Adversarial Training Towards Robust Multimedia Recommender System

Jinhui Tang, *Senior Member, IEEE,* Xiaoyu Du, Xiangnan He, Fajie Yuan, Qi Tian, *Fellow, IEEE,* and Tat-Seng Chua

Abstract—With the prevalence of multimedia content on the Web, developing recommender solutions that can effectively leverage the rich signal in multimedia data is in urgent need. Owing to the success of deep neural networks in representation learning, recent advance on multimedia recommendation has largely focused on exploring deep learning methods to improve the recommendation accuracy. To date, however, there has been little effort to investigate the robustness of multimedia representation and its impact on the performance of multimedia recommendation.

In this paper, we shed light on the robustness of multimedia recommender system. Using the state-of-the-art recommendation framework and deep image features, we demonstrate that the overall system is not robust, such that a small (but purposeful) perturbation on the input image will severely decrease the recommendation accuracy. This implies the possible weakness of multimedia recommender system in predicting user preference, and more importantly, the potential of improvement by enhancing its robustness. To this end, we propose a novel solution named *Adversarial Multimedia Recommendation* (AMR), which can lead to a more robust multimedia recommender model by using adversarial learning. The idea is to train the model to defend an adversary, which adds perturbations to the target image with the purpose of decreasing the model's accuracy. We conduct experiments on two representative multimedia recommendation tasks, namely, image recommendation and visually-aware product recommendation. Extensive results verify the positive effect of adversarial learning and demonstrate the effectiveness of our AMR method. Source codes are available in https://github.com/duxy-me/AMR.

Index Terms—Multimedia Recommendation, Adversarial Learning, Personalized Ranking, Collaborative Filtering.

1 Introduction

Recommender system plays a central role in user-centric online services, such as E-commerce, media-sharing, and social networking sites. By providing personalized content suggestions to users, recommender system not only can alleviate the information overload issue and improve user experience, but also can increase the profit for content providers through increasing the traffic. Thus many research efforts have been devoted to advance recommendation technologies, which have become an attractive topic of research in both academia and industry in the recent decade [1], [2], [3], [4]. On the other hand, multimedia data becomes prevalent on the current Web. For example, products are usually associated with images to attract customers in E-commerce sites [5], and users usually post images or micro-videos to interact with their friends in social media sites [6], [7]. Such multimedia content contains rich visually-relevant signal that can reveal user preference, providing opportunities to improve recommender systems that are typically based on collaborative filtering on user behavior data only [8], [9].

Early multimedia recommendation works have largely employed annotated tags [10], [11] or low-level representations [12] such as color-based features and texture features like SFIT, to capture the semantics of multimedia content. Owing to the success of deep neural networks (DNNs) in learning representations [13], recent advance on multimedia recommendation has shifted to integrating deep multimedia features into recommender model [14], [15], [16], [17], [18]. For example, in image-based recommendation, a typical paradigm is to project the CNN features of image into the same latent space as that of users [16], [19], or simultaneously learn image representation and recommender model [20].

Although the use of DNNs to learn multimedia representation leads to better recommendation performance than manually crafted features, we argue that a possible downside is that the overall system becomes **less robust**. As have shown in several previous works [21], [22], [23], many state-of-the-art DNNs are vulnerable to adversarial attacks. Taking the image classification task as an example, by applying small but intentionally perturbations to well-trained images from the dataset, these DNN models output wrong labels for the images with high confidence. This implies that the image representations learned by DNNs are not robust, which further, may negatively affect downstream applications based on the learned representations.

- *Jinhui Tang is with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, Jiangsu, China, 210094.*
  E-mail: jinhuitang@njust.edu.cn
- *Xiaoyu Du is with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, Sichuan, China, 610054.*
  E-mail: duxy.me@gmail.com
- *Xiangnan He and Tat-Seng Chua are with the School of Computing, National University of Singapore, Singapore, 117417.*
  E-mail: xiangnanhe@gmail.com, dcscts@nus.edu.sg
- *Fajie Yuan is with the School of Computing Science, University of Glasgow, Glasgow G12 8RZ, UK.*
  E-mail: f.yuan.1@research.gla.ac.uk
- *Qi Tian is with the Department of Computer Science, the University of Texas at San Antonio, San Antonio, TX 78249, USA.*
  E-mail: qi.tian@utsa.edu

Xiangnan He is the corresponding author. *Manuscript received Aug 1, 2018.*

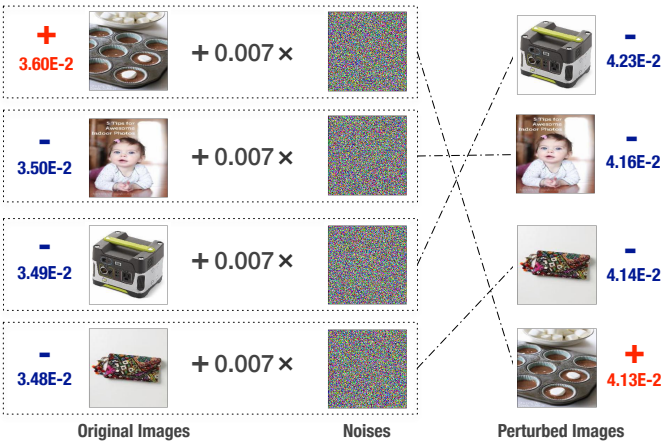


Fig. 1: An example on how small perturbations on images would have a profound impact on the recommendation results. We sampled a user, one interacted image (in red "+") and three non-interacted images (in blue "-") by the user. The number besides each image is the ranking score generated by VBPR [19] before (left) and after (right) perturbations. By adding small perturbations with the scale of $\epsilon = 0.007$, the positive image is ranked much lower than before, even though the difference of perturbed images can be hardly perceived by human. Here the perturbations are generated by the fast gradient sign method [22].

Figure 1 shows an illustrative example on how the lack of robustness affects the recommendation results. We first trained the Visual Bayesian Personalized Ranking (VBPR) method [19] on a Pinterest dataset; VBPR is a state-of-the-art visually-aware recommendation method, and we used the ResNet-50 [13] to extract image features for it. We then sampled a user u, showing her interacted image in the testing set (*i.e.,* the top-left image with sign "+") and three non-interacted images (*i.e.,* the bottom-left three images with sign "-"). From the prediction scores of VBPR (*i.e.,* the numbers beside images), we can see that, originally, VBPR successfully ranks the positive image higher than other negative images for the user. However, after applying adversarial perturbations to these images, even though the perturbation scale is very small ($\epsilon = 0.007$) *s.t.* human can hardly perceive the change on the perturbed images, VBPR outputs very different prediction scores and fails to rank the positive image higher than other negative images. This example demonstrates that adversarial perturbations for DNNs would have a profound impact on the downstream recommender model, making the model less robust and weak in generalizing to unseen predictions.

In this paper, we enhance the robustness of multimedia recommender system and thus its generalization performance by performing adversarial learning [24]. With VBPR as the main recommender model, we introduce an adversary which adds perturbations on multimedia content with the aim of maximizing the VBPR loss function. We term our method as *Adversarial Multimedia Recommendation* (AMR), which can be interpreted as playing a minimax game — the perturbations are learned towards maximizing the VBPR loss, whereas the model parameters are learned towards minimizing both the VPBR loss and the adversary's loss. Through this way, we can enhance model robustness to adversarial perturbations on the multimedia content, such that the perturbations have a smaller impact on the model's prediction. To verify our proposal, we conduct experiments on two public datasets, namely, the Pinterest image data [16] and Amazon product data [19]. Empirical results demonstrate the positive effect of adversarial learning and the effectiveness of our AMR method for multimedia recommendation.

We summarize the main contributions of this work as follows.

1) This is the first work to emphasize the vulnerability issue of state-of-the-art multimedia recommender systems due to the use of DNNs for feature learning.
2) A novel method is proposed to train a more robust and effective recommender model by using the recent developments on adversarial learning.
3) Extensive experiments are conducted on two representative multimedia recommendation tasks of personalized image recommendation and visually-aware product recommendation to verify our method.

The remainder of the paper is organized as follows. We first provide some preliminaries in Section 2, and then elaborate our proposed method in Section 3. We present experimental results in Section 4 and review related literature in Section 5. Finally, we conclude this paper and discuss future directions in Section 6.

2 Preliminaries

This section provides some technical background to multimedia recommendation. We first recapitulate the Latent Factor Model (LFM), which is the most widely used recommender model in literature [25]. We then introduce the Visual Bayesian Personalized Ranking (VBPR) [19], which is a state-of-the-art method for multimedia recommendation, and we use it as AMR's building block.

2.1 Latent Factor Model

The key of recommendation is to estimate the preference of a user on an item. The paradigm of LFM is to describe a user (and an item) as a vector of latent factors, *a.k.a. latent vector*; then the preference score is estimated as the inner product of the user latent vector and item latent vector. Formally, let u denote a user, i denote an item, and $\hat{y}_{ui}$ denote the estimated preference score of u on i. Then the predictive model of LFM can be abstracted as:

$$\hat{y}_{ui} =< f_U(u), f_I(i) >, \tag{1}$$

where f_U denotes the function that projects a user to the latent space, *i.e.,* $f_U(u)$ denotes the latent vector for user u; similar semantics apply to f_I, the notation of item side.

For a LFM, the design of function f_U and f_I plays a crucial role on its performance, whereas the design is also subjected to the availability of the features to describe a user and an item. In the simplest case, when only the ID information is available, a common choice is to directly associate a user (and an item) with a vector, *i.e.,* $f_U(u) = \mathbf{p}_u$ and $f_I(i) = \mathbf{q}_i$, where $\mathbf{p}_u \in \mathbb{R}^K$ and $\mathbf{q}_i \in \mathbb{R}^K$ are also called

as the *embedding vector* for user u and item i, respectively, and K denotes the embedding size. This instantiation is known as the matrix factorization (MF) model [25], a simple yet effective model for the collaborative filtering task.

Targeting at multimedia recommendation, f_I is typically designed to incorporate content-based features, so as to leverage the visual signal of multimedia item. For example, Geng *et al.* [16] defines it as $f_I(i) = \mathbf{E}\mathbf{c}_i$, where $\mathbf{c}_i \in \mathbb{R}^{4096}$ denotes the deep image features extracted by AlexNet [26], and $\mathbf{E} \in \mathbb{R}^{K \times 4096}$ transforms the image features to the latent space of LFM. A side benefit of such content-based modeling is that the item cold-start issue can be alleviated, since for out-of-sample items, we can still obtain a rather reliable latent vector from its content features. Besides this straightforward way to incorporate multimedia content, other more complicated operations have also been developed. For example, the Attentive Collaborative Filtering (ACF) model [14] uses an attention network to discriminate the importance of different components of a multimedia item, such as the regions in an image and frames of a video.

Owing to the strong generalization ability of LFM in predicting unseen user-item interactions, LFM is recognized as the most effective model for personalized recommendation [14]. As such, we build our adversarial recommendation method upon LFM, more specifically VBPR — an instantiation of LFM for multimedia recommendation. Next, we describe the VBPR method.

2.2 Visual Bayesian Personalized Ranking

It is arguable that a user would not buy a new clothing product from Amazon without seeing it in person, so the visual appearance of items plays an important role in user preference prediction. VBPR is designed to incorporate such visual signal into the learning of user preference from implicit feedback [19]. To be specific, its predictive model is formulated as:

$$\hat{y}_{ui} = \mathbf{p}_u^T \mathbf{q}_i + \mathbf{h}_u^T (\mathbf{E}\mathbf{c}_i), \tag{2}$$

where the first term $\mathbf{p}_u^T \mathbf{q}_i$ is same as MF to model the collaborative filtering effect, and the second term $\mathbf{h}_u^T (\mathbf{E}\mathbf{c}_i)$ models user preference based on the item's image. Specifically, $\mathbf{p}_u \in \mathbb{R}^K$ ($\mathbf{q}_i \in \mathbb{R}^K$) denotes the ID embedding for user u (item i), $\mathbf{h}_u \in \mathbb{R}^K$ is u's embedding in the image latent space, $\mathbf{c}_i \in \mathbb{R}^D$ denotes the visual feature vector for item i (which is extracted by AlexNet), and $\mathbf{E} \in \mathbb{R}^{K \times D}$ converts the visual feature vector to latent space. The K is a hyper-parameter and the D is 4096 if using AlexNet. We can interpret this model as a LFM by defining $f_U(u) = [\mathbf{p}_u, \mathbf{h}_u]$ and $f_I(i) = [\mathbf{q}_i, \mathbf{E}\mathbf{c}_i]$, where $[\cdot, \cdot]$ denotes vector concatenation. Note that in Equation (2), we have only included the key terms on the interaction prediction in VBPR and omitted other bias terms for clarity.

To estimate model parameters, VBPR optimizes the BPR pairwise ranking loss [8] to tailor the model for implicit interaction data such as purchases and clicks. The assumption is that interacted user-item pairs should be scored higher than the non-interacted pairs by the model. To implement this assumption, for each observed interaction (u, i), BPR maximizes the margin between it and its unobserved counterparts. The objective function to minimize is:

$$L_{BPR} = \sum_{(u,i,j) \in \mathcal{D}} - \ln \sigma(\hat{y}_{ui} - \hat{y}_{uj}) + \beta ||\Theta||^2, \tag{3}$$

where $\sigma(\cdot)$ is the sigmoid function, β controls the strength of L_2 regularization on model parameters to prevent overfitting. The set $D = \{(u, i, j) | u \in \mathcal{U}, i \in \mathcal{I}_u^+, j \in \mathcal{I} \setminus \mathcal{I}_u^+\}$ denotes all pairwise training instances, where $\mathcal{U}$, $\mathcal{I}$, and $\mathcal{I}_u^+$ denote all users, items, and the interacted items of user u. To handle the sheer number of pairwise training instances, Rendle *et al.* [8] advocate the use of stochastic gradient descent (SGD) for optimization, which is much less costly and converges faster than batch gradient descent.

2.2.1 Vulnerability of VBPR

Despite a sound solution for multimedia recommendation, we argue that VBPR is not robust in predicting user preference. As demonstrated in Figure 1, even small pixel-level perturbations on image candidates can yield large changes on the ranking of the candidates, which is out of expectation. Note that an image i is converted to feature vector $\mathbf{c}_i$ by DNN and the predictive model uses $\mathbf{c}_i$ to predict user preference on the image (i.e., the $\mathbf{h}_u^T (\mathbf{E}\mathbf{c}_i)$ term). As such, it implies that two possibilities for the vulnerability of VBPR: 1) the small pixel-level changes result in large change on $\mathbf{c}_i$, which subsequently leads to large change on the prediction value, and 2) the small pixel-level changes result in small changes on $\mathbf{c}_i$, but even small fluctuations on $\mathbf{c}_i$ can significantly change the prediction value.

It is worth noting that both possibilities could be valid (e.g., exist for different instances) and can be supported by existing works. For example, Goodfellow *et al.* [22] show that many DNN models are not robust to pixel-level perturbations (which provides evidence for the first possibility), and He *et al.* [27] show that the MF model is not robust to purposeful perturbations on user and item embeddings (which provides evidence for the second possibility). Regardless of which exact reason, it points to the weak generalization ability of the overall multimedia recommender system — if we imagine the prediction function as a curve in high-dimensional space, we can deduce that the curve is not smooth and has big fluctuations at many points. We believe that the vulnerability issue also exists for other deep feature-based multimedia recommendation methods, if no special action is taken to address the issue in the method. In this work, we address this universal issue in multimedia recommender systems by performing adversarial learning, which to our knowledge has not been explored before.

3 Adversarial Multimedia Recommendation

This section elaborates our proposed method. We first present the predictive model, followed by the adversarial loss function, optimizing which can lead to a more robust recommender model. Lastly, we present the optimization algorithm.

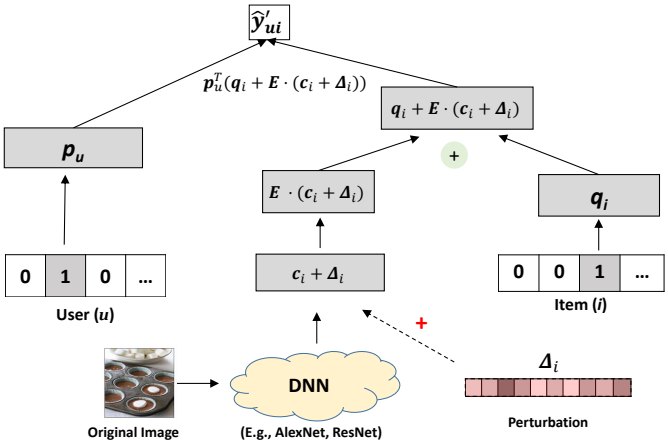


Fig. 2: An illustration of the predictive model with perturbation Δ_i, which is enforced on the image's feature vector extracted by DNN.

3.1 Predictive Model

Note that the focus of this work is to train robust models for multimedia recommendation, rather than developing new predictive models. As such, we simply apply the model of VBPR and make slight adjustments on it:

$$\hat{y}_{ui} = \mathbf{p}_u^T(\mathbf{q}_i + \mathbf{E} \cdot \mathbf{c}_i), \tag{4}$$

where $\mathbf{p}_u \in \mathbb{R}^K$, $\mathbf{q}_i \in \mathbb{R}^K$, $\mathbf{E} \in \mathbb{R}^{K \times D}$ and $\mathbf{c}_i \in \mathbb{R}^D$ have the same meaning as that in Equation (2). The difference of this visually-aware recommender model with VBPR is that it associates each user with one embedding vector $\mathbf{p}_u$ only, while in VBPR each user has two embedding vectors $\mathbf{p}_u$ and $\mathbf{h}_u$. This simplification is just to ensure a fair comparison with the conventional MF model when the embedding size K is set as a same number (*i.e.*, making the models have the same representation ability). Moreover, we have experimented with both ways of user embedding, and did not observe significant difference between them.

3.2 Objective Function

Several recent efforts have shown that adversarial training can improve the robustness of machine learning models [22], [27], [28]. Inspired by their success, we develop adversarial training method to improve multimedia recommender model. The basic ideas are two-fold: 1) constructing an adversary that degrades model performance by adding perturbations on model inputs (and/or parameters), and meanwhile 2) training the model to perform well under the affect of adversary. In what follows, we describe the two ingredients of AMR's training objective function, namely, how to construct the adversary and how to learn model parameters.

1. Adversary Construction. The goal of the constructed adversary is to decrease the model's performance as much as possible. Typically, additive perturbations are applied to either model inputs [22] or parameters [27]. To address the vulnerability issue illustrated in Figure 1, an intuitive solution is to apply perturbations to model inputs, *i.e.*, the raw pixels of the image, since the unexpected change on ranking result is caused by the perturbations on image pixels. Through this way, training the model to be robust to adversarial perturbations can increase the robustness of both the DNN (that extracts image deep features) and LFM (that predicts user preference). However, this solution is difficult to implement due to two practical reasons:

First, it requires the whole system to be end-to-end trainable; in other words, the DNN for image feature extraction needs to be updated during the training of recommender model. Since user-item interaction data is sparse by nature and the DNN usually has many parameters, it may easily lead to overfitting issue if we train the DNN simutaneously.

Second, it leads to a much higher learning complexity. Given a training instance (u, i), the recommender model part only needs to update two embedding vectors ($\mathbf{p}_u$ and $\mathbf{q}_i$) and the feature transformation matrix $\mathbf{E}$, whereas the DNN model needs to update the whole deep network, for which the parameters are several magnitudes larger. Moreover, to update the perturbations, we need to back-propagate the gradient through the DNN, which is also very time-consuming.

To avoid the difficulties in applying pixel-level perturbations, we instead propose to apply perturbations to the image's deep feature vector, *i.e.*, $\mathbf{c}_i$. To be specific, the perturbed model is formulated as:

$$\hat{y}'_{ui} = \mathbf{p}_u^T(\mathbf{q}_i + \mathbf{E} \cdot (\mathbf{c}_i + \Delta_i)), \tag{5}$$

where Δ_i denotes the perturbations added on deep image feature vector by the adversary. Figure 2 illustrates the perturbed model. This way of adding perturbations has two implications: 1) the DNN model can only serve as an image feature extractor, which is neither updated nor involved in the adversary construction process, making the learning algorithm more efficient, and 2) adversarial training can't improve the quality of deep image representation $\mathbf{c}_i$, but it can improve the image's representation in MF's latent space (that is $\mathbf{E}\mathbf{c}_i$, since $\mathbf{E}$ is updated by adversarial training towards the aim of being robust).

We now consider how to find optimal perturbations that lead to the largest influence on the model, which are also known as the worst-case perturbations [22]. Since the model is trained to minimize the BPR loss (see Equation (3)), a natural idea is to set an opposite goal for the perturbations — maximizing the BPR loss. Let $\Delta = [\Delta_i] \in \mathbb{R}^{|\mathcal{I}| \times D}$, which denotes the perturbations for all images and the i-th column is Δ_i. We obtain optimal perturbations by maximizing the BPR loss on training data:

$$\begin{aligned} \Delta^* &= \arg\max_{\Delta} L'_{BPR} = \arg\max_{\Delta} \sum_{(u,i,j) \in \mathcal{D}} -\ln \sigma(\hat{y}'_{ui} - \hat{y}'_{uj}), \\ &\text{where} \quad ||\Delta_i|| \leq \epsilon, \quad \text{for} \quad i = 1, ..., |\mathcal{I}|, \end{aligned} \tag{6}$$

where $||\cdot||$ denotes the L_2 norm, and ϵ is a hyper-parameter that controls the magnitude of perturbations. The constraint of $||\Delta_i|| \leq \epsilon$ is to avoid a trivial solution that increases the BPR loss by simply increasing the scale of Δ_i. Note that compared with the orignal BPR loss, we remove the L_2 regularizer on model parameters in this perturbed BPR loss, since the construction of Δ is based on the current values of model parameters, which are irrelevant to Δ and thus can be safely removed.

2. Model Optimization. To make the model less sensitive to the adversarial perturbations, in addition to minimize the original BPR loss, we also minimize the adversary's objective function. Let Θ be the model parameters, which includes $\mathbf{p}_u$ for all users, $\mathbf{q}_i$ for all items, and transformation matrix $\mathbf{E}$. We define the optimization objective for the model as

$$\begin{aligned}\Theta^* &= \arg\min_{\Theta} L_{BPR} + \lambda L'_{BPR}, \\ &= \arg\min_{\Theta} \sum_{(u,i,j)\in\mathcal{D}} -\ln\sigma(\hat{y}_{ui} - \hat{y}_{uj}) - \lambda \ln\sigma(\hat{y}'_{ui} - \hat{y}'_{uj}) \\ &\quad + \beta||\Theta||^2\end{aligned} \tag{7}$$

where λ is a hyper-parameter to control the impact of the adversary on the model optimization. When λ is set to 0, the adversary has no impact on training and the method degrades to VBPR. In this formulation, the adversary's loss L'_{BPR} can be seen as regularizing the model to make it be more robust, thus it is also called as *adversarial regularizer* in literature [27].

To unify the two processes, we formulate it as a minimax objective function. The optimization of model parameters Θ is the minimizing player, and the construction of perturbations Δ is the maximizing player:

$$\begin{aligned}\Theta^*, \Delta^* = \arg\min_{\Theta}\max_{\Delta} L_{BPR}(\Theta) + \lambda L'_{BPR}(\Theta, \Delta), \\ \text{where} \quad ||\Delta_i|| \leq \epsilon, \quad \text{for} \quad i = 1, ..., |\mathcal{I}|.\end{aligned} \tag{8}$$

Compared to VBPR, our AMR has two more hyper-parameters to be specified — ϵ and λ. Both hyper-parameters are crucial to recommendation performance and need to be carefully tuned. Particularly, too large values will make the model robust to adversarial perturbations but at the risk of destroying the training process, while too small values will limit the impact of the adversary and make no improvements on the model's robustness and generalization ability. In the next subsection, we discuss how to optimize the minimax objective function.

3.3 Learning Algorithm

Due to the large number of pairwise training instances in BPR loss, batch gradient descent could be very time consuming and slow to converge [8]. As such, we prioritize the SGD learning algorithm. Algorithm 1 illustrates our devised SGD learning algorithm for AMR. The subproblem to consider in SGD is that given a stochastic training instance (u, i, j), how to optimize parameters related to this instance only (line 4-9):

- For adversary construction (line 4-5), the objective function (maximized) regarding to this instance is:

$$l'_{uij} = -\ln\sigma(\hat{y}'_{ui} - \hat{y}'_{uj}), \text{where } ||\Delta_i|| \leq \epsilon,\ ||\Delta_j|| \leq \epsilon. \tag{9}$$

By **maximizing** the objective function, we obtain the worst-case perturbations Δ_i and Δ_j, which can make the largest change on the BPR loss on the single instance (u, i, j).

- For model parameter learning (line 6-9), the objective function (to be minimized) regarding to this instance is:

$$\begin{aligned}l_{uij} = &-\ln\sigma(\hat{y}_{ui} - \hat{y}_{uj}) - \lambda\ln\sigma(\hat{y}'_{ui} - \hat{y}'_{uj}) \\ &+ \beta(||\mathbf{p}_u||^2 + ||\mathbf{q}_i||^2 + ||\mathbf{E}||^2).\end{aligned} \tag{10}$$

Algorithm 1: SGD learning algorithm for AMR.

Input: Training data $\mathcal{D}$, adversarial noise level ϵ, adversarial regularizer strength λ, L_2 regularizer strength β, and learning rate η;
Output: Model parameters Θ;

1 Initialize Θ from VBPR ;
2 **while** *not converge* **do**
3 Randomly draw an example (u, i, j) from $\mathcal{D}$;
// Learning adversarial perturbations
4 $\Delta_i \leftarrow \epsilon\frac{\Gamma_i}{||\Gamma_i||}$ where $\Gamma_i = \frac{\partial l'_{uij}}{\partial \Delta_i}$;
5 $\Delta_j \leftarrow \epsilon\frac{\Gamma_j}{||\Gamma_j||}$ where $\Gamma_j = \frac{\partial l'_{uij}}{\partial \Delta_j}$;
// Learning model parameters
6 $\mathbf{p}_u \leftarrow \mathbf{p}_u - \eta\frac{\partial l_{uij}}{\partial \mathbf{p}_u}$;
7 $\mathbf{q}_i \leftarrow \mathbf{q}_i - \eta\frac{\partial l_{uij}}{\partial \mathbf{q}_i}$;
8 $\mathbf{q}_j \leftarrow \mathbf{q}_j - \eta\frac{\partial l_{uij}}{\partial \mathbf{q}_j}$;
9 $\mathbf{E} \leftarrow \mathbf{E} - \eta\frac{\partial l_{uij}}{\partial \mathbf{E}}$;
10 **end**
11 **return** Θ

By **minimizing** the objective function, we obtain the model parameters $\mathbf{p}_u, \mathbf{q}_i, \mathbf{E}$, which can the model resistant to the adversarial perturbations on the instance (u, i, j).

In the next, we elaborate how to perform the two optimization procedures for a stochastic instance (u, i, j).

1. Learning Adversarial Perturbations. This step obtains perturbed vectors that are relevant to model updates for the instance (u, i, j), that is Δ_i and Δ_j. Due to the non-linearity of the objective function l'_{uij} and the ϵ-constraint in optimization, it is difficult to get the exact solution. As such, we borrow the idea from the fast gradient sign method [22], approximating the objective function by linearizing it around Δ_i and Δ_j; and then, we solve the constrained optimization problem on this approximated linear function. According to Taylor series, the linear function is the first-order Taylor expansion, for which the line's slope is the first-order derivative of the objective function on the variables. It is clear that to maximize a linear function, the optimal solution is to move the variables towards the direction of their gradients. Taking the ϵ-constraint into account, we can obtain the solution for adversarial perturbations as

$$\Delta_i = \epsilon\frac{\Gamma_i}{||\Gamma_i||} \quad \text{where} \quad \Gamma_i = \frac{\partial l'_{uij}}{\partial \Delta_i}, \tag{11}$$

$$\Delta_j = \epsilon\frac{\Gamma_j}{||\Gamma_j||} \quad \text{where} \quad \Gamma_j = \frac{\partial l'_{uij}}{\partial \Delta_j}, \tag{12}$$

Note that when a mini-batch of examples are sampled, l'_{uij} should be defined as the sum of loss over the examples in the mini-batch, since the target item i may also appear in other examples. Here we have omitted the details for the derivation, because modern machine learning toolkits like TensorFlow and PyTorch provide the auto-differential functionality. Moreover, we have also tried the fast gradient sign method as proposed in [22], which only keeps the sign of the derivation, *i.e.*, $\Delta_i = \epsilon\text{sign}(\Gamma_i)$. However, we find it is less effective than our solution on recommendation tasks.

2. Learning Model Parameters. This step updates model parameters by minimizing Equation (10). Since the perturbations Δ are fixed in this step, it becomes a conventional minimization problem and can be approached with gradient descent. Specifically, we perform a gradient step for each involved parameter:

$$\theta = \theta - \eta \frac{\partial l_{uij}}{\partial \theta}, \tag{13}$$

where $\theta = \{\mathbf{p}_u, \mathbf{q}_i, \mathbf{q}_j, \mathbf{E}\}$. η denotes the learning rate, which can be parameter-dependent if adaptive SGD methods are used, and we use the Adagrad [29] in our experiments.

For convergence, one can either check the decrease of L_{BPR} after a training epoch (defined as iterating $|\mathcal{I}^+|$ number of examples where $|\mathcal{I}^+|$ denotes the number of observed interactions in the dataset), or monitor the recommendation performance based on a holdout validation set.

Lastly, it is worth mentioning that the pre-training step (line 1 of Algorithm 1) is critical and indispensable for AMR. This is because that only when the model has achieved reasonable performance, the model's generalization can be improved by enhancing its robustness with perturbations; otherwise, normal training process is sufficient to lead to better parameters and adversarial training will negatively slow down the convergence.

3.4 Time Complexity Analysis

We analyze the time complexity our AMR, with VBPR as a contrast. Since AMR and VBPR employ the same prediction model (the difference is in the training loss), they have the same time complexity in model prediction.

In order to better express the time complexity during training, let O_f be the time complexity of forward propagation for $\hat{y}_{ui}$, O_b be that of backward propagation, and O_u be that of updating parameters. To compute the perturbations, there is an extra cost O_{adv} to obtain the gradients on the content feature. According to the definitions, VBPR costs two O_f, two O_b and one O_u because of the pair-wised loss. That is totally $2 \times O_f + 2 \times O_b + O_u$. AMR does O_f O_f O_{adv} O_{adv} O_f O_f O_b O_b O_b O_b O_u in sequence which is totally $4 \times O_f + 4 \times O_b + 2 \times O_{adv} + O_u$. Usually, the four operations are linearly correlated. For example, their time complexities in VBPR and AMR are all $O(K + KD)$ (the divided $O(K)$ indicates the original part of MF). Thus the complexity of AMR is about two times of that of VBPR. Empirically, the time training AMR for one epoch is about three times of that training VBPR on both datasets. The redundant time cost may be led by some constant level operations.

4 Experiments

In this section, we conduct experiments with the aim of answering the following questions:

- **RQ1** Can our proposed AMR outperform the state-of-the-art multimedia recommendation methods?
- **RQ2** How is the effect of the adversarial training and can it improve the generalization and robustness of the model?
- **RQ3** How do the key hyper-parameters ϵ and λ affect the performance?

We first describe the experimental settings, followed by results answering the above research questions.

TABLE 1: Statistics of our experimented data.

Dataset	User#	Item#	Interaction#	Sparsity
Pinterest	3,226	4,998	9,844	99.939%
Amazon	83,337	299,555	706,949	99.997%

4.1 Experimental Settings

4.1.1 Data Descriptions

We conduct experiments on two real-world datasets: Pinterest [16] and Amazon [19]. On both datasets, 1) each item is associated with one image; and 2) the user-item interaction matrix is highly sparse. Table 1 summarizes the statistics of the two datasets.

Pinterest. The Pinterest data is used for evaluating the image recommendation task. Since the original data is extremely large (over one million users and ten million images), we sample a small subset of the data to verify our method. Specifically, we randomly select ten thousand users, and then discard users with less than two interactions and items without interactions.

Amazon. The Amazon data is constructed by [30] for visually-aware product recommendation. We use the *women* category for evaluation. Similar to Pinterest, we first discard users with less than five interactions. We then remove items that have no interactions and correlated images.

4.1.2 Evaluation Protocol

Following the prominent work in recommendation [8], [9], we employ the standard *leave-one-out* protocol. Specifically, for each user we randomly select one interaction for testing, and utilize the remaining data for training. After splitting, we find that about 52.6% and 45.9% items in the testing set on Pinterest and Amazon respectively are cold-start (*i.e.*, out-of-sample) items. This poses challenges to traditional collaborative filtering methods and highlights the necessity of doing content-based filtering. During training, these cold-start items are not involved (note that they can not be used as negative samples to avoid information leak); during testing, we initialize the ID embedding of cold-start items as a zero vector, using only their image features to get the item embedding.

Since it is time-consuming to rank all items for every user during evaluation, we follow the common approach [9], [31] to sample 999 items that are not interacted with the user, and then rank the testing item among the 999 items. To evaluate the performance of top-N recommendation, we truncate the ranking list of the 1,000 items at position N, measuring its quality with the *Hit Ratio* (HR) and *Normalized Discounted Cumulative Gain* (NDCG). To be specific, HR@N measures whether the testing item occurs in the top-N list — 1 for yes and 0 for no; NDCG@N measures the position of the testing item in the top-N list, the higher the better. The default setting of N is 10 without special mention. We report the average scores of all users and perform one-sample paired t-test to judge the statistical significance when necessary.

4.1.3 Baselines

We compare AMR with the following methods.

POP is a non-personalized method that ranks items by their popularity, measured by the number of interactions

in the training data. It benchmarks the performance of personalized recommendation.

MF-eALS [32] is a CF method that trains the MF model with a weighted regression loss, where different missing entries are assigned to different weights (i.e., confidence to be true negatives). It eschews negative sampling by assigning a uniform target of 0 on all missing entries, which allows fast optimization on MF.

MF-BPR [8] is a CF method that trains the MF model with BPR pairwise ranking loss. Since MF is learned solely based on user-item interactions, it serves as a benchmark for models with visual signals.

DUIF [16] is a variant of LFM. It replaces the item embedding in MF by the projecting the deep image feature into the latent space. For a fair comparison with other methods, we also optimize DUIF with the BPR loss. We have tested DUIF by both training it from scratch and pre-training it with user embeddings of MF, and report the best results.

VBPR [19] is an extension of MF-BPR, which is tailored for visually-aware recommendation. The detailed description can be found in Section 2.2.1. For model initialization, we find that using the ID embeddings learned by MF leads to better performance, so we report this specific setting.

Our **AMR** method is implemented using Tensorflow, which is available at: https://github.com/duxy-me/AMR. For visually-aware methods (DUIF, VBPR and AMR), we use the same ResNet-50 [13] model[1] as the deep image feature extractor to make the comparison fair. Moreover, all models are optimized using mini-batch Adagrad with a mini-batch size of 512, and other hyper-parameters have been fairly tuned as follows.

4.1.4 Hyper-parameters Settings

To explore the hyper-parameter space for all methods, we randomly holdout a training interaction for each user as the validation set. We fix the embedding size to 64 and tune other hyper-parameters as follows. First, for baseline models MF-BPR, DUIF, and VBPR, we tune the learning rate in $[0.005, 0.01, 0.05, 0.1]$ and the L_2 regularizer in $[0, 10^{-6}, 10^{-4}, 10^{-2}, 1]$. After obtaining the optimal values of learning rate and L_2 regularizer for VBPR, we use them for our method and then tune the adversary-related hyper-parameters: ϵ and λ. Specifically, we first fix $\lambda = 1$, tuning ϵ in $[10^{-2}, 10^{-1}, 1, 10]$. Then, with the best ϵ, we tune λ in $[10^{-2}, 10^{-1}, 1, 10]$. Note that if the optimal value was found in the boundary, we further extend the boundary to explore the optimal setting. We report the best results for all methods.

4.2 Performance Comparison (RQ1)

Here we compare the performance of our AMR with baselines. We explore the top-N recommendation where $N \in \{5, 10, 20\}$. The results are listed in Table 2. Inspecting the results from top to bottom, we have the following observations.

First, on both datasets, personalized models (*i.e.,* MF-eALS, MF-BPR, VBPR and AMR) largely outperform the non-personalized method POP. Particularly, the largest improvements can achieve 280% on Pinterest as indicated by the RI column. This demonstrates the positive effect of doing personalization.

Second, among the personalized methods, VBPR outperforms MF-eALS, MF-BPR and DUIF in most cases. The improvements of VBPR over MF-BPR confirm that traditional CF models can be significantly enhanced by adding rich multimedia features. Meanwhile, we notice that DUIF shows much worse results than MF-BPR and MF-eALS even it has used the same visual features as VBPR. Considering the fact that DUIF leverages the multimedia features only to represent an item, we speculate that CF features (*i.e.,* ID embeddings) are more important than pure multimedia features in personalized recommendation.

Third, AMR consistently outperforms all baselines in terms of all metrics on both datasets. One advantage is that AMR is built on VBPR which performs better than BPR in general. More importantly, by introducing the adversarial examples in the training phase, AMR can learn better model parameters than the non-adversarial VBPR. Moreover, we test the performances on cold items, i.e., testing items that are not occurred in the training set. In this case, only image features are used to obtain the item embedding vector. In Pinterest, the HR@10/NDCG@10 improvement over VBPR is 74%/101%; in contrast, on non-cold items, the HR@10/NDCG@10 improvement is 6.2%/ 2.2%. This justifies the positive effect of our AMR on learning better image representations, thus leads to better multimedia recommendation performance, particularly for cold items are heavily relied on the image representations.

Finally, focusing on Amazon, we find that the improvements of MF-BPR over POP, VBPR over MF-BPR, and AMR over VBPR, are smaller than that on the Pinterest data. The reasons lie in several aspects. First, the relatively strong performance of POP indicates that popular products on Amazon are more likely to be purchased by users; by contrast, the click behaviors on Pinterest images do not exhibit such pattern. Second, the small improvements of VBPR over MF-BPR reveal that adding multimedia content features have only minor benefits when the CF effect is strong (evidenced by richer user-item interactions, see Table 1 for more details). That may explain why multimedia information is typically regarded as an auxiliary but not dominant feature in recommender system domain. Therefore, the result that AMR has smaller improvements over VBPR is acceptable, since on the Amazon data, the recommendation quality is not dominant by visual features and thus modeling them only have minor effects. Despite this, by using adversarial training, our AMR can still improve over VBPR significantly, as evidenced by the t-test. This demonstrates the usefulness of adversarial training in improving the overall generalization of the model.

4.3 Effect of Adversarial Training (RQ2)

In this subsection, we analyze the effect of adversarial training from two aspects: generalization and robustness.

4.3.1 Generalization

We show the training process of VBPR and AMR in Figure 3, where the y-axis denotes the testing performance evaluated per 50 epochs. We also show the performance

1. https://github.com/KaimingHe/deep-residual-networks

TABLE 2: Top-N recommendation performance where $N \in \{5, 10, 20\}$. RI is the relative improvement of AMR over baselines on average. * indicates that the improvements over baselines are statistically significant for $p < 0.05$.

	Pinterest							Amazon						
	HR@N			NDCG@N			RI	HR@N			NDCG@N			RI
	$N=5$	$N=10$	$N=20$	$N=5$	$N=10$	$N=20$		$N=5$	$N=10$	$N=20$	$N=5$	$N=10$	$N=20$	
Pop	0.0353	0.0604	0.0927	0.0213	0.0296	0.0376	281.73%	0.1003	0.1460	0.2040	0.0685	0.0832	0.0978	34.27%
MF-eALS	0.1262	0.1584	0.1953	0.0922	0.1025	0.1118	22.18%	0.1322	0.1660	0.2026	0.1009	0.1118	0.1211	7.98%
MF-BPR	0.1228	0.1534	0.1891	0.0949	0.1048	0.1138	22.82%	0.1306	0.1720	0.2183	0.0950	0.1084	0.1201	7.91%
DUIF	0.1116	0.1600	0.2179	0.0806	0.0962	0.1108	26.17%	0.0865	0.1317	0.1964	0.0568	0.0714	0.0876	52.61%
VBPR	0.1352	0.1829	0.2347	0.1005	0.1157	0.1287	7.70%	0.1333	0.1747	0.2249	0.0980	0.1113	0.1240	5.14%
AMR	**0.1392***	**0.2033***	**0.2697***	**0.1026***	**0.1230***	**0.1398***	-	**0.1402**	**0.1864***	**0.2360***	**0.1022**	**0.1171***	**0.1296***	-

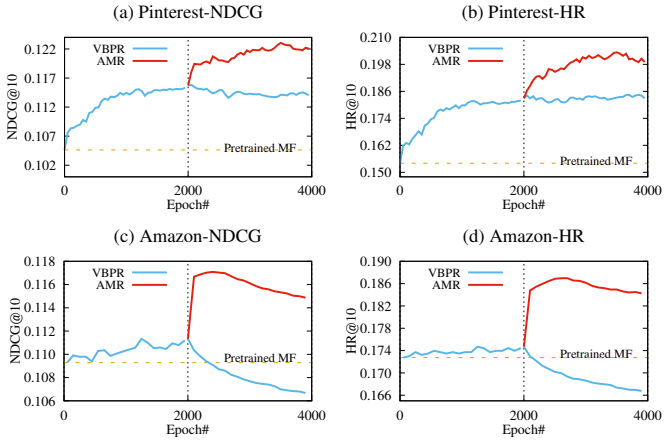


Fig. 3: Testing performance of VBPR and AMR evaluated per 50 epochs. We first train VBPR for 2000 epochs (which is initialized from MF parameters for better performance). We then continue training AMR for another 2000 epochs (with continue training VBPR as a comparison).

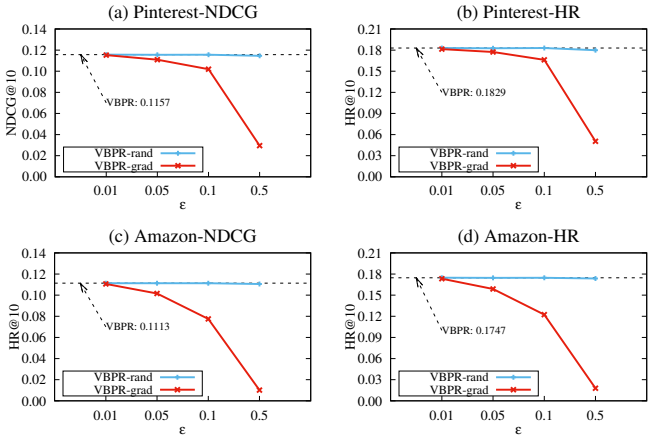


Fig. 4: Impact of applying random (VBPR-rand) and adversarial (VBPR-grad) perturbations to image features on VBPR. The key observation is that adversarial perturbations have a large impact on BPR.

of pretrained MF as a benchmark, since VBPR and AMR are initialized from MF parameters. Specifically, we first train VBPR until convergence (about 2000 epochs). Then we proceed to train AMR by initializing its parameters with the parameters learned by VBPR. As a comparison, we use the same parameters to initialize a new VBPR model and continue training it. As can be seen, by performing adversarial training based on VBPR parameters, we can gradually improve the performance to a large extent. By contrast, when performing normal training on VBPR, the performance is not improved, or even decreased due to overfitting (see results on Amazon). To be specific, on the Pinterest dataset, the best NDCG and HR of VBPR are 0.116 and 0.183 respectively, which are further improved to 0.123 and 0.203 by training with AMR. These results verify the highly positive effect of adversarial learning in AMR, which leads to better parameters that can improve model generalization.

4.3.2 Robustness

We now recap the motivating example about model robustness in Figure 1. To have a quantitative sense on the model robustness, we add adversarial perturbations to the original image and measure performance drop; smaller drop ratio means stronger robustness.

We first demonstrate the impact of perturbations on VBPR. Figure 4 exhibits the performances. The horizontal dashed line indicates the performance of unperturbed

TABLE 3: Performance drop (relatively decreasing ratio in NDCG@10) of VBPR and AMR in the presence of adversarial perturbations during the testing phase.

	$\epsilon = 0.05$		$\epsilon = 0.1$		$\epsilon = 0.2$	
Dataset	**VBPR**	**AMR**	**VBPR**	**AMR**	**VBPR**	**AMR**
Pinterest	-4.2%	-2.6%	-11.9%	-6.2%	-31.8%	-18.4%
Amazon	-8.7%	-1.4%	-30.4%	-5.3%	-67.7%	-20.2%

VBPR. The random perturbation (VBPR-rand) would decrease the performance by a small ratio. In contrast, our proposed adversarial perturbation (VBPR-grad) introduced in Equation 8 leads to a terrible impact on VBPR. Thus AMR addresses adversarial perturbation for the robust model. Another interesting observation is that the drop caused by the perturbation are exponential-like increased. The perturbation with $\epsilon = 0.01$ is inconspicuous, while the perturbation with $\epsilon = 0.5$ causes a fatal drop. That is a reference for setting ϵ in AMR.

Table 3 shows the relative performance drop of VBPR and AMR with different settings of ϵ (which controls the perturbation scale). We can see that across settings AMR has a much smaller performance drop than VBPR. For example, on Amazon, when ϵ sets to 0.05, VBPR decreases for 8.7% whereas AMR decreases for 1.4%, which is about 6 times smaller. These results provide important empirical evidence on the robustness of AMR, which is less vulnerable to adversarial examples than VBPR. Moreover, larger perturbations, denoted by the increasing ϵ, impact both models more severely. The perturbation with $\epsilon = 0.2$ almost dam-

age VBPR model since the performance drop of VBPR on Amazon is more than 67.7%. The drop of AMR is about $1/3$ of that of VBPR. These comparisons reveal the robustness of our proposed model AMR.

We further explore the changes when applying the perturbations with $\epsilon = 0.1$. We record the subtraction of the sample ranks without and with perturbations. The positive values indicate the bad impact. 0 means there are no changes. Figure 5 records the distribution of the subtractions. There are three major observations:

1) Most of the perturbations lead to worse performance, while a few lead to better performance. This is caused by the disorder of the predictions.
2) On both Pinterest and Amazon, the large impact of AMR is fewer than that of VBPR, and the mean and the variance of the drops of AMR are less than those of VBPR. Specifically, on Amazon, most of the samples do not have any changes so that AMR would give a stable recommending results. These situations verify the robustness of AMR.
3) There are a large ratio of changes larger than 500 in Pinterest while the ratios is much smaller in Amazon. That demonstrates that larger dataset may be relatively stable facing the perturbations.

4.4 Hyper-parameter Exploration (RQ3)

In this final subsection, we examine the impact of hyper-parameters of adversarial learning, *i.e.,* ϵ and λ, which control the scale of perturbation and the weight of adversary, respectively. In exploring the change of one hyper-parameter, all other hyper-parameters are fixed to the same (roughly optimal) value.

Figure 6 illustrates the performance change with respect to ϵ. We can see that the optimal results are obtained when $\epsilon = 0.1$ and $\epsilon = 1$ on Pinterest and Amazon, respectively. When ϵ is smaller than 1, increasing it leads to gradual improvements. This implies the utility of adversarial training when the perturbations are within a controllable scale. However, when ϵ is larger than the optimal point, the performance of AMR drops rapidly, which reveals that too large perturbations will destroy the training process. Figure 7 shows the results of varying λ. We can see that similar trends can be observed — when λ is smaller than a threshold, increasing it will improve the performance, and further increasing it beyond the threshold will decrease the performance significantly. Moreover, the threshold (*i.e.,* optimal λ) is different for the two datasets — 1 for Pinterest and 0.1 for Amazon, which indicates that the optimal setting of λ is data-independent and should be separately tuned for a dataset.

5 Related Work

In this section, we briefly review related work on multimedia recommendation and adversarial learning.

5.1 Multimedia Recommendation

In recommender system research, two lines of contributions are most significant to date: 1) pure Collaborative Filtering (CF) techniques such as matrix factorization [25] and its variants [9], and 2) content- or context- aware methods that rely on more complex models such as feature-based embeddings [2] and deep learning [33], [34]. While multimedia recommendation falls into the second category of content-based recommendations, it is more challenging yet popular, due to massive and abundant multimedia (*e.g.,* visual, acoustic and semantic) features in real-world information systems [17], [35].

To effectively leverage rich multimedia features, a variety of multimedia recommendation techniques have been proposed. For example, it is intuitive to integrate high-level visual features that are extracted from DNNs into traditional CF models. A typical method is VBPR [19] that extends the dot product-based embedding function in BPR [8] into visual feature-based predictors. While simple, VBPR demonstrates considerable improvements in recommendation quality due to the proper use of multimedia features. Similarly, DUIF [16] builds item embedding by converting from the CNN feature of the image. Following the two works, Liu *et al.* [36] takes the categories and styles annotated by CNNs as item features. Moreover, Lei *et al.* [20] and Kang *et al.* [35] do not directly use the features extracted in advance, but instead construct an end-to-end model by CNNs. At a finer granularity, Chen *et al.* [37] and ACF [14] crop images into several parts, and then integrate the features from each part with an attention mechanism, which has been an import technique in recommendation [38], [39]. Several other features have been exploited, such as acoustic [40], [41], aesthetic [5], relation-based [42] and location-aware features [43], [44].

The key idea of AMR is to increase the model robustness by making it less vulnerable to worst-case perturbations in input features. While the idea is originated from the two ICLR papers [21], [22], we are the first to implement the idea on multimedia recommender models and verify its effectiveness. Specifically, the two original ICLR papers worked on the classification task (which optimized the point-wise cross-entropy loss), whereas our work AMR addresses the ranking task (which optimized the pair-wise loss). Thus, for multimedia recommendation methods that optimize pair-wise ranking loss, such as the ACF [14], our method can be directly applied; for methods that optimize pointwise loss, such as the Personalized Key-frame Recommendation [14] and Deep Content-based Music Recommendation [17], we just need to adapt the loss in model optimization (Equation 7), whereas the learning procedure remains unchanged.

5.2 Adversarial Learning

Another relevant line of research is adversarial learning [45], which aims to find malicious examples to attack a machine learning method and then addresses the vulnerabilities of the method. Recent efforts have been intensively focused on DNNs owing to their extraordinary abilities in learning complex predictive functions. For example, Szegedy *et al.* [21] finds that several state-of-the-art DNNs consistently mis-classify adversarial images, which are formed by adding small perturbations that maximize the model's prediction error. While the authors speculated that the reason is caused by the extreme nonlinearity of DNNs, later findings by Goodfellow *et al.* [22] showed that the reason

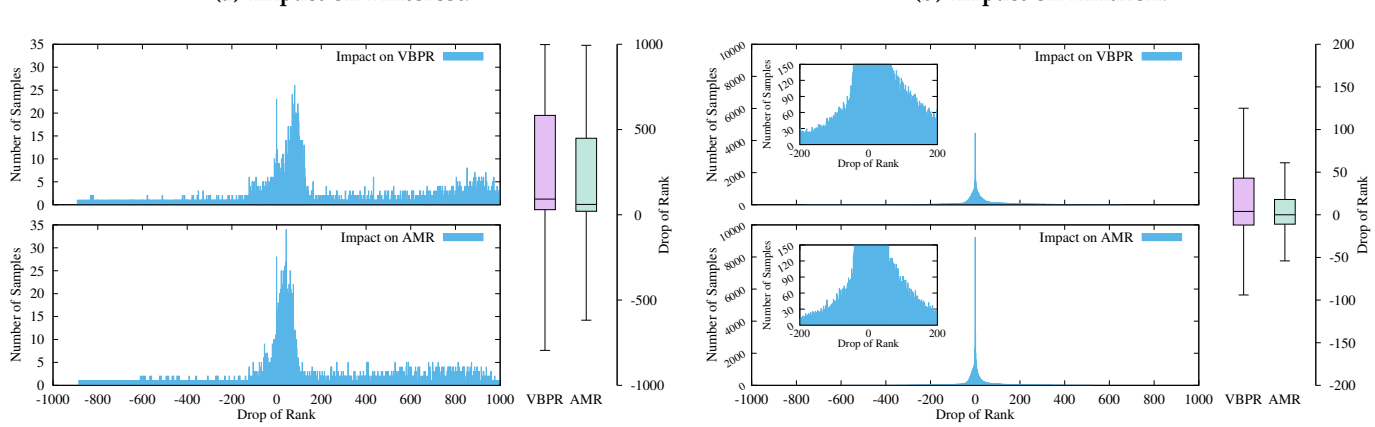


Fig. 5: The impacts with perturbations under $\epsilon = 0.1$. The boxplots represent the statistical distributions at the left side of them. The results, that the rank drop of AMR are closer to the zero point than that of VBPR, reveal that AMR gives more robust predictions than VBPR when facing perturbations.

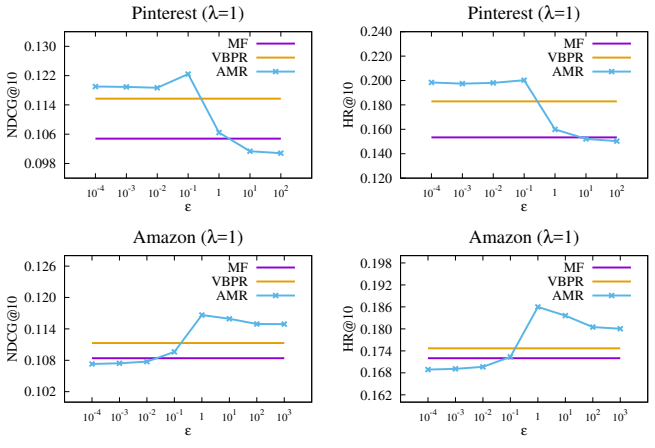


Fig. 6: Performance of AMR *w.r.t.* different values of ϵ. AMR obtains the best performance when $\epsilon = 0.1$ and $\epsilon = 1$ on Pinterest and Amazon, respectively.

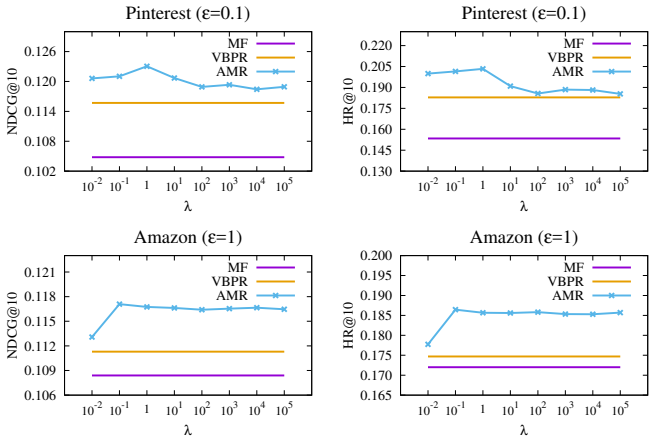


Fig. 7: Performance of AMR *w.r.t.* different values of λ. AMR obtains the best performance with $\lambda = 1$ and $\lambda = 0.1$ on Pinterest and Amazon, respectively.

is opposite — the vulnerability stems from the linearity of DNNs. They then proposed the fast gradient sign method that can efficiently generate adversarial examples with the linear assumption. Later on, the idea has been extended to several NLP tasks such as text classification [28]. Besides adding perturbations to input, other attempts have been made on the embedding layer [28] and dropout [46].

In the domain of recommendation, there are very few efforts exploring the vulnerability of recommender models. Some previous work [47] enhance the robustness of a recommender system by making it resistant to profile injection attacks, which try to insert fake user profiles to change the behavior of collaborative filtering algorithms. This line of research is orthogonal to this work, since we consider improving the robustness of recommender system from a different perspective of multimedia content. The work that is most relevant with ours is [27], which proposes a general adversarial learning framework for personalized ranking (aka., adversarial personalized ranking, short for APR). The key differences of AMR with APR are 1) APR is a general recommender framework focusing on the fundamental CF structure while AMR is a model focusing on multimedia recommendation with rich visual features, and 2) APR applies the perturbations on embeddings to increase the robustness of latent representations while AMR applies the perturbations on image features to increase the model tolerance for noisy inputs. To the best of our knowledge, this is the first work that explores adversarial learning in multimedia recommendation, opening a new door of improving the robustness and generalization of multimedia recommender systems.

6 Conclusion

In this work, we first showed that VBPR, a state-of-the-art image-aware recommendation method, is vulnerable to adversarial perturbations on images. The evidence is that by changing the images with very small perturbations that are imperceptible by human, we observed significant drop in recommendation accuracy. To address the lack of robustness issue of DNN-based multimedia recommender systems, we

presented a new recommendation solution named AMR. By simultaneously training the model and the adversary that attacks the model with purposeful perturbations, AMR obtains better parameters, which not only make the model more robust but also more effective. Extensive results on two real-world datasets demonstrate the utility of adversarial learning and the strength of our method.

In essence, AMR is a generic solution not limited to the model explored in this paper, but can serve as a general blueprint for improving any content-based recommender models. In future, we plan to extend the AMR methodology to more models, such as the attention-based neural recommender models [14] which might be more effective than LFM. Moreover, we will incorporate more contexts for multimedia recommendation, such as time, location, and user personality. Lastly, we are interested in building interactive recommender systems by unifying the recent advances in dialog agents with recommendation technologies.

REFERENCES

[1] X. He, Z. He, J. Song, Z. Liu, Y.-G. Jiang, and T.-S. Chua, "Nais: Neural attentive item similarity model for recommendation," *IEEE Transactions on Knowledge and Data Engineering*, 2018.

[2] F. Yuan, G. Guo, J. M. Jose, L. Chen, H. Yu, and W. Zhang, "Lambdafm: learning optimal ranking with factorization machines using lambda surrogates," in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, ser. CIKM '16, 2016, pp. 227–236.

[3] S. Wang, J. Tang, Y. Wang, and H. Liu, "Exploring hierarchical structures for recommender systems," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 6, pp. 1022–1035, 2018.

[4] D. Lian, Y. Ge, F. Zhang, N. J. Yuan, X. Xie, T. Zhou, and Y. Rui, "Scalable content-aware collaborative filtering for location recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 6, pp. 1122–1135, 2018.

[5] W. Yu, H. Zhang, X. He, X. Chen, L. Xiong, and Z. Qin, "Aesthetic-based clothing recommendation," in *WWW*. International World Wide Web Conferences Steering Committee, 2018, pp. 649–658.

[6] T. Chen, X. He, and M.-Y. Kan, "Context-aware image tweet modelling and recommendation," in *Proceedings of the 2016 ACM on Multimedia Conference*, ser. MM '16, 2016, pp. 1018–1027.

[7] J. Zhang, L. Nie, X. Wang, X. He, X. Huang, and T. S. Chua, "Shorter-is-better: Venue category estimation from micro-video," in *Proceedings of the 2016 ACM on Multimedia Conference*, ser. MM '16, 2016, pp. 1415–1424.

[8] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "Bpr: Bayesian personalized ranking from implicit feedback," in *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*, ser. UAI '09, 2009, pp. 452–461.

[9] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proceedings of the 26th International Conference on World Wide Web*, ser. WWW '17, 2017, pp. 173–182.

[10] J. Fan, D. A. Keim, Y. Gao, H. Luo, and Z. Li, "Justclick: Personalized image recommendation via exploratory search from large-scale flickr images," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 2, pp. 273–288, 2009.

[11] B. Chen, J. Wang, Q. Huang, and T. Mei, "Personalized video recommendation through tripartite graph propagation," in *Proceedings of the 20th ACM international conference on Multimedia*, ser. MM '12, 2012, pp. 1133–1136.

[12] J.-H. Su, W.-J. Huang, S. Y. Philip, and V. S. Tseng, "Efficient relevance feedback for content-based image retrieval by mining user navigation patterns," *IEEE transactions on knowledge and data engineering*, vol. 23, no. 3, pp. 360–372, 2011.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, ser. CVPR '16, 2016, pp. 770–778.

[14] J. Chen, H. Zhang, X. He, L. Nie, W. Liu, and T.-S. Chua, "Attentive collaborative filtering: Multimedia recommendation with item- and component-level attention," in *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '17, 2017, pp. 335–344.

[15] X. Chen, Y. Zhang, Q. Ai, H. Xu, J. Yan, and Z. Qin, "Personalized key frame recommendation," in *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '17, 2017, pp. 315–324.

[16] X. Geng, H. Zhang, J. Bian, and T. Chua, "Learning image and user features for recommendation in social networks," in *International Conference on Computer Vision*, ser. ICCV '15, 2015, pp. 4274–4282.

[17] A. v. d. Oord, S. Dieleman, and B. Schrauwen, "Deep content-based music recommendation," in *Proceedings of the 26th International Conference on Neural Information Processing Systems*, ser. NIPS'13, 2013, pp. 2643–2651.

[18] Z. Cheng, X. Chang, L. Zhu, R. C. Kanjirathinkal, and M. Kankanhalli, "Mmalfm: Explainable recommendation by leveraging reviews and images," *TOIS*, 2018.

[19] R. He and J. McAuley, "VBPR: visual bayesian personalized ranking from implicit feedback," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, ser. AAAI '16, 2016, pp. 144–150.

[20] C. Lei, D. Liu, W. Li, Z.-J. Zha, and H. Li, "Comparative deep learning of hybrid representations for image recommendations," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, ser. CVPR '16, 2016, pp. 2545–2553.

[21] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *International Conference on Learning Representations*, ser. ICLR '14, 2014.

[22] I. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *International Conference on Learning Representations*, ser. ICLR '15, 2015.

[23] S. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," in *IEEE Conference on Computer Vision and Pattern Recognition*, ser. CVPR '17, 2017, pp. 86–94.

[24] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," in *International Conference on Learning Representations*, ser. ICLR '17, 2017.

[25] X. He, H. Zhang, M.-Y. Kan, and T.-S. Chua, "Fast matrix factorization for online recommendation with implicit feedback," in *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '16, 2016, pp. 549–558.

[26] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, ser. NIPS '12, 2012, pp. 1097–1105.

[27] X. He, Z. He, X. Du, and T.-S. Chua, "Adversarial personalized ranking for recommendation." in *Proceedings of the 41th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '18, 2018, pp. 355–364.

[28] T. Miyato, A. M. Dai, and I. Goodfellow, "Adversarial training methods for semi-supervised text classification," in *International Conference on Learning Representations*, ser. ICLR '17, 2017.

[29] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, no. Jul, pp. 2121–2159, 2011.

[30] J. McAuley, C. Targett, Q. Shi, and A. Van Den Hengel, "Image-based recommendations on styles and substitutes," in *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '15, 2015, pp. 43–52.

[31] A. M. Elkahky, Y. Song, and X. He, "A multi-view deep learning approach for cross domain user modeling in recommendation systems," in *Proceedings of the 24th International Conference on World Wide Web*, ser. WWW '15, 2015, pp. 278–288.

[32] X. He, H. Zhang, M.-Y. Kan, and T.-S. Chua, "Fast matrix factorization for online recommendation with implicit feedback," in *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 2016, pp. 549–558.

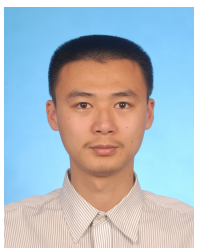
[33] X. He and T.-S. Chua, "Neural factorization machines for sparse predictive analytics," in *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '17, 2017, pp. 355–364.

[34] J. Xiao, H. Ye, X. He, H. Zhang, F. Wu, and T. Chua, "Attentional factorization machines: Learning the weight of feature interactions via attention networks," in *IJCAI*, 2017, pp. 3119–3125.

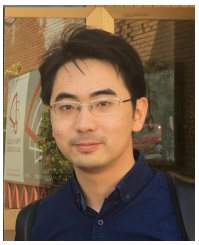
[35] W.-C. Kang, C. Fang, Z. Wang, and J. McAuley, "Visually-aware fashion recommendation and design with generative image mod-

els," in *International Conference on Data Mining*, ser. ICDM '17, 2017, pp. 207–216.
[36] Q. Liu, S. Wu, and L. Wang, "Deepstyle: Learning user preferences for visual recommendation," in *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '17, 2017, pp. 841–844.
[37] X. Chen, Y. Zhang, H. Xu, Y. Cao, Z. Qin, and H. Zha, "Visually explainable recommendation," *arXiv preprint arXiv:1801.10288*, 2018.
[38] X. Wang, X. He, F. Feng, L. Nie, and T. Chua, "TEM: tree-enhanced embedding model for explainable recommendation," in *WWW*, 2018, pp. 1543–1552.
[39] Z. Cheng, Y. Ding, X. He, L. Zhu, X. Song, and M. S. Kankanhalli, "Aˆ 3ncf: An adaptive aspect attention model for rating prediction." in *IJCAI*, 2018, pp. 3748–3754.
[40] Y. Deldjoo, M. Elahi, M. Quadrana, and P. Cremonesi, "Using visual features based on mpeg-7 and deep learning for movie recommendation," *International Journal of Multimedia Information Retrieval*, pp. 1–13, 2018.
[41] Z. Cheng, J. Shen, L. Zhu, M. S. Kankanhalli, and L. Nie, "Exploiting music play sequence for music recommendation." in *IJCAI*, 2017, pp. 3654–3660.
[42] X. Yang, Y. Ma, L. Liao, M. Wang, and T.-S. Chua, "Transnfcm: Translation-based neural fashion compatibility modeling," in *AAAI*, 2019.
[43] P. Zhao, X. Xu, Y. Liu, V. S. Sheng, K. Zheng, and H. Xiong, "Photo2trip: Exploiting visual contents in geo-tagged photos for personalized tour recommendation," in *Proceedings of the 2017 ACM on Multimedia Conference*, ser. MM '17, 2017, pp. 916–924.
[44] J. Chen, X. He, X. Song, H. Zhang, L. Nie, and T.-S. Chua, "Venue prediction for social images by exploiting rich temporal patterns in lbsns," in *MMM*. Springer, 2018, pp. 327–339.
[45] D. Lowd and C. Meek, "Adversarial learning," in *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, ser. KDD '05, 2005, pp. 641–647.
[46] S. Park, J.-K. Park, S.-J. Shin, and I.-C. Moon, "Adversarial dropout for supervised and semi-supervised learning," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, ser. AAAI '18, 2018.
[47] R. Burke, M. P. O'Mahony, and N. J. Hurley, *Robust Collaborative Recommendation*. Boston, MA: Springer US, 2015, pp. 961–995.

Jiuhui Tang is currently a Professor in School of Computer Science and Engineering, Nanjing University of Science and Technology, China. He received the B.Eng. and Ph.D. degrees from the University of Science and Technology of China, Hefei, China, in 2003 and 2008, respectively. From 2008 to 2010, he worked as a research fellow in School of Computing, National University of Singapore. His current research interests include multimedia content analysis and retrieval, social media mining and machine learning. He has authored over 100 papers in top-tier journals and conferences. Dr. Tang is a recipient of the inaugural ACM China Rising Star Award, the Best Paper Awards in ACM MM 2007, PCM 2011 and ICIMCS 2011, the Best Paper Runner-up in ACM MM 2015, and the Best Student Paper Awards in MMM 2016 and ICIMCS 2017.



Xiaoyu Du is currently a lecturer in the School of Software Engineering of Chengdu University of Information Technology, Chengdu, a visiting scholar in the NeXT++ of National University of Singapore, and a Ph.D. candidate of University of Electronic Science and Technology of China, Chengdu. He received his M.E. degree in computer software and theory in 2011 and B.S. degree in computer science and technology in 2008, both from Beijing Normal University, Beijing. His research interests include information retrieval, computer vision, and machine learning.



Xiangnan He is currently a senior research fellow with School of Computing, National University of Singapore (NUS). He received his Ph.D. in Computer Science from NUS. His research interests span information retrieval, data mining, and multi-media analytics. He has over 50 publications appeared in several top conferences such as SIGIR, WWW, and MM, and journals including TKDE, TOIS, and TMM. His work on recommender systems has received the Best Paper Award Honourable Mention in WWW 2018 and ACM SIGIR 2016. Moreover, he has served as the PC member for several top conferences including SIGIR, WWW, MM, KDD etc, and the regular reviewer for journals including TKDE, TOIS, TMM, TNNLS etc.

Fajie Yuan is currently a Ph.d student in in the School of Computing Science at the University of Glasgow. His research interests cover recommender systems, natural language processing and machine learning. His work appears in major international conferences, including ACL, CIKM, IJCAI and IUI etc.

Qi Tian received the B.E. degree in electronic engineering from Tsinghua University, China, the M.S. degree in electrical and computer engineering from Drexel University, and the Ph.D. degree in electrical and computer engineering from the University of Illinois at Urbana-Champaign, in 1992, 1996, and 2002, respectively. He is currently a Professor with the Department of Computer Science, University of Texas at San Antonio (UTSA). His research interests include multimedia information retrieval and computer vision.

Tat-Seng Chua is the KITHCT Chair Professor at the School of Computing, National University of Singapore. He was the Acting and Founding Dean of the School during 1998-2000. Dr Chuas main research interest is in multimedia information retrieval and social media analytics. In particular, his research focuses on the extraction, retrieval and question-answering (QA) of text and rich media arising from the Web and multiple social networks. He is the co-Director of NExT, a joint Center between NUS and Tsinghua University to develop technologies for live social media search. Dr Chua is the 2015 winner of the prestigious ACM SIGMM award for Outstanding Technical Contributions to Multimedia Computing, Communications and Applications. He is the Chair of steering committee of ACM International Conference on Multimedia Retrieval (ICMR) and Multimedia Modeling (MMM) conference series. Dr Chua is also the General Co-Chair of ACM Multimedia 2005, ACM CIVR (now ACM ICMR) 2005, ACM SIGIR 2008, and ACM Web Science 2015. He serves in the editorial boards of four international journals. Dr. Chua is the co-Founder of two technology startup companies in Singapore. He holds a PhD from the University of Leeds, UK.