasing the difficulty of adversarial samples, starting with weaker adversarial samples and then progressing to stronger ones, thereby reducing the overfitting problem. DAT [27] introduces a first-order stability condition to evaluate model convergence quality and dynamically adjusts the training process based on the strength of the current adversarial attack. FAT [39] employs an early-stopping mechanism to recycle gradient information during model updates, thus eliminating the computational cost of generating adversarial samples while preventing overtraining. Despite the promise of curriculum learning, it suffers from catastrophic forgetting" and the lack of a quantitative standard for attack intensity. To tackle this issue, our work introduces an adaptive pacing strategy to adjust the training process, mitigating the limitations of curriculum learning and effectively defending against diverse adversarial attacks, thereby further enhancing model robustness.

3 Methodology

In this section, we will provide a detailed explanation of the proposed model.

As illustrated in the Fig. 1, we demonstrate the process from embedding to joint optimization using the metric learning model and the self-paced adversarial training model. The input data consisting of users, items, and tags is mapped through the embedding layer to generate low-dimensional embedding representations. In the metric learning model, we use Euclidean distance to measure the similarity between users, items, and their associated tags. By optimizing the loss function, we maximize the distance between positive and negative samples, ensuring the model accurately distinguishes the semantic relationships represented by different tags. In the self-paced adversarial training model, we apply small perturbations to the input samples using projected gradient descent (PGD) to generate adversarial samples. By calculating the distances of these adversarial samples in the embedding space, the model learns to effectively distinguish

between similar yet challenging samples. The self-paced adversarial training module determines the difficulty level of adversarial samples based on their loss values and dynamically adjusts the weights of difficult samples. The model begins by training on simple adversarial samples that are easier to distinguish and gradually introduces more complex adversarial samples. This self-paced learning model, which progressively increases sample difficulty, enables the model to steadily improve its ability to fit complex adversarial samples while retaining knowledge of "simple" adversarial samples. Finally, the losses from both metric learning and self-paced adversarial training are jointly optimized in an adversarial manner, enhancing the model's robustness and its generalization ability to unseen data.

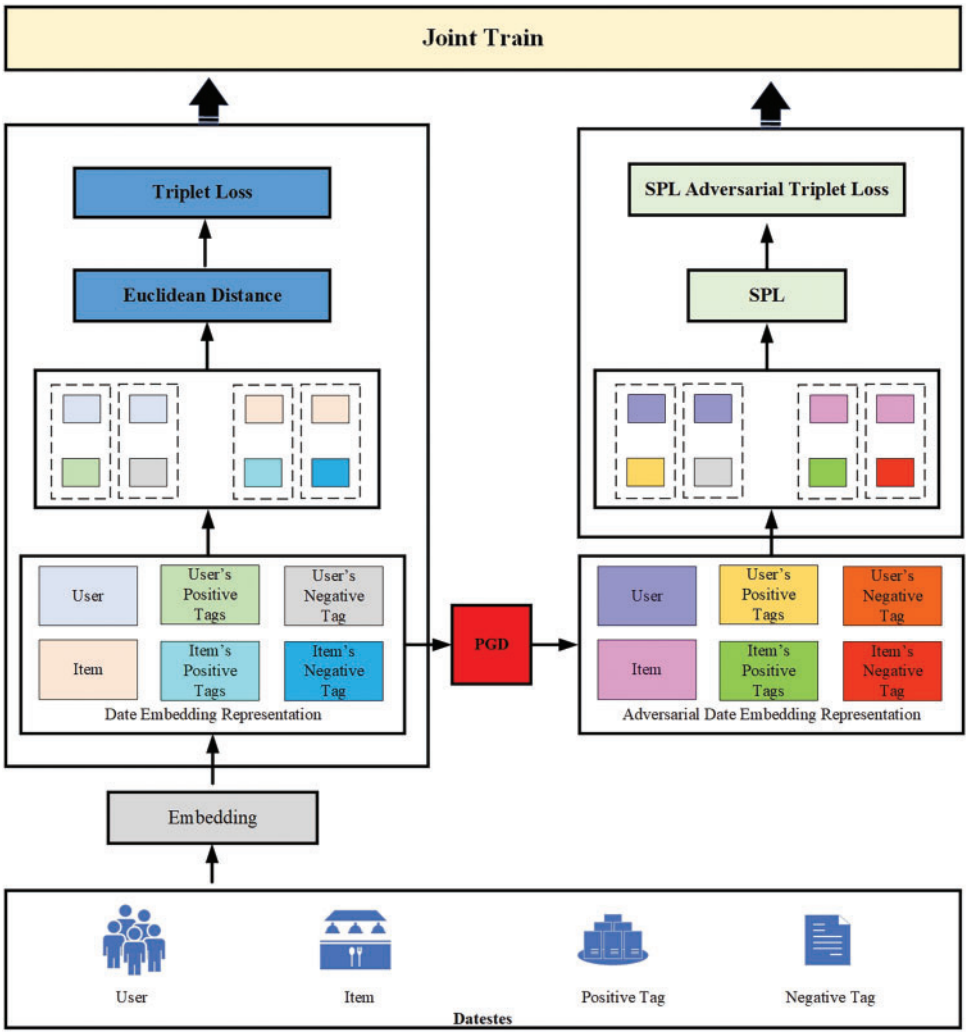

Figure 1: The proposed framework for SPLMA

3.1 Top-N Tag Recommendation

Tag recommendation aims to accurately predict the possibility of a user selecting a certain tag for an item. It includes three types of entities: the set of users $\mathcal{U}$, the set of items $\mathcal{I}$, and the set of tags $|\mathcal{T}|$. The number of the users set is denoted as $|\mathcal{U}|$, the number of the items set is denoted as $|\mathcal{I}|$, and the number of

the tags set is denoted as $|\mathcal{T}|$. Historical interaction data among users, items, and tags collectively forms a three-dimensional set, which takes the form of:

$$\mathcal{S} \subseteq \mathcal{U} \times \mathcal{I} \times \mathcal{I} \tag{1}$$

In tag recommendation, user IDs, item IDs, and tag IDs typically exist in a sparse ID format. Since these IDs do not contain numerical value information, they are not directly suitable for computations. To transform these sparse IDs into numerical representations suitable for the model, we employ an embedding technique. Specifically, the model assigns each ID a low-dimensional dense embedding vector, stored in embedding matrices. For a given user u, item i, positive tag t, and negative tag t', their corresponding embedding representations are obtained by looking up the embedding matrices, which are shown as follows:

$$\begin{aligned} x_u &= \mathbf{U}[u], \quad x_i = \mathbf{I}[i] \\ x_{t_u} &= \mathbf{T_U}[t], \quad x_{t_i} = \mathbf{T_I}[t] \\ x_{t'_u} &= \mathbf{T_U}[t'], \quad x_{t'_i} = \mathbf{T_I}[t'] \end{aligned} \tag{2}$$

where x_u indicates the embedding representation of users, x_i indicates the embedding representation of items, x_{t_u} indicates the embedding representation of tags corresponding to users, x_{t_i} indicates the embedding representation of tags corresponding to items, $\mathbf{U} \in \mathbb{R}^{|\mathcal{U}| \times k}$, $\mathbf{I} \in \mathbb{R}^{|\mathcal{I}| \times k}$, $\mathbf{T_U} \in \mathbb{R}^{|\mathcal{T}| \times k}$, $\mathbf{T_I} \in \mathbb{R}^{|\mathcal{T}| \times k}$ represent feature matrices for users, items, user-specific tags, and item-specific tags, the k represents the dimension of embedding.

The task of tag recommendation involves establishing a scoring function Y that captures the implicit feedback of users selecting tags for items. Specifically, $Y \in \mathbb{R}^{|\mathcal{U}| \times |\mathcal{I}| \times |T|}$, where the dimensions are $|\mathcal{U}| \times |\mathcal{I}| \times |\mathcal{T}|$, corresponding to the number of users, items, and tags. Each element $\mathcal{Y}_{(u,i,t)}$ in the Y represents whether the user has used the tags for the items. If the user has annotated the items with the tags, $\mathcal{Y}_{(u,i,t)} = 1$, otherwise $\mathcal{Y}_{(u,i,t)} = 0$, which can formally be expressed as:

$$\mathcal{Y}_{(u,i,t)} = \begin{cases} 1, & (u,i,t) \in \mathcal{S} \\ 0, & otherwise \end{cases} \tag{3}$$

During the tag recommendation process, the system calculates relevance scores for all potential tags based on users-items interaction $T_{(u,i)}$. Based on these scores, the system ranks the tags and selects the top N tags with the highest scores to generate a Top-N recommendation list, which can be expressed as:

$$T_{(u,i,N)} = \arg \max_{t \in T}^{N} y_{(u,i,t)} \tag{4}$$

where N represents the length of tag recommendation list.

3.1.1 Tag Recommendation Based on Metric Learning

In the specific task of tag recommendation, metric learning methods model the users-items-tags distance by calculating the distances between embedding representations to assess the relevance of tags to users and items, quantifying the similarity or dissimilarity between the two embedding representations. Therefore, the distance among users, items, and tags $\mathcal{Y}(u,i,t)$, which can be expressed as:

$$\mathcal{Y}(u,i,t) = \| x_u - x_{t_u} \|_2^2 + \| x_i - x_{t_i} \|_2^2 \tag{5}$$

At the same time, we aim to enforce the separation of the distance among users, items, and negative tags, thereby ensuring that the relationship between negative tags and users-items pairs is weaker than that of positive tags. Thus, the distance relationship $\mathcal{Y}(u, i, t')$, which can be expressed as:

$$\mathcal{Y}(u, i, t') = \| x_u - x_{t'_u} \|_2^2 + \| x_i - x_{t'_i} \|_2^2 \tag{6}$$

Triplet loss is a widely used metric learning method. Our goal is to ensure that in the embedding space, the distance between users and positive tags is less than that between users and negative tags, thereby enabling the model to correctly distinguish positive tags from negative tags. The objective function for triplet loss can be formulated as follows:

$$\mathcal{L}_{ml} = \sum_{(u,i,t)\in\mathcal{F}} \sum_{(u,i,t')\notin\mathcal{F}} [\mathcal{Y}(u, i, t) - \mathcal{Y}(u, i, t') + m]_+ \tag{7}$$

where $\mathcal{F}$ represents the set of training instances. The value of m is a fixed margin that ensures sufficient distinction between positive tags and negative tags. The $[x]_+ = \max(x, 0)$ denotes the positive part of the value, ensuring that the loss function has non-zero values.

Although metric learning can effectively utilize the embedding distances among users, items, and tags to achieve good recommendation performance, solely relying on these distance relationships may lead to weak generalization of the model. The model has two main limitations:

1. Duan et al. [40] and Chen et al. [41] point out that metric learning models are highly sensitive to small perturbations or noise, which can undermine their robustness in real-world scenarios. Furthermore, Wang et al. [42] emphasized that minor fluctuations in user preferences during tag recommendation tasks can lead to variations in behavioral data, thereby impacting the embedding representations of items and tags.
2. Mao et al. [43] pointed out that metric learning models are prone to overfitting the training data. Li et al. [44] further noted that in the embedding space, if the models focuses solely on bringing the positive tags close to the users and items in the training set, while ignoring noise or outliers, their performance on the test set or new data may deteriorate.

3.2 Adversarial Training

To strengthen the robustness and generalization ability of metric learning, we introduce adversarial training into the metric learning model. By adding targeted adversarial perturbations, the model can still make accurate predictions in the presence of such perturbations. Specifically, we use the PGD method, which iteratively generates adversarial perturbations through repeated updates, ensuring that the magnitude of the perturbations remains within a limited range. This process continuously adjusts the perturbations to generate adversarial samples with higher attack strength. To improve the robustness of metric learning in modeling the relationships among users, items, and tags, we design targeted adversarial perturbations for the metric learning model. Specifically, adversarial perturbations are added to the embedding representations. After $k+1$ iterations, the adversarial samples are defined as:

$$\begin{aligned}
&\tilde{\mathbf{x}}_u^{(k+1)} = \mathrm{Proj}_{\mathbf{x}_u}\left(\tilde{\mathbf{x}}_u^{(k)} + \varepsilon \cdot \mathrm{sign}(\nabla_{\mathbf{x}_u}\mathcal{L}_{ml})\right), \tilde{\mathbf{x}}_i^{(k+1)} = \mathrm{Proj}_{\mathbf{x}_i}\left(\tilde{\mathbf{x}}_i^{(k)} + \varepsilon \cdot \mathrm{sign}(\nabla_{\mathbf{x}_i}\mathcal{L}_{ml})\right) \\
&\tilde{\mathbf{x}}_{t_u}^{(k+1)} = \mathrm{Proj}_{\mathbf{x}_{t_u}}\left(\tilde{\mathbf{x}}_{t_u}^{(k)} + \varepsilon \cdot \mathrm{sign}(\nabla_{\mathbf{x}_{t_u}}\mathcal{L}_{ml})\right), \tilde{\mathbf{x}}_{t_i}^{(k+1)} = \mathrm{Proj}_{\mathbf{x}_{t_i}}\left(\tilde{\mathbf{x}}_{t_i}^{(k)} + \varepsilon \cdot \mathrm{sign}(\nabla_{\mathbf{x}_{t_i}}\mathcal{L}_{ml})\right) \\
&\tilde{\mathbf{x}}_{t'_u}^{(k+1)} = \mathrm{Proj}_{\mathbf{x}_{t_u}}\left(\tilde{\mathbf{x}}_{t'_u}^{(k)} + \varepsilon \cdot \mathrm{sign}(\nabla_{\mathbf{x}_{t_i}}\mathcal{L}_{ml})\right), \tilde{\mathbf{x}}_{t'_i}^{(k+1)} = \mathrm{Proj}_{\mathbf{x}_{t_i}}\left(\tilde{\mathbf{x}}_{t_i}^{(k)} + \varepsilon \cdot \mathrm{sign}(\nabla_{\mathbf{x}_{t_i}}\mathcal{L}_{ml})\right)
\end{aligned} \tag{8}$$

where $\tilde{\mathbf{x}}^{(k)}$ denotes the perturbation after the k iteration, and ϵ represents the step size controlling the magnitude of the perturbation. The function Proj projects the updated perturbations back into the allowable range to ensure that the generated adversarial perturbations do not excessively deviate. The operator $sign$ retains only the sign and direction of the gradient. $(\nabla_{\mathbf{x}}\mathcal{L}ml)$ denotes the gradient of the loss function with respect to the input x, where x can be $x_u, x_i, x_{t_u}, x_{t_i}, x_{t'_u}, x_{t'_i}$. In each iteration, we calculate the current perturbation and generate the next adversarial perturbation. By gradually enhancing these perturbations through iteration and adding them to the original embedding representation, we eventually generate high-intensity adversarial samples.

After obtaining the adversarial samples, we integrate them into the metric learning model for adversarial training. The model demonstrates greater stability when exposed to adversarial inputs, thereby improving its robustness and generalization ability. To achieve this, we designed a loss function that incorporates adversarial perturbations, as described below:

$$\mathcal{L}_{adv} = \sum_{(u,i,t)\in S} \sum_{(u,i,t')\in S} \left[(\|\tilde{x}_u - \tilde{x}_{t_u}\|_2^2 + \|\tilde{x}_i - \tilde{x}_{t_i}\|_2^2) - (\|\tilde{x}_u - \tilde{x}_{t'_u}\|_2^2 + \|\tilde{x}_i - \tilde{x}_{t'_i}\|_2^2) - m \right]_+ \tag{9}$$

where $\tilde{x}_u, \tilde{x}_i, \tilde{x}_{t_u}, \tilde{x}_{t'_u}, \tilde{x}_{t_i}, \tilde{x}_{t'_i}$ represent the adversarial samples of users, items, user-specific positive tags and negative tags, item-specific positive tags and negative tags respectively. The model enhances its robustness and generalization performance by maximizing the distance between adversarial positive samples and adversarial negative samples in the embedding space.

Although adversarial training has made significant progress in enhancing the robustness of models, it still has two main limitations:

1. Cai et al. [26] and Wang et al. [45] pointed out that adversarial training often focuses on high-intensity adversarial samples, which are closer to the model's decision boundary and more likely to contain noise. The model's excessive attention to these high-noise samples can cause it to deviate from the normal decision boundary, negatively impacting its performance on real-world data. This phenomenon is referred to as overfitting to adversarial samples. Additionally, Cai et al. [26] noted that curriculum learning-based adversarial training alleviates overfitting to some extent, but it also introduces new challenges.
2. He et al. [46] highlighted the lack of a quantitative measure for attack intensity. Specifically, the number of iterations in the PGD directly affects the strength of the generated adversarial samples. Too few iterations result in weak perturbations that fail to adequately test the model's robustness, whereas too many iterations increase computational costs and reduce training efficiency.
3. Cai et al. [26] further noted that adversarial training starts directly with high-intensity adversarial samples, which may lead the model to overlook simpler adversarial samples. As training progresses, the model tends to focus on more high-intensity adversarial examples, forgetting the features learned from simpler ones. This could weaken the model's defense mechanisms when encountering low-intensity adversarial attacks.

3.2.1 Self-Paced Adversarial Training

To overcome the challenges faced by curriculum learning-based adversarial training, we propose a novel method called self-paced adversarial training [46]. This method tackles the lack of a quantitative measure for attack intensity by evaluating the difficulty of adversarial samples and dynamically adjusting weights using soft and hard weighting schemes [25], based on the adversarial training loss. This indirectly achieves a quantification of attack intensity. By assigning higher weights to simpler adversarial samples, the model

initially focuses on learning from these samples, which are farther from the decision boundary and contain less noise. This facilitates the stable optimization of the decision boundary. As training progresses, more complex adversarial samples are gradually incorporated. The dynamic weight adjustment prevents the model from exclusively focusing on complex adversarial samples, thus avoiding the forgetting of features learned from simpler ones and enhancing the model's robustness. $\mathcal{L}_{spat}$ can be expressed by the following formula:

$$\mathcal{L}_{spat} = \sum_{i=1}^{M} v_i \cdot \mathcal{L}_{adv}(i) + \sum_{i=1}^{M} g(\mathbf{v}) \tag{10}$$

where M is the number of triples, v_i is a weight parameter representing the importance of the adversarial samples, and L_{adv} is the triplet loss function value of the adversarial samples. The $g(\mathbf{v})$ is a newly introduced regularization term, which typically employs two weighting strategies: soft weighting scheme and hard weighting scheme.

Soft weighting scheme is a continuous weighting method where the model learns from all adversarial samples while adjusts the weight of each sample based on the adversarial training loss. The soft weighting scheme formula is as follows:

$$g(\mathbf{v}) = -\sum_{i=1}^{M} \lambda \ln(v_i + \epsilon), \quad \text{where } v_i = \frac{1}{1 + \exp\left(\alpha \cdot (L_{adv} - \beta)\right)} \tag{11}$$

where λ controls the proportion of particularly simple adversarial sample. ϵ is a small value used to prevent numerical issues when calculating logarithms and to penalize those samples with weights that are too high or too low, encouraging the model to learn from all samples equally. The model adjusts the sample weight v_i to control the importance of adversarial samples during training, and α controls the sensitivity of weight variation. A larger α makes weight adjustments more sensitive, causing the weight of an adversarial sample to shift from 1 to nearly 0 more quickly, making the model focus more on easy adversarial sample. A smaller α makes the weight change smoother, making the model's attention to all adversarial samples more balanced. β is a parameter used to determine the difficulty of adversarial samples. It determines which samples are considered "easy" and receive higher weights, and which are considered "hard" and receive lower weights. By tuning these two parameters, the model can dynamically adjust the adversarial sample weights, allowing it to prioritize easy adversarial sample first and gradually introduce more complex adversarial sample.

Hard weighting scheme is a binary adversarial sample selection method where the model only trains on adversarial samples with loss values below a predefined threshold, ignoring those that do not meet the criteria. This approach is simple and efficient by enabling the model to prioritize training on easily learnable adversarial samples through an initial filtering process. The hard weighting formula is as follows:

$$g(\mathbf{v}) = \sum_{i=1}^{N} v_i, \quad \text{where } v_i = \begin{cases} 1, & \text{if } \mathcal{L}_{adv} \leq \gamma \\ 0, & \text{if } \mathcal{L}_{adv} > \gamma \end{cases} \tag{12}$$

where γ is the threshold for selecting adversarial sample. The weight v_i assigned to the adversarial sample is either 0 or 1. When $\mathcal{L}_{adv}$ is less than or equal to γ, the adversarial sample weight is 1, meaning the adversarial sample is included in the training. When $\mathcal{L}_{adv}$ exceeds γ, the adversarial sample weight is 0, and the adversarial sample is ignored. This mechanism ensures that the model focuses on easier to learn sample during the early stages of training, while more difficult samples are ignored.

3.2.2 Joint Training

The metric learning model optimizes the distances between users, items, and tags, while the self-paced adversarial training model enhances the model's robustness by progressively increasing the difficulty of adversarial samples. To achieve this goal, we designed a joint training loss function, enabling the model to optimize both objectives during training. The joint training loss function integrates the adversarial training loss with the original metric learning loss, enhancing the model's robustness ability and improving stability of prediction results. Therefore, the final loss function is as follows:

$$\mathcal{L}_{SPAML} = \mathcal{L}_{ml} + \mu \mathcal{L}_{spat} \tag{13}$$

where μ controls the intensity of adversarial training attacks, the distance metric loss function and the self-paced adversarial training loss function are jointly trained in an adversarial manner.

3.3 Computational Complexity

The computational complexity of the SPAML model is composed of two parts: the metric learning loss $\mathcal{L}_{ml}$ and the self-paced adversarial training loss $\mathcal{L}_{spat}$.

Metric learning loss $\mathcal{L}_{ml}$: The computation of positive-negative tag pairs requires two nested iterations over the set of training instances $\mathcal{F}$, resulting in a complexity of $O(|\mathcal{F}|^2)$, where $|\mathcal{F}|$ represents the number of training instances. Each computation involves calculating the distance between the embeddings of users, items, and tags, with a complexity of $O(K)$, where K denotes the embedding dimension. The total computational complexity of distance calculations is therefore $O(|\mathcal{F}|^2 \cdot K)$.

Self-paced adversarial training loss $\mathcal{L}_{spat}$: This consists of adversarial sample generation $\mathcal{L}_{adv}$ and weight strategies $g(\mathbf{v})$. For adversarial sample generation, the gradient computation for each iteration has a complexity of $O(|\mathcal{F}|^2 \cdot K)$, and generating adversarial samples requires t iterations, leading to a total complexity of $O(t \cdot |\mathcal{F}|^2 \cdot K)$. The weight strategies involve two schemes: the hard weighting scheme, which requires simple comparisons of loss values with a threshold and has a complexity of $O(\mathcal{F})$, and the soft weighting scheme, which involves logarithmic and exponential operations on weights and also has a complexity of $O(\mathcal{F})$. The total computational complexity of $\mathcal{L}_{spat}$ is $O(t \cdot |\mathcal{F}|^2 \cdot K)$. When the number of iterations for adversarial sample generation $t > 1$, the dominant complexity is $O(|\mathcal{F}|^3 \cdot K)$. The overall computational complexity of SPAML is $O(|\mathcal{F}|^2 \cdot K + |\mathcal{F}|^3 \cdot K)$.

4 Experiments

4.1 Datasets

We conducted experiments using two publicly available datasets, MovieLens and LastFM, to evaluate the baselines and the proposed method. Table 1 presents the general statistical information for the different datasets.

Table 1: General statistical information on the different datasets

Dataset	Users	Items	Tags	Training set	Testing set
Movielens	469	1524	1017	30503	6911
LastFm	966	3870	1204	105056	28889

MovieLens: A typical dataset for movie recommendation. In this dataset, users' tagging behavior towards movies reveals users' interests and preferences for movies, which is valuable for researching and evaluating recommendation methods in the movie recommendation field.

LastFM: A dataset in the domain of music recommendation and music information retrieval. By analyzing the tags assigned to music by users, it helps to understand users' interests and preferences, providing valuable information for music recommendation.

The MovieLens dataset can be downloaded from https://grouplens.org (accessed on 08 December 2024), and the LastFM dataset from http://www.last.fm (accessed on 08 December 2024). Given the differences in scale and sparsity between these two datasets, we performed preprocessing to extract a core subset of interaction data, specifically obtaining a p-core for each dataset. In both MovieLens and LastFM, interaction data among users, items, and tags is typically sparse. To ensure sufficient interactions between users, items, and tags, we adopted the 10-core dataset, where each user, item, and tag must appear at least 10 times. For testing, the last interaction of each user with an item using a tag was used as the testing set, while the remaining interactions served as the training set. This preprocessing step ensures a fair and consistent evaluation of different methods under the same conditions. In comparison, 5-core or 3-core datasets result in sparser user-item-tag interactions, which limits the model's ability to capture meaningful relationships. Similarly, the original datasets are even more sparse, containing a large proportion of low-quality interactions that degrade model training and increase computational overhead. Therefore, the 10-core dataset strikes a balance between data density and volume, facilitating effective and reliable evaluations. We employed F1@N and NDCG@N as evaluation metrics to assess performance of various baseline methods and our proposed approach.

4.2 Baselines

We evaluated the effectiveness of our proposed method by comparing it with the following baseline methods:

CF [3]: Utilizes matrix factorization to convert the users-tags interactions into low-rank vector inner products for similarity prediction.

PITF [6]: Models the interaction between users-tags and items-tags relationships, predicting scores through inner product operations.

NITF [7]: Introduces the Gaussian radial basis function, expanding the feature space capacity and enhancing the model's ability to capture nonlinear features.

ABNT [8]: Leverages fully connected networks and attention mechanisms to capture complex nonlinear relationships among users, items, and tags, thus providing the model with stronger feature extraction capabilities.

CML [31]: Adopts metric learning to model the distance relationships between user-tags and item-tags.

SML [21]: Introduces adaptive margin to dynamically adjust the distances between entities, improving the model's flexibility and robustness.

LRML [17]: Generates latent relation vectors through memory attention mechanisms, enhancing the model's ability to capture deep relationships.

AML [41]: Enhances model robustness by adding adversarial perturbations and optimizes model parameters to reduce overfitting.

ATHN [47]: Improves model stability and prediction accuracy in diverse datasets by generating hard samples and adopting the adversarial learning method.

4.3 Evaluation Metrics

We assess the recommendation quality of all methods in the experiment using F1@N and NDCG@N, which are standard metrics commonly employed in the recommendation field.

$$\text{Precision@N} = \frac{1}{|P_{\mathcal{S}_{test}}|} \sum_{(u,i)\in P_{\mathcal{S}_{test}}} \frac{|Top(u,i,N) \cap \mathcal{S}_{test}|}{N} \tag{14}$$

$$\text{Recall@N} = \frac{1}{|P_{\mathcal{S}_{test}}|} \sum_{(u,i)\in P_{\mathcal{S}_{test}}} \frac{|Top(u,i,N) \cap \mathcal{S}_{test}|}{|\{t|t \in \mathcal{S}_{test}\}|} \tag{15}$$

$$F1@N = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{16}$$

$$\text{DCG@N} = \sum_{i=1}^{N} \frac{\text{rel}(i)}{\log_2(i)} \tag{17}$$

$$\text{NDCG@N} = \frac{\text{DCG@N}}{\text{IDCG@N}} \tag{18}$$

where $\mathcal{S}_{test}$ represents the testing set, $P_{\mathcal{S}_{test}}$ represents the number of users-items pairs in the testing set, and *IDCG* represents the ideal discounted cumulative gain, which is maximum possible *DCG* that can be obtained through an ideal ranking.

4.4 Implementation

All baseline methods are implemented on the Tensorflow 1.15 framework using NVIDIA GeForce RTX 3090 GPU, The hyperparameters are set according to the best-reported values in the respective literature. The learning rate is set to 0.001, and the embedding dimension k for all methods is 64. For metric learning-based methods, the fixed margin m is uniformly set to 0.2. In the proposed method, the step size ε for PGD is set to 0.01, and the iteration t is set to 10. For the soft weighting scheme, the proportion of simple adversarial samples λ is 0.1, and the adversarial training intensity μ is set to 0.01. The Adam algorithm with a mini-batch size of 1024 is employed to optimize all models.

4.5 Performance Comparison

The experimental results on the LastFm and Movielens datasets are presented in Tables 2 and 3.

Table 2: Performance comparison on Movielens datasets

Method	F1@3	F1@5	F1@10	NDCG@3	NDCG@5	NDCG@10
CF	0.09883	0.08764	0.06540	0.15432	0.18076	0.21214
PITF	0.21364	0.17725	0.11855	0.31820	0.36085	0.39933
NITF	0.21140	0.17817	0.11894	0.31683	0.36228	0.40124
ABNT	0.09796	0.08445	0.06275	0.15495	0.17851	0.20716
CML	0.20807	0.17190	0.11495	0.31443	0.35604	0.39337
SML	0.16322	0.13389	0.08890	0.25633	0.28732	0.31556
LRML	0.20446	0.17166	0.11436	0.31153	0.35482	0.39213
AML	0.21140	0.17735	0.12049	0.31977	0.36458	0.40636

(Continued)

Table 2 (continued)

Method	F1@3	F1@5	F1@10	NDCG@3	NDCG@5	NDCG@10
ATHN	0.21379	0.18049	0.12160	0.32188	0.36767	0.40910
SPAML-H	0.21813	0.18284	0.12262	0.32789	0.37381	0.41496
SPAML-S	0.22008	0.18294	0.12259	0.32774	0.37284	0.41288

Table 3: Performance comparison on LastFm datasets

Method	F1@3	F1@5	F1@10	NDCG@3	NDCG@5	NDCG@10
CF	0.10004	0.09207	0.07337	0.15098	0.18222	0.22286
PITF	0.16423	0.14975	0.14975	0.24499	0.29461	0.34722
NITF	0.17453	0.16205	0.12271	0.25733	0.31315	0.41421
ABNT	0.11016	0.10506	0.08417	0.16382	0.20280	0.25051
CML	0.18246	0.16333	0.12011	0.27188	0.32330	0.37854
SML	0.14692	0.13106	0.09651	0.22357	0.26440	0.30926
LRML	0.18244	0.16367	0.12017	0.27390	0.32562	0.38063
AML	0.18916	0.17077	0.12618	0.28276	0.33765	0.39635
ATHN	0.19182	0.17503	0.12933	0.28670	0.34412	0.40507
SPAML-H	0.19505	0.17639	0.12987	0.29120	0.34846	0.40867
SPAML-S	0.19488	0.17613	0.13018	0.29044	0.34731	0.40835

The CF method, which predicts similarities by converting the users-tags matrices into low-rank vector inner products, shows relatively weak performance. Compared to tensor decomposition-based methods (PITF, NITF, and ABNT), PITF calculates similarities between users-tags and items-tags through inner products. However, it does not satisfy the triangular inequality and neglects distance metrics, limiting its effectiveness. NITF improves over PITF by expanding the feature space using Gaussian radial basis functions. ABNT further enhances this by utilizing fully connected networks and attention mechanisms to model nonlinear relationships between users and items. Nevertheless, due to its reliance on a large number of parameters, ABNT's performance in recommendation tasks remains suboptimal. In comparison, the proposed SPMLA-H and SPMLA-S methods exhibit superior results. Specifically, on the Movielens dataset, SPMLA-H improves F1@3 and NDCG@3 by 2.1% and 3.0% over PITF, respectively, while SPMLA-S shows even greater improvements, with 3.1% and 2.9% increases in the same metrics. Compared to the tensor decomposition-based optimal baseline method (NIFT), the F1@3 scores increased by 5.7% and 11.7%, respectively. SPMLA also demonstrates remarkable performance on the LastFm dataset. These results indicate that the proposed method significantly enhances recommendation accuracy by more effectively capturing the similarity relationships among users, items, and tags.

Compared to metric learning-based methods (CML, LRML, and SML), CML models the relationships between users-tags and items-tags through distance metrics. LRML enhances CML by introducing a memory attention mechanism to generate latent relation vectors, while SMT establishes adaptive margins to adjust entity distances. metric learning methods effectively address the limitations of PITF by using distance metrics, thereby improving recommendation accuracy. SPMLA-H improved by 4.8% and 4.2% in F1@3 and NDCG@3, respectively, over CML on the Movielens dataset, while SPMLA-S improved by 5.7% and 4.2%. Compared to metric learning-based optimal baseline method (CML), the F1@3 scores increased by 5.7%

and 6.9%, respectively. The proposed method incorporates an adversarial training mechanism into metric learning models, generating adversarial samples to improve the model's robustness and generalization ability in noisy data and adversarial attack scenarios. Experimental results demonstrate that traditional metric learning methods, lacking adversarial training, perform poorly in complex environments. Experiments on the LastFM dataset further validate that adversarial training significantly enhances the model's robustness and adaptability, particularly in handling noisy data and adversarial scenarios, showing clear advantages in managing complex data environments.

Compared to adversarial learning-based methods (AML and ATHN), AML enhances model robustness by introducing adversarial perturbations, which force the model to strengthen its defense mechanisms. However, its adversarial training strategy remains relatively simplistic. ATHN focuses on improving feature extraction through adversarial training by generating hard negative samples, though it primarily emphasizes feature extraction rather than broader adversarial methods. On the LastFm and Movielens datasets, compared to adversarial learning-based method (ATHN), the F1@3 scores increased by 2.9% and 1.6%, respectively. The proposed method employs self-paced adversarial training strategy that dynamically adjusts the weight distribution of adversarial samples during training, achieving quantitative control over attack intensity. This strategy progressively adapts to adversarial samples of varying difficulty, effectively preventing the model from forgetting the features of simpler adversarial samples while focusing solely on complex ones. It also enhances the model's robustness against strong adversarial attacks and reduces the risk of overfitting. Consequently, SPAML demonstrates outstanding recommendation performance and stability when handling diverse and challenging adversarial samples. Furthermore, a comparison between the two weighting schemes, SPAML-H and SPAML-S, indicates that the soft weighting scheme in SPAML-S considers additional factors during training, enabling more refined and efficient adaptability.

4.6 Effect of Hyperparameter

This section analyzes the effect of hyperparameters on model performance, with a focus on the parameters α and β in the soft weighting scheme, and γ in the hard weighting scheme.

4.6.1 Soft Weighting Scheme

For the soft weighting scheme, we fine-tuned two key parameters: α, which controls the sensitivity of the weight changes, as shown in Fig. 2. β, which defines the parameter for selecting adversarial samples during training, as shown in Fig. 3.

α Analysis: We fixed the parameter β at 0.4 across different datasets and varied α within the range of 0.1 to 0.5. The experiments revealed that setting $\alpha = 0.4$ struck the optimal balance between training speed and recommendation accuracy. Smaller α values resulted in smoother weight adjustments, which led to more uniform attention across all adversarial samples, which reduced the model's ability to adapt to more difficult samples. Conversely, larger α values made the model excessively sensitive to weight changes, prioritizing challenging adversarial samples too quickly and negatively affecting generalization and the model's ability to handle new data.

β Analysis: We varied the parameter β within the range of 0.1 to 0.3 to evaluate its effect on sample selection during training. The results demonstrated that when $\beta = 0.4$, the model achieved optimal performance. At this value, the model effectively filtered out difficult samples in the early stages of training, progressively introducing more challenging ones as the training advanced. A lower β value caused the model to focus too heavily on simple samples, leading to slower convergence. On the other hand, a higher β value introduced complex samples too early, destabilizing the learning process and negatively affecting overall performance.

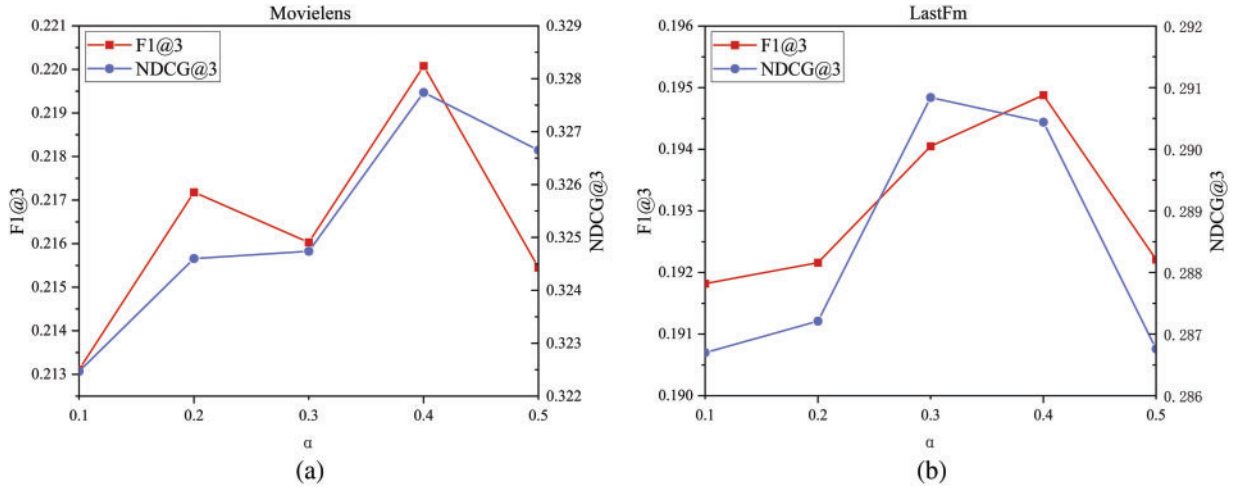

Figure 2: Effect of α on recommendation performance

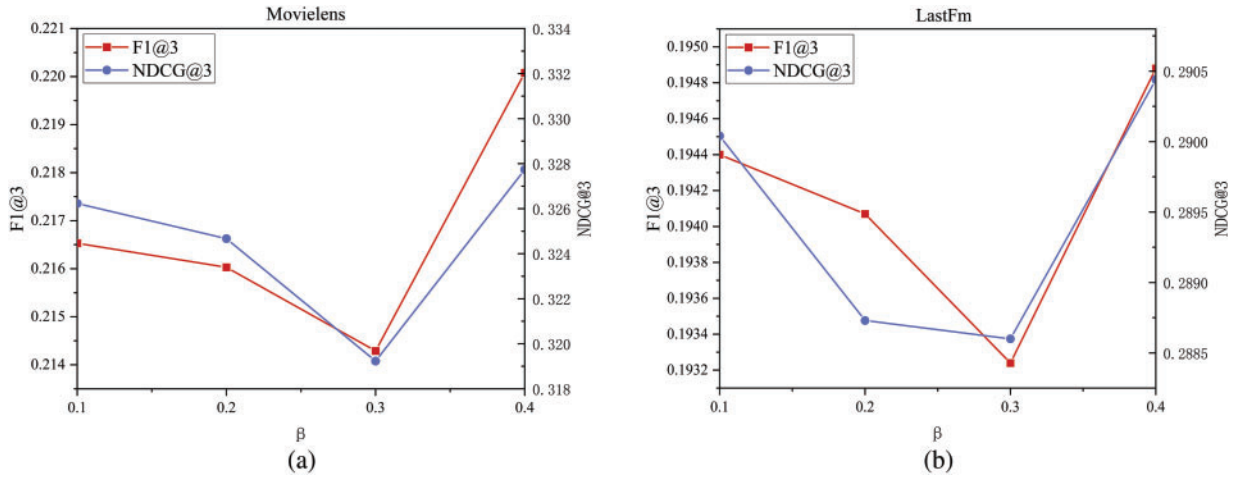

Figure 3: Effect of β on recommendation performance

4.6.2 Hard Weighting Scheme

For the hard weighting scheme, the primary optimized parameter is γ, which determines the threshold for selecting samples based on their loss value, as shown in Fig. 4.

γ Analysis: Experiments were conducted with γ values ranging from 0.1 to 0.3. The results indicated that the optimal performance was achieved when $\gamma = 0.25$. At this value, the model initially prioritized simpler samples and gradually incorporated more complex ones as training progressed, leading to a significant improvement in convergence speed. Lower γ values excluded too many simple samples, slowing the learning process, while higher values introduced overly complex samples too early, resulting in reduced prediction accuracy.

Compared with the soft weighting scheme and hard weighting scheme, we observed that the soft weighting scheme allowed the model to achieve a better balance between exploring diverse samples and focusing on more complex ones. This smooth transition from simple to complex samples helped enhance final recommendation accuracy. On the other hand, the hard weighting scheme, with its more aggressive sample selection, facilitated faster model convergence.

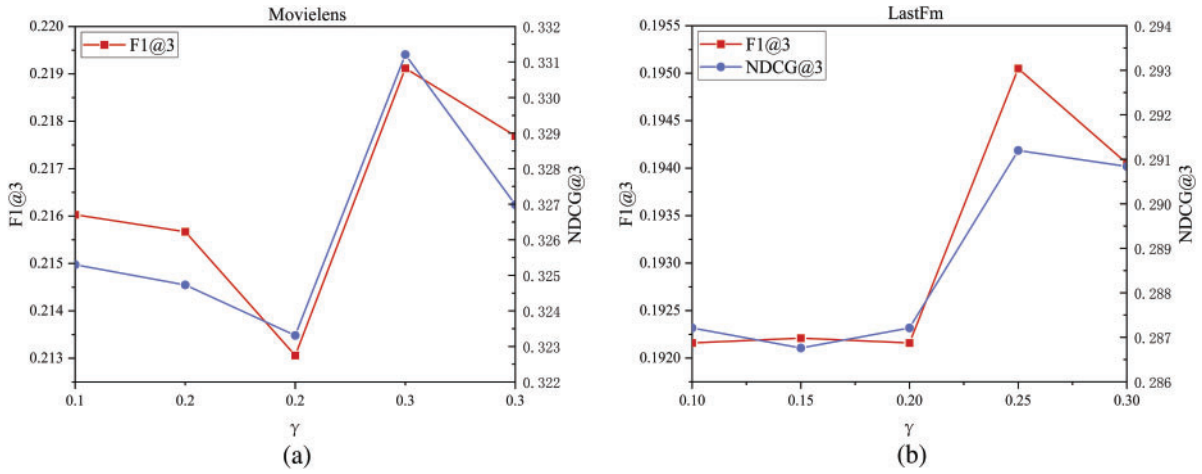

Figure 4: Effect of γ on recommendation performance

4.7 Ablation Study

To verify the effectiveness of the proposed method, we conducted ablation experiments to evaluate the effect of different components of SPAML-H and SPAML-S on model performance as shown in Table 4, focusing particularly on the role of the self-paced adversarial training module.

Table 4: Effect of different components on MovieLens and LastFM datasets

	Movielens		LastFm	
	F1@3	**NDCG@3**	**F1@3**	**NDCG@3**
SPAML-H	0.21813	0.32789	0.19505	0.29120
SPAML-S	0.22008	0.32774	0.19488	0.29044
W/O SPL	0.21278	0.31947	0.19087	0.28643
W/O SPL & PGD	0.20807	0.31443	0.18246	0.27188

The ablation study shows the effectiveness of each component of the proposed method and its contributions to recommendation performance. Relying solely on metric learning to compute the distances between users, items, and tags is insufficient for handling adversarial samples due to its lack of robustness against perturbations. The introduction of adversarial training significantly enhances the model's robustness, allowing it to maintain high recommendation quality under complex scenarios. Furthermore, SPAML-H and SPAML-S, which integrate the self-paced adversarial training strategy, further improve the model's ability to handle both simple and complex adversarial samples. Notably, SPAML-S, employing a soft weighting scheme, achieves the best performance, underscoring the effectiveness of dynamic weight adjustment during training. These findings confirm that the self-paced adversarial training strategy is critical for improving both the robustness and the effectiveness of the proposed method.

4.8 Effectiveness in Adversarial Training

To evaluate the performance of the proposed method under varying types of attacks and attack parameters. Systematic experiments were conducted on the MovieLens and LastFm datasets using SPAML-H and SPAML-S. The best-performing metrics are highlighted in bold, while the second-best metrics are underlined in Table 5.

Table 5: Performance comparison of the proposed method with using varying types of attacks and attack parameters on MovieLens and LastFM datasets

	Movielens		LastFm	
	F1@3	**NDCG@3**	**F1@3**	**NDCG@3**
Ours(H)-FGM [23]	0.21465	0.32212	0.19426	0.28967
Ours(H)-FGSM [48]	0.21501	0.32774	0.19133	0.28616
Ours(H)-PGD(5) [32]	0.21378	0.32272	0.19256	0.28732
Ours(H)-PGD(15)	**0.21817**	0.32377	0.19413	0.29061
Ours(H)-FreeLB [48]	0.21718	0.32654	0.19225	0.28805
Ours(H)-MIM [49]	0.21501	0.32534	0.19182	0.28649
Ours(S)-FGM	0.21241	0.32086	0.19407	0.28949
Ours(S)-FGSM	0.21504	0.32661	0.19228	0.28560
Ours(S)-PGD(5)	0.21769	**0.32911**	0.19365	0.28895
Ours(S)-PGD(15)	0.21690	0.32336	**0.19439**	0.28918
Ours(S)-FreeLB	0.21321	0.32063	0.19387	**0.29302**
Ours(S)-MIM	0.21784	0.32197	0.19412	0.28857

The experimental results demonstrate the effectiveness of the proposed method in enhancing model performance under various adversarial attack types and parameters. On the MovieLens and LastFm datasets, the PGD adversarial training method achieves outstanding results in terms of F1@3 and NDCG@3 metrics, showcasing its ability to counter complex adversarial samples. In comparison, the FGSM adversarial training method delivers relatively average performance, indicating its limited capability in handling high-intensity attacks. FreeLB and MIM adversarial training methods exhibit consistently stable performance across diverse settings, further validating their general adaptability. Notably, SPAML-S consistently outperforms SPAML-H across all attack methods and parameter configurations. This result highlights the effectiveness of the soft weighting strategy in SPAML-S, enabling it to adapt dynamically to adversarial sample maintaining superior performance. Overall, these findings confirm that the proposed adaptive adversarial training strategy significantly enhances both the robustness of the model in diverse and challenging scenarios.

4.9 Effect of Different Embedding Dimensions

We explored the effect of different embedding dimensions k on model performance and recorded the experimental results on two datasets, using F1@3 and NDCG@3 as the primary evaluation metrics, as shown in Fig. 5.

We further analyzed the sensitivity of the SPMLA-H and SPMLA-S models to different embedding dimensions. The results clearly indicate that as k increases, both SPMLA-H and SPMLA-S exhibit improved recommendation performance, specifically reflected in the improvement of F1@3 and NDCG@3 metrics. However, larger embedding dimensions lead to longer training times and higher storage requirements. To address this trade-off, we ultimately selected $k = 64$, which ensures high F1@3 and NDCG@3 scores while maintaining an ideal balance between training efficiency and resource consumption, making it suitable for practical applications. Additionally, as k continues to increase, we observed a decline in F1@3 and NDCG@3 scores, likely due to model overfitting, which negatively impacts recommendation performance.

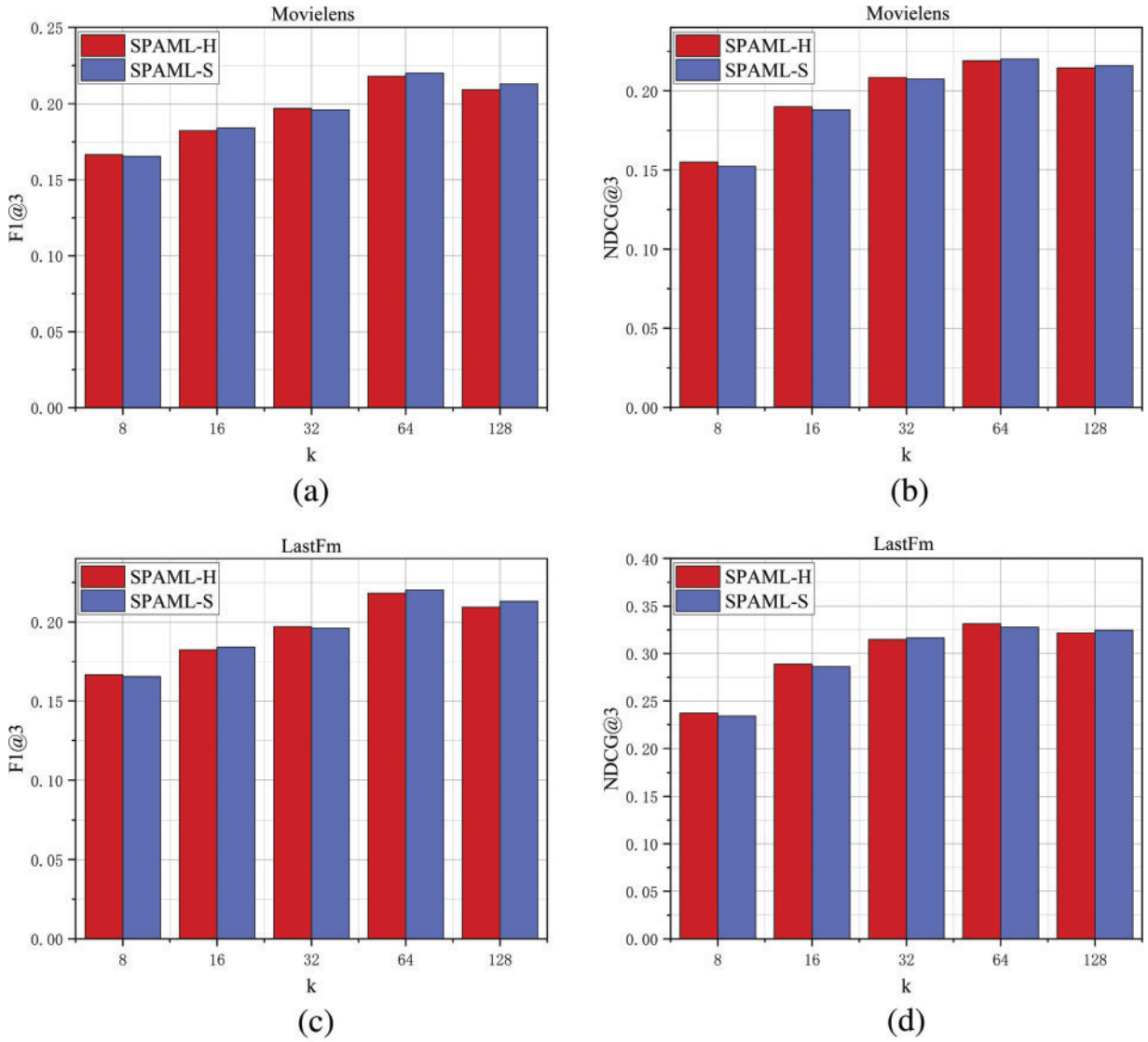

Figure 5: Effect of embedding dimensions on recommendation performance

4.10 Effect of Different Iterations

To evaluate the robustness of the proposed SPAML-H and SPAML-S methods during training, we tested the performance of all methods across different iteration cycles while keeping other parameters at their optimal values. The results are presented in Fig. 6.

Compared to other baseline methods, SPAML-H and SPAML-S consistently demonstrate superior performance across each iteration. On both datasets, the training process for F1@3 and NDCG@3 metrics is notably more stable, which suggests that self-paced adversarial training significantly enhances the robustness of the metric learning model. By continually optimizing the model through adversarial training, more stable and reliable prediction results are achieved, even as iterations increase.

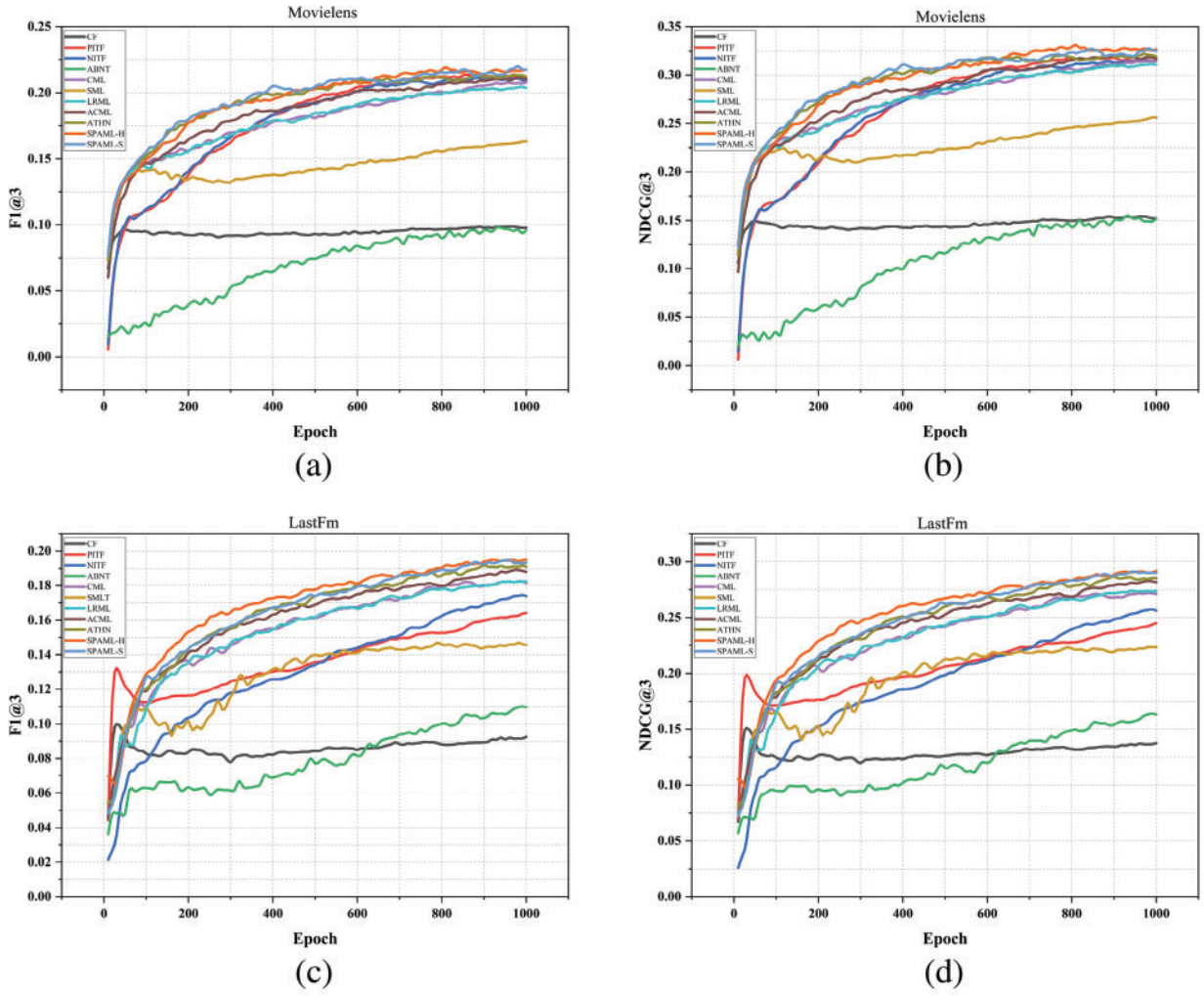

Figure 6: Effect of different iterations on recommendation performance

4.11 Evaluation of Model Robustness with and without Attacks

Tag recommendation systems in real-world applications often encounter unpredictable noise and adversarial perturbations. To evaluate the robustness of the proposed method, experiments were conducted under two scenarios: no-attack and attack. In the no-attack scenario, the dataset remains unchanged. In the attack scenario, random tag replacements simulate adversarial or noisy conditions with three levels of attack intensity: low (10% perturbed), medium (20%), and high (30%).

The experimental results in Table 6 provide a comprehensive evaluation of the model's robustness under no-attack and varying attack scenarios. On the original dataset, the model achieves high F1 and NDCG scores on both MovieLens and LastFm datasets, showcasing its ability to capture intrinsic data features and provide reliable recommendations without external disturbances. Under low-intensity attacks, the model's performance slightly decreases but remains robust, demonstrating strong resistance to minor adversarial perturbations. For medium-intensity attacks, the model shows moderate performance degradation, yet it continues to deliver satisfactory recommendations. Notably, under high-intensity attacks, the model maintains a stable performance, particularly with the SPAML-S variant outperforming SPAML-H, indicating its enhanced resistance in highly challenging scenarios. These results confirm the critical role of the

proposed adaptive adversarial training strategy in mitigating the impact of adversarial perturbations, thereby enhancing both robustness and stability across diverse scenarios.

Table 6: Performance comparison of the proposed method with and without Attacks on MovieLens and LastFM datasets

	Movielens		LastFm	
	F1@3	**NDCG@3**	**F1@3**	**NDCG@3**
SPAML-H	0.21813	0.32789	0.19505	0.29120
SPAML-S	0.22008	0.32774	0.19488	0.29044
SPAML-H(low)	0.20793	0.31219	0.18893	0.28410
SPAML-H(medium)	0.19693	0.29873	0.18353	0.27420
SPAML-H(high)	0.18738	0.28605	0.17705	0.26949
SPAML-S(low)	0.20654	0.31093	0.18960	0.28361
SPAML-S(medium)	0.1996	0.29971	0.18205	0.27595
SPAML-S(high)	0.18745	0.28491	0.17572	0.26724

5 Discussion

The traditional tag recommendation methods primarily rely on inner product to model the similarity relationships between users, items, and tags. However, the inner product does not satisfy the triangle inequality, leading to suboptimal recommendation performance. In contrast, SPAML improves recommendation quality and accuracy by using distance metrics, replacing the inner product with distance to model the relationships between users, items, and tags. While metric learning-based tag recommendation methods perform well in capturing the distance similarity between items and users, they are vulnerable to noisy data. These methods often struggle to extract deep information from the data, resulting in poor generalization ability and robustness. In contrast, SPAML combines self-paced adversarial training, which enhances the method's robustness in handling noisy data and adversarial attacks. Adversarially trained tag recommendation methods typically use a fixed adversarial training strategy, which may lead to overfitting or difficulty in adapting to diverse adversarial perturbations. In contrast, SPAML's self-paced adversarial training mechanism dynamically adjusts the weight of adversarial samples, effectively handling adversarial examples of varying complexity.

The practical significance of SPAML lies in its benefits for users, businesses, and recommendation systems. For users, the proposed method reduces the errors in recommendations caused by highly similar tags, thereby improving recommendation accuracy and enhancing user satisfaction and experience. For businesses, the method effectively mitigates the impact of malicious inputs or noise, providing reliable product-related tags to users and ensuring the stability and reliability of recommendation results. This, in turn, increases user retention and revenue. For recommendation systems, the method offers new insights into handling recommendation scenarios with noisy data and adversarial attacks. The proposed algorithm demonstrates a certain level of generalizability and can be effectively applied to the Delicious dataset. Delicious is a social bookmarking dataset containing users' tag annotations for web. Its structure is highly similar to that of the MovieLens and LastFM datasets, as all three include interaction information among users, items, and tags. Future research could focus on optimizing the model's computational efficiency, developing lightweight adversarial training mechanisms, or more efficient adversarial sample selection strategies to accommodate real-time deployment in large-scale systems.

6 Conclusions

In this paper, we propose SPAML, a self-paced adversarial metric learning method. SPAML captures deep distance relationships between normal samples and dynamically adjusts the weights of adversarial samples using hard and soft weighting schemes, enabling a gradual progression from simple to complex adversarial examples. The joint optimization of the metric learning and self-paced adversarial training loss functions fosters robust defenses and stable predictions, enhancing performance in tag recommendation tasks. Compared to traditional adversarial training methods, SPAML not only improves adversarial robustness but also enhances generalization, reducing overfitting. However, it has certain limitations in terms of training complexity and stability. The generation and selection of adversarial samples introduce additional complexity during training. Furthermore, adversarial training increases computational overhead, especially on high-dimensional and sparse datasets typical of recommendation systems, resulting in longer training times and higher computational costs.

Future research could focus on optimizing adversarial sample generation mechanisms, improving training efficiency, and exploring the model's performance in more complex scenarios. The application of hyperbolic space in recommendation systems has shown significant potential. Constructing hyperbolic distance models among users, items, and tags, and investigating the application of adaptive adversarial training in this geometric structure could further enhance model performance and adaptability.

Acknowledgement: The authors would like to thank all the reviewers who participated in the review.

Funding Statement: This work was partially supported by the Key Research and Development Program of Zhejiang Province (No. 2024C01071), the Natural Science Foundation of Zhejiang Province (No. LQ15F030006).

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: Zhengshun Fei, Jianxin Chen; data collection: Gui Chen; analysis and interpretation of results: Xinjian Xiang; draft manuscript preparation: Zhengshun Fei, Jianxin Chen, Gui Chen, Xinjian Xiang. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: All data and code generated and used in this study are available upon reasonable request from the corresponding author. Our study is clearly documented and accessible for replication by others.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Ko H, Lee S, Park Y, Choi A. A survey of recommendation systems: recommendation models, techniques, and application fields. Electronics. 2022;11(1):141. doi:10.3390/electronics11010141.
2. Chen J, Dong H, Wang X, Feng F, Wang M, He X. Bias and debias in recommender system: a survey and future directions. ACM Trans Inf Syst. 2023 Feb;41(3):1–39. doi:10.1145/3564284.
3. Koren Y, Bell R, Volinsky C. Matrix factorization techniques for recommender systems. Computer. 2009;42(8):30–7.
4. Wu L, He X, Wang X, Zhang K, Wang M. A survey on accuracy-oriented neural recommendation: from collaborative filtering to information-rich recommendation. IEEE Trans Knowl Data Eng. 2023;35(5):4425–45. doi:10.1109/TKDE.2022.3145690.
5. Zang T, Zhu Y, Liu H, Zhang R, Yu J. A survey on cross-domain recommendation: taxonomies, methods, and future directions. ACM Trans Inf Syst. 2022 Dec;41(2):1–39. doi:10.1145/3548455.

6. Rendle S, Schmidt-Thieme L. Pairwise interaction tensor factorization for personalized tag recommendation. In: Proceedings of the Third ACM International Conference on Web Search and Data Mining, ser. WSDM '10; 2010; New York, NY, USA: Association for Computing Machinery. p. 81–90. doi:10.1145/1718487.1718498.
7. Fang X, Pan R, Cao G, He X, Dai W. Personalized tag recommendation through nonlinear tensor factorization using gaussian kernel. Proc AAAI Conf Artif Intell. 2015 Feb;29(1):1. doi:10.1609/aaai.v29i1.9214.
8. Yuan J, Jin Y, Liu W, Wang X. Attention-based neural tag recommendation. In: Li G, Yang J, Gama J, Natwichai J, Tong Y, editors. Database systems for advanced applications. Cham: Springer International Publishing; 2019. p. 350–65. doi:10.1007/978-3-030-18579-4_21.
9. Huai Z, Yang Y, Zhang M, Zhang Z, Li Y, Wu W. M2GNN: metapath and multi-interest aggregated graph neural network for tag-based cross-domain recommendation. In: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, ser. SIGIR '23; 2023; New York, NY, USA: Association for Computing Machinery. p. 1468–77. doi:10.1145/3539618.3591720.
10. Zuo Y, Liu S, Zhou Y, Liu H. Tral: a tag-aware recommendation algorithm based on attention learning. Appl Sci. 2023;13(2):814. doi:10.3390/app13020814.
11. Liu N, Zhao J. Recommendation system based on deep sentiment analysis and matrix factorization. IEEE Access. 2023;11(6):16994–17001. doi:10.1109/ACCESS.2023.3246060.
12. Lu S, Xu P, Liu B, Sun H, Jing L, Yu J. Retrieval augmented cross-modal tag recommendation in software Q&A sites. arXiv:2402.03635. 2024.
13. Li L, Wang P, Zheng X, Xie Q, Tao X, Velásquez JD. Dual-interactive fusion for code-mixed deep representation learning in tag recommendation. Inf Fusion. 2023;99:101862. doi:10.1016/j.inffus.2023.101862.
14. Shrivastava A, Li P. Asymmetric LSH (ALSH) for sublinear time maximum inner product search (MIPS). In: Advances in neural information processing systems. Montreal, Canada: Curran Associates, Inc.; 2014. Vol. 3, p. 2321–9.
15. Ge Y, Chen D, Li H. Mutual mean-teaching pseudo label refinery for unsupervised domain adaptation on person re-identification. arXiv:2001.01526. 2020.
16. Ye M, Shen J, Lin G, Xiang T, Shao L, Hoi SC. Deep learning for person re-identification: a survey and outlook. IEEE Trans Pattern Anal Mach Intell. 2021;44(6):2872–93.
17. Tay Y, Anh Tuan L, Hui SC. Latent relational metric learning via memory-based attention for collaborative ranking. In: Proceedings of the 2018 World Wide Web Conference, ser. WWW '18; 2018; Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee. p. 729–39. doi:10.1145/3178876.3186154.
18. Vinh Tran L, Tay Y, Zhang S, Cong G, Li X. HyperML: a boosting metric learning approach in hyperbolic space for recommender systems. In: Proceedings of the 13th International Conference on Web Search and Data Mining, ser. WSDM '20; 2020; New York, NY, USA: Association for Computing Machinery. p. 609–17. doi:10.1145/3336191.3371850.
19. Liu F, Cheng Z, Sun C, Wang Y, Nie L, Kankanhalli M. User diverse preference modeling by multimodal attentive metric learning. In: Proceedings of the 27th ACM International Conference on Multimedia, ser. MM '19; 2019; New York, NY, USA: Association for Computing Machinery. p. 1526–34. doi:10.1145/3343031.3350953.
20. Wei T, Ma J, Chow TW. Collaborative residual metric learning. In: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, ser. SIGIR '23; 2023; New York, NY, USA: Association for Computing Machinery. p. 1107–16. doi:10.1145/3539618.3591649.
21. Li M, Zhang S, Zhu F, Qian W, Zang L, Han J, et al. Symmetric metric learning with adaptive margin for recommendation. Proc AAAI Conf Artif Intell. 2020 Apr;34(4):4634–41. doi:10.1609/aaai.v34i04.5894.
22. Ma C, Ma L, Zhang Y, Tang R, Liu X, Coates M. Probabilistic metric learning with adaptive margin for top-k recommendation. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, ser. KDD'20; 2020; New York, NY, USA. pp. 1036–44.
23. Goodfellow IJ, Shlens J, Szegedy C. Explaining and harnessing adversarial examples. arXiv:1412.6572.2015. 2025.
24. Wong E, Rice L, Kolter JZ. Fast is better than free: revisiting adversarial training. arXiv:2001.03994.2020. 2020.

25. Wang X, Chen Y, Zhu W. A survey on curriculum learning. IEEE Trans Pattern Anal Mach Intell. 2022;44(9):4555–76. doi:10.1109/TPAMI.2021.3069908.
26. Cai Q-Z, Du M, Liu C, Song D. Curriculum adversarial training. arXiv:1805.04807. 2018.
27. Wang Y, Ma X, Bailey J, Yi J, Zhou B, Gu Q. On the convergence and robustness of adversarial training. arXiv:2112.08304. 2022.
28. Zhang J, Xu X, Han B, Niu G, Cui L, Sugiyama M, et al. Attacks which do not kill training make adversarial learning stronger. In: III HD, Singh A, editors. In: Proceedings of the 37th International Conference on Machine Learning, PMLR; 2020 Jul 13–18; Vienna, Austria. p. 11278–87.
29. Zhao K, Liu Z, Li J, Zhao B, Jia Z, Shao H. Self-paced decentralized federated transfer framework for rotating machinery fault diagnosis with multiple domains. Mech Syst Signal Process. 2024;211:111258. doi:10.1016/j.ymssp.2024.111258.
30. Yanming L, Jinglei L. Leveraging self-paced learning and deep sparse embedding for image clustering. Neural Comput Appl. 2024;36:5135–51. doi:10.1007/s00521-023-09335-w.
31. Hsieh C-K, Yang L, Cui Y, Lin T-Y, Belongie S, Estrin D. Collaborative metric learning. In: Proceedings of the 26th International Conference on World Wide Web, WWW '17; 2017; Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee. p. 193–201. doi:10.1145/3038912.3052639.
32. Mądry A, Makelov A, Schmidt L, Tsipras D, Vladu A. Towards deep learning models resistant to adversarial attacks. Statistics. 2017;1050(9):1205–32.
33. Ruijin Xue QW, Feng S. Improving diversity with multi-loss adversarial training in personalized news recommendation. Comput Mater Contin. 2024;80(2):3107–22. doi:10.32604/cmc.2024.052600.
34. Zheng Y, Zhang R, Mao Y. Regularizing neural networks via adversarial model perturbation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2021 Jun; Nashville, TN, USA. p. 8156–65.
35. Jia X, Chen Y, Mao X, Duan R, Gu J, Zhang R, et al. Revisiting and exploring efficient fast adversarial training via law: lipschitz regularization and auto weight averaging. IEEE Trans Inf Forensics Secur. 2024:1. doi:10.1109/TIFS.2024.3420128.
36. Kim WJ, Cho Y, Jung J, Yoon S-E. Feature separation and recalibration for adversarial robustness. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2023 Jun; Vancouver, BC, Canada. p. 8183–92.
37. Li H, Pan SJ, Wang S, Kot AC. Domain generalization with adversarial feature learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2018 Jun; Salt Lake City, UT, USA.
38. Bashivan P, Bayat R, Ibrahim A, Ahuja K, Faramarzi M, Laleh T, et al. Adversarial feature desensitization. In: Ranzato M, Beygelzimer A, Dauphin Y, Liang P, Vaughan JW, editors. Advances in neural information processing systems. Vancouver, BC, Canada: Curran Associates, Inc.; 2021. Vol. 34, p. 10665–77.
39. Shafahi A, Najibi M, Ghiasi MA, Xu Z, Dickerson J, Studer C, et al. Adversarial training for free!. In: Wallach H, Larochelle H, Beygelzimer A, d' Alché-Buc F, Fox E, Garnett R, editors. Advances in neural information processing systems. Vancouver Convention Center, Vancouver, BC, Canada: Curran Associates, Inc.; 2019. Vol. 32.
40. Duan Y, Zheng W, Lin X, Lu J, Zhou J. Deep adversarial metric learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2018 Jun; Salt Lake City, UT, USA.
41. Chen S, Gong C, Yang J, Li X, Wei Y, Li J. Adversarial metric learning. arXiv:1802.03170. 2018.
42. Wang Z, Wang Y, Dong B, Pracheta S, Hamlen K, Khan L. Adaptive margin based deep adversarial metric learning. In: 2020 IEEE 6th International Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing, (HPSC) and IEEE International Conference on Intelligent Data and Security (IDS); 2020; Baltimore, MD, USA. p. 100–8. doi:10.1109/BigDataSecurity-HPSC-IDS49724.2020.00028.
43. Mao C, Zhong Z, Yang J, Vondrick C, Ray B. Metric learning for adversarial robustness. In: Wallach H, Larochelle H, Beygelzimer A, d' Alché-Buc F, Fox E, Garnett R, editors. Advances in neural information processing systems. Vancouver Convention Center, Vancouver, BC, Canada: Curran Associates, Inc.; 2019. Vol. 32.

44. Li P, Brost B, Tuzhilin A. Adversarial learning for cross domain recommendations. ACM Trans Intell Syst Technol. 2022 Nov;14(1):1–25. doi:10.1145/3548776.
45. Wang Y, Zou D, Yi J, Bailey J, Ma X, Gu Q. Improving adversarial robustness requires revisiting misclassified examples. International Conference on Learning Representations. New Orleans, LA, USA; 2019.
46. He L, Ai Q, Yang X, Ren Y, Wang Q, Xu Z. Boosting adversarial robustness via self-paced adversarial training. Neural Netw. 2023;167(1):706–14. doi:10.1016/j.neunet.2023.08.063.
47. Wang J, Chen G, Xin K, Fei Z. Metric learning with adversarial hard negative samples for tag recommendation. J Supercomput. 2024;80(14):21475–507. doi:10.1007/s11227-024-06274-8.
48. Zhu C, Cheng Y, Gan Z, Sun S, Goldstein T, Liu J. FreeLB: enhanced adversarial training for natural language understanding. 2020. doi:10.48550/arXiv.1909.11764.
49. Dong Y, Liao F, Pang T, Su H, Zhu J, Hu X, et al. Boosting adversarial attacks with momentum. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2018; Salt Lake City, UT, USA. p. 9185–93.