Adversarial Item Promotion: Vulnerabilities at the Core of Top-N Recommenders that Use Images to Address Cold Start

Zhuoran Liu
Radboud University, Netherlands
z.liu@cs.ru.nl

Martha Larson
Radboud University, Netherlands
m.larson@cs.ru.nl

ABSTRACT

E-commerce platforms provide their customers with ranked lists of recommended items matching the customers' preferences. Merchants on e-commerce platforms would like their items to appear as high as possible in the top-N of these ranked lists. In this paper, we demonstrate how unscrupulous merchants can create item images that artificially promote their products, improving their rankings. Recommender systems that use images to address the cold start problem are vulnerable to this security risk. We describe a new type of attack, *Adversarial Item Promotion* (AIP), that strikes directly at the core of Top-N recommenders: the ranking mechanism itself. Existing work on adversarial images in recommender systems investigates the implications of conventional attacks, which target deep learning classifiers. In contrast, our AIP attacks are embedding attacks that seek to push features representations in a way that fools the ranker (not a classifier) and directly leads to item promotion. We introduce three AIP attacks *insider attack*, *expert attack*, and *semantic attack*, which are defined with respect to three successively more realistic attack models. Our experiments evaluate the danger of these attacks when mounted against three representative visually-aware recommender algorithms in a framework that uses images to address cold start. We also evaluate potential defenses, including adversarial training and find that common, currently-existing, techniques do not eliminate the danger of AIP attacks. In sum, we show that using images to address cold start opens recommender systems to potential threats with clear practical implications.

CCS CONCEPTS

• **Information systems** → **Recommender systems**; • **Security and privacy** → **Web application security**.

1 INTRODUCTION

Visually-aware recommender systems [20, 24, 32] incorporate image information into their ranking mechanism. This information
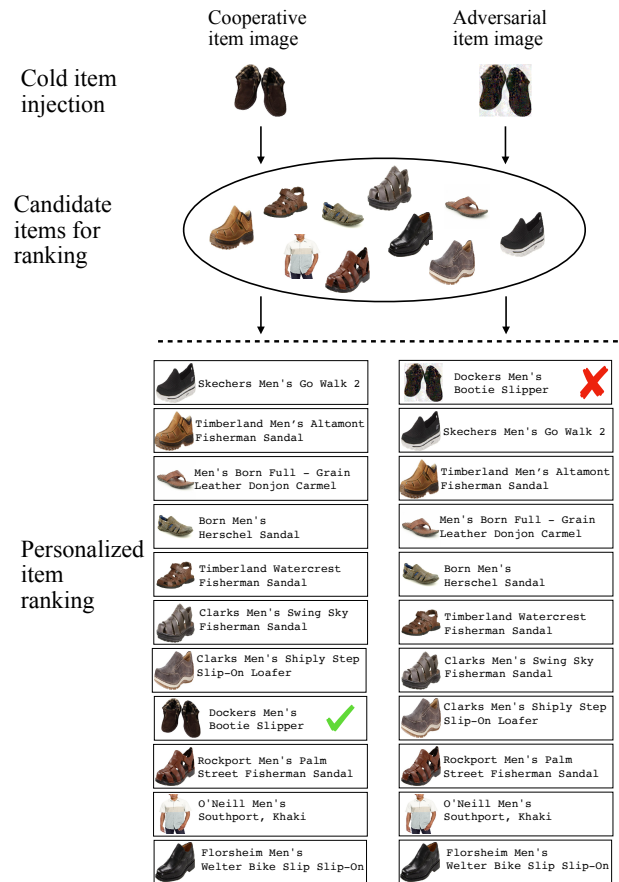
Figure 1: The cooperative cold item image and its corresponding adversarial cold item image (top) are each injected into the candidate set, which is generated by a first-stage ranker (here, BPR). In the personalized ranked list generated by the visually-aware second-stage ranker (here, DVBPR) the advesarial item (right) lands much higher than the cooperative item (left). (Diagram shows a real example. The adversarial item image is generated by our INSA attack with $\epsilon = 32$ and epoch = 5 explained in detail in Section 4.1).

helps to address the challenge of cold start since it compensates for insufficient interactions associated with new users or items. In this paper, we show how the use of image content for cold start opens visually-aware recommenders to vulnerability.[1]

The vulnerability is due to adversarial examples, which are samples deliberately designed to cause a machine learning system to

---

[1]Code available at: https://github.com/liuzrcc/AIP

make mistakes. The computer vision community has developed an in-depth understanding of how adversarial images can be used to attack classifiers, starting with [3, 41]. Classifier-targeted adversarial images can have an impact on recommender systems that leverage image content, as has been demonstrated by TAaMR [13]. However, until now, recommender system researchers have not considered how images can be modified to create adversarial items that attack visually-aware Top-N recommender systems by directly targeting the ranker, rather than a classifier.

We expose the vulnerability of visually-aware recommender systems to adversarial items by presenting a series of attacks and by experimentally assessing the threat that they pose. We also examine possible defenses. Adversarial training has been proposed in order to improve the general performance of multimedia recommender systems. The dominant approach is currently AMR [42]. Our experiments show that AMR is not sufficient to defend against our adversarial attacks. Further, other common defenses, such as image compression also fall short. In sum, the vulnerabilities of visually-aware recommender systems that we investigate here are serious and require further attention of the research community.

Our work is part of the long tradition of research devoted to the security and robustness of recommender system algorithms [5, 9, 15, 16, 27, 29, 33, 35, 43]. Most work, however, focuses on vulnerabilities related to user profiles. Early work looked at *shilling* [27], which uses fake users. Shilling was later generalized to *profile injection attacks* [33] or *poisoning attack* [29]. In our work, in contrast, we are looking at attackers who are able to manipulate items directly, and, specifically, to choose item images. In other words, instead of looking at profile-related attacks we are looking at an *item representation attacks*. Concretely, the risk of such attacks presents itself in the case of e-commerce platforms that sell the items of individual merchants, e.g., e-commerce or customer to customer (C2C) marketplaces. The merchants create their own item description, including images. We show that if such merchants act unscrupulously they can artificially promote their items and compromise the security of the recommender system.

Figure 1 illustrates the mechanics of the attack that we consider in this paper, called an *Adversarial Item Promotion* (AIP) attack. On the left, we see personalized item ranking for a user when the recommender system is *not* under attack (i.e., the cold start item is a "cooperative item"). On the right, we see the ranking when the recommender system is under attack by an unscrupulous merchant, who has used a manipulated image in an item representation (i.e., the cold start item is an "adversarial item"). This setup reflects the way that recommender system platform would add a certain number of cold items to the personalized ranked lists of users in order to allow the items to start accumulating interactions. We choose a two-stage recommender, since they are used in industry [11, 45]. With the two stage recommender, we ensure that the adversarial cold item is competing against selected candidate items that are already very relevant to the user.

We provide a short walk-through of Figure 1, which illustrates the attack at the level of a single cold item and a single user. A set of "candidate items for ranking" for that user has been selected using a conventional personalized Top-N recommender. Then the cold item is injected into that set. Finally, a visually-aware personalized Top-N recommender is used to rank the candidate item set before it

is presented to the user. We see that in the case of the cooperative image (left), the cold item lands somewhere in the ranked list, but probably not at the top. In contrast, in the case of the adversarial image (right), the cold item lands at the top of the personalized item ranking.

The overall impact of the attack depends on the accumulated effect of the attack over all users, and not just a single instance of the attack shown in Figure 1. It is important to understand that the final rank position of the cooperative vs. adversarial item will be different for each instance of the attack. However, in general, if the adversarial item of an unscrupulous merchant lands consistently farther towards the top of users' personalized recommendation lists than it deserves to, then, at large scale, the merchant will accrue considerable benefit.

We choose to focus on the cold-start problem because of its importance for recommender systems. However, there is also another reason. A straightforward, practical approach to blocking adversarial image promotion attacks on non-cold-start items is to prevent merchants from being able to change images once their items have started to accumulate interactions. Cold start is the most important moment of opportunity for a merchant to introduce an adversarial image into a representation. Every item starts in some way cold, and the issue particularly extreme in C2C marketplaces selling many unique items.

With this paper, we make the following contributions:

- We propose three Adversarial Item Promotion (AIP) attacks on the ranking mechanism of visually-aware recommender systems in a cold item scenario and experimentally assess their impact. The attacks correspond to three different levels of knowledge and we test them using two real-world data sets.
- We show that there is no easy defense against AIP attacks. The currently dominant adversarial training, as well as conventional defenses such as compression, are not sufficient to eliminate the vulnerability.
- We release an implementation of our attacks and defenses that allows for testing and extension.

This paper follows the standard procedure for security research. First, we specify a framework including the types of attacks expected (attack models), the systems to be attacked, and a means of measuring the impact of the attacks. Then, we propose attacks for each attack model and evaluate their success. The systems that we attack are representative of visually-aware recommender systems, i.e., a visual feature-based similarity model (AlexRank) based on AlexNet [25], a Collaborative-Filtering (CF) model leveraging visual features (VBPR [20]), and state-of-the-art learning-based neural model (DVPBR [24]). Finally, we turn to the analysis of possible defenses and close with a conclusion and outlook.

2 RELATED WORK

2.1 Robustness of Recommender System

In this section, we review previous work on recommender robustness. Note that this work focused on user profiles, not image content. O'Mahony *et al.* [35] introduce the definition of recommender system robustness and present several attacks to identify characteristics that influence robustness. Lam and Riedl [27] explore shilling

attacks in recommender systems by evaluating recommendation performance under different scenarios. In particular, they find that new or obscure items are more especially susceptible to attack, and they suggest that obtaining ratings from a trusted source for these items could make them less vulnerable. Mobasher *et al.* [33] propose a formal framework to characterize different recommender system attacks, and they also propose an approach to detect attack profiles. In [33] and [5], evaluation metrics, e.g., hit rate for item and prediction shift, for the robustness of recommender systems are discussed. Recently, instead of model-agnostic profile injection attacks, poisoning attacks that leverage exact recommendation model information have been proposed. Li *et al.* [29] propose poisoning attacks on factorization-based CF algorithms that approximate the gradient based on first-order KKT conditions. Christakopoulou and Banerjee [9] propose a generative approach to generate fake user profiles to mount profile injection attacks. Fang *et al.* propose poisoning attacks to graph-based recommender systems [16]. They also propose to generate fake user-item interactions based on influence function [15]. Tang *et al.* [43] propose effective transfer-based poisoning attacks against recommender systems, but they mention that their approach is less effective on cold items. Our "item representation attack" is distinct from a "profile injection attack" or "poisoning attack", but both kinds of attacks have similar impacts, namely, pushing items that have been targeted for promotion.

2.2 Visually-aware Recommender System

Visually-aware recommender systems incorporate visual information into their recommendation ranking mechanism. Originally, visually-aware recommenders relied on image content retrieval to make preference predictions. Given a query image, Kalantidis *et al.* [23] gather segmentation parts and retrieve visually similar items within each of the predicted classes. Later, semantic information of images is also incorporated to improve retrieval performance. Jagadeesh *et al.* [21] collect a large-scale dataset, Fashion-136K, with detailed annotations and propose several retrieval-based approaches to recommend a matching item based on the query image.

Beyond image retrieval-based recommendation approaches, user-item interactions are leveraged in visually-aware recommenders. IBR [32] models human notions of similarity by considering alternative or complementary items. Algorithms also incorporate the visual signal into CF models so as to exploit user feedback and visual features simultaneously, e.g., VBPR [20] and Fashion DNA [4]. Recently, with the advances in computational resources, learning-based neural frameworks have been proposed and achieve state of the art performance on fashion recommendation (DVBPR [24]) and reciprocal recommendations (ImRec [34]). In our paper, to comprehensively evaluate AIP attacks in different recommenders, we select three representatives: an image-retrieval-based similarity model, a CF model leveraging visual features and a learning-based neural model.

2.3 Adversarial Machine Learning

Adversarial examples are data samples that are deliberately designed in order to mislead machine learning algorithms [3, 41]. A limited amount of work, as mentioned above, has addressed adversarial images for recommender systems. The work most closely

related to our own [13] looks only at classification-based issues. Di Noia *et al.* [13] propose Targeted Adversarial Attack against Multimedia Recommender Systems (TAaMR), and they use two classification-based adversarial attacks, namely Fast Gradient Sign Method (FGSM) [41] and Projected Gradient Descent (PGD) [26], to evaluate two visually-aware recommender systems. In contrast to [13], we show that the problem of adversarial examples in recommender system goes beyond the problem of classifier-targeted adversarial examples.

Adversarial training is a promising techniques to tackle adversarial examples [31]. As stated in Section 1, research on adversarial training in visually-aware recommnder systems has, until this point, focused on improving general performance. Specifically, AMR [42] aims to improve recommendation with adversarial training (cf. Section 7.1 for details) and considers the robustness of recommender systems perturbations in system-internal representations. In contrast, our goal is to investigate security vulnerability originating from an external adversary who attacks item images. We show that simple adversarial training (i.e., AMR) is not a guarantee for robustness against AIP attacks (cf. Section 7.1).

3 BACKGROUND AND FRAMEWORK

This section introduces the background and framework in which the attack models are developed and evaluated. Figure 2 gives the overview of the setup. As introduced in Section 1, we use a two-stage approach. The first-stage recommender generates a personalized set of candidate items. For this purpose, we choose Bayesian Personalized Ranking (BPR) [38], a representative CF model that is trained on the user-item interaction data. We use the visually-aware second-stage recommender to make a comparison between the cold start of a cooperative item and an adversarial item.

In this section, we first present our three attack models (Section 3.1), which are the basis for three specific AIP attacks, INSA, EXPA, and SEMA, explained in Section 4. Then, we present the three representative visually-aware recommenders that we attack (Section 3.2). Finally, we explain the dimensions along which we evaluate the impact of the attacks (Section 3.3).

Table 1: The three AIP attack models characterized by the knowledge to which the attacker has access for each.

Attack model	General Knowledge	Visual feature extraction model	Embeddings
High knowledge cf. INSA (Section 4.1)			×
Medium knowledge cf. EXPA (Section 4.2)	×	×	
Low knowledge cf. SEMA (Section 4.3)	×		

3.1 Attack Models

We define three attack model following the three dimensions relevant for trustworthy recommender systems [33]. The *Intent* dimension captures the objective of the attacker. All our models use 'push' intent, i.e., attackers are merchants who want their items
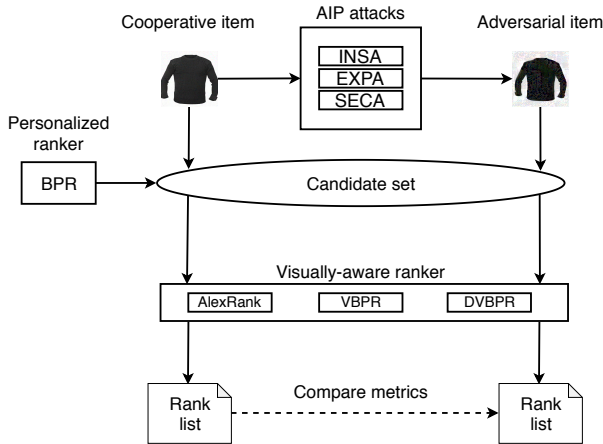
Figure 2: Setup for attack and attack evaluation.

promoted to higher ranks in users' personalized recommendation lists. The *Knowledge* dimension captures how much information that the attacker has about the system being attacked. Defining knowledge levels is common in adversarial machine learning research [3, 7]. Our three attack models correspond to three different levels of knowledge: high, medium, and low. The *Scale* dimension captures the scope of the interference. In all our attack models, we assume an attack with minimum scale, e.g., only a single item is attacked at any given moment. Such a small-scale attack is least likely to be noticed. It is clear that the harm caused by the attack will increase with scale.

Table 1 summarizes our attack models in terms of the level of knowledge involved. For our high-knowledge attack model we assume that the attacker is an insider at the recommender system platform and has access to the user embeddings of the trained recommender model. This scenario is not particularly realistic, but it is important because it demonstrates an upper bound for the potential damage that can be inflicted by an AIP attack.

The medium- and low-knowledge attack models are more realistic and assume that the attacker has general knowledge of the market in which the recommender system operates. In particular, the attacker must be able to identify (by observing sales trends or advertising) at least one item that sells well on the platform. We refer to this item as a *hook* item. The attack is strongest when the hook item image is an image used by the recommender, but it could also be an image of the item acquired elsewhere. As will be explained in Section 4, the adversarial item will use the hook item to pull itself up in the ranked list. In the medium-knowledge attack model, in addition to general knowledge, the attacker must have access to the pre-trained visual feature extractor used by the visually-aware recommender systems. Recommender systems leveraging a pre-trained visual feature extractor are prevalent in both academic research and industry, e.g., [17, 20, 28, 32, 37, 40]. These models are often released as publicly available resources in transfer learning.

3.2 Visually-aware Recommender Systems

In this section, we introduce the three representative visually-aware recommender systems that we will attack in our experiments: the visual feature-based similarity model (AlexRank), the CF model

leveraging visual features (VBPR) and the learning-based neural approach (DVBPR). We chose these recommenders because they represent the three types of commonly used visually-aware recommender systems.

3.2.1 AlexRank. Image-content-retrieval-based recommendation is a nearest neighbor approach that ranks items by visual feature similarity of product images. Such methods are commonly used as baseline approaches in visually-aware recommender system research [24]. Here we use the output of the second fully-connected layer of AlexNet [25] as the visual feature of item images. Given an image of item i, the average Euclidean distance between the visual feature of item i and all items that user u has interacted with is calculated, so smaller distance means higher preference prediction. Equation 1 show the calculation of similarity predictor:

$$p_{u,i} = \sum_{j \in \mathcal{I}_u^+} \frac{-\|\Phi_f(\mathbf{X}_i) - \Phi_f(\mathbf{X}_j)\|^2}{|\mathcal{I}_u^+|}, \tag{1}$$

where $\mathcal{I}_u^+$ is the set of items that user u has interacted with, and $\mathbf{X}_i, \mathbf{X}_j$ represent images of item i and j. Φ_f is the pre-trained model for image feature extraction. Note that the final ranking of item i for user u is solely determined by preference score $p_{u,i}$.

3.2.2 VBPR. Extended from BPR [38], VBPR [20] incorporates visual features into the CF model. By leveraging image features of pre-trained CNN models, VBPR improves the recommendation performance of BPR. The preference prediction of VBPR is described in Equation 2:

$$p_{u,i} = \alpha + \beta_u + \beta_i + \gamma_u^T \gamma_i + \theta_u^T (E\Phi_f(\mathbf{X}_i)), \tag{2}$$

where θ_u is the user content embedding, and E is the parameter for the visual feature. $\Phi_f(\mathbf{X}_i)$ represents the visual feature of item $\mathbf{X}_i$ from pre-trained model Φ_f, and α, β_u and β_i are user, item biases and global offset term. γ_u and γ_i represent the latent interaction-based embeddings for user and item. For model learning, VBPR adopts the pairwise ranking optimization framework from BPR. The training triples set $\mathcal{D}_s$ is described in Equation 3:

$$\mathcal{D}_s = \{(u, i, j) | u \in \mathcal{U} \land i \in \mathcal{I}_u^+ \land j \in \mathcal{I}/\mathcal{I}_u^+\} \tag{3}$$

where $\mathcal{U}$ and $\mathcal{I}$ represent the user and item sets, i represents the interacted item, and j represents the non-interacted item. A bootstrap sampling of training triples is used for model training. The optimization objective of VBPR is described in Equation 4:

$$\underset{\Theta}{\operatorname{argmin}} \sum_{(u,i,j) \in \mathcal{D}_s} -\ln \sigma(p_{u,i} - p_{u,j}) + \lambda_\Theta \|\Theta\|^2 \tag{4}$$

where Θ represents all model parameters and λ_Θ is the weight of regularization term.

3.2.3 DVBPR. DVBPR [24] is a concise end-to-end model whose visual feature extractor is trained directly in a pair-wise manner. DVBPR achieves the state-of-the-art performance on several data sets for visually-aware recommendations [24]. The preference prediction of DVBPR is described in Equation 5:

$$p_{u,i} = \theta_u^T \Phi_e(\mathbf{X}_i), \tag{5}$$

where θ_u is the user content embedding, and $\Phi_e(\mathbf{X}_i)$ is the item content embedding where the CNN model Φ_e is trained directly in a pair-wise manner.

3.3 Attack Evaluation Dimensions

We evaluate attacks according to the aspects of *integrity* and *availability* distinguished in machine learning security [1]. Our main concern is the ability of the attack to compromise the integrity of the recommender system, which is related to the success of the 'push' intent of our attack models (cf. Section 3.1). We measure the ability of our attacks to raise the rank of cold-start items such that adversarial cold start items land higher than the corresponding cooperative cold start items in users' personalized recommendation lists. Specifically, we report the change in rank of cold-start items with prediction shift and change in hit rate (cf. Section 5.2). In addition, we measure availability, which is related to whether the recommender system remains useful to other merchants and to customers while under attack. For this purpose, we use 'ordinary' test items. (We adopt the test items defined by the data sets.) We calculate change in hit rate when a ordinary test item is in a candidate set with a cooperative cold item and when the same ordinary test item is in a candidate set with an adversarial cold item.

We also consider the extent to which the attack is noticeable to the human eye. As attacks increase in strength, perturbations become visible in images and adversarial images can be identified by experts who know what they were looking for. However, e-commerce platforms are so large and the turnover of items so fast that it is impossible to manually vet all of the images representing items, cf. the known difficulty of filtering item collections for banned or unsafe products [2]. For these reason, strong attacks (i.e., larger image perturbations) are quite realistic. By focusing our experiments on strong attacks, we can evaluate the extent of the vulnerability of the system. We also carry out additional experiments that demonstrate the effect of an attack increasing in strength from weak to strong. In this way, we shed light on what might happen if user clicks are affected by an impression of low quality due to the presence of perturbations in images. Our additional experiments show that the success of the attack is not dependent on a highly noticeable change to the image appearance.

4 ADVERSARIAL ITEM PROMOTION ATTACKS

In this section, we introduce three adversarial item promotion (AIP) attacks corresponding to the three attack models previously introduced (cf. Section 3.1).

4.1 Insider Attack (INSA)

The high-knowledge attack model assumes the attacker has insider access to the user embeddings of the trained model (see Section 3.1). Some visually-aware recommenders (e.g., AlexRank) only use visual embedding (feature) to build nearest neighbor-based recommender, and other recommenders (e.g., DVBPR) model the visual content embedding together with user content embedding using dot product in a CF manner (cf. Section 3.2). For instance, in DVBPR, the inner product of user embedding θ_u and item embedding $\Phi_e(\mathbf{X}_i)$ represents the preference of user u on item i.

We propose an insider attack (INSA) in which the attacker can modify the embeddings of item images in order to increase the predicted preference score that solely determines the recommendation ranking. Specifically, INSA changes item embeddings by

adding perturbations δ on the item images. The perturbations are optimized iteratively such that the strength of preference for the item is maximized over all user profiles. In this paper, the magnitude of δ is restricted by L_∞ norm, which represents the maximum value of δ and is commonly used in computer vision research to measure imperceptibility [7, 41]. Formally, given a product image $\mathbf{X}_i$ of item i, we optimize perturbations δ to increase the preference $p_{u,i}$ of all users on item i. The optimization objectives for different recommenders are specified in Equation 6:

$$\text{AlexRank:} \quad \underset{\delta}{\text{argmax}} \sum_{u \in \mathcal{U}} \sum_{j \in I_u^+} \frac{-\|\Phi_f(\mathbf{X}_i + \delta) - \Phi_f(\mathbf{X}_j)\|^2}{|\mathcal{I}_u^+|}$$

$$\text{VBPR:} \quad \underset{\delta}{\text{argmax}} \sum_{u \in \mathcal{U}} \theta_u^T (E\Phi_f(\mathbf{X}_i + \delta)) \qquad (6)$$

$$\text{DVBPR:} \quad \underset{\delta}{\text{argmax}} \sum_{u \in \mathcal{U}} \exp(\theta_u^T \Phi_e(\mathbf{X}_i + \delta))$$

$\Phi : \mathbf{X}_i \to \theta_i$ is the feature extraction or embedding model where θ_i represents the content embedding for item i. θ_u represents the user content embedding. The optimization can be implemented by mini-batch gradient descent, and it stops when certain conditions are met, e.g., it reaches certain number of iterations.

4.2 Expert Attack (EXPA)

The medium-knowledge attack model assumes that the attacker can select a hook (i.e., popular) item. It also assumes that the attacker has access to the visual feature extraction model (see Section 3.1) and has the expertise needed to use it in a transfer learning pipeline. We propose an expert attack (EXPA) in which the attacker uses the hook item to mark the region of item space to which the adversarial item should be moved. Specifically, the EXPA attack generates perturbations added to the cooperative item in order to create the adversarial item by decreasing the representation distance to the hook item.

Formally, generating an adversarial item image by EXPA is described in Equation 7:

$$\underset{\delta}{\text{argmin}} \quad \|\Phi(\mathbf{X}_i + \delta) - \Phi(\mathbf{X_{hook}})\|_2, \qquad (7)$$

where Φ is the feature extraction or embedding model. The EXPA attack leverages the same mechanism as the targeted visual feature attack proposed by [39]. The novelty of EXPA is its use of a hook image that moves the adversarial image in image space in a way that makes it rise in personalized recommendation lists. Note that the hook image itself is not necessarily present in candidate set, which is selected by BPR, and thereby also not necessarily in the recommendation lists.

Algorithm 1 describes the process to generate adversarial product images with INSA and EXPA. $\mathbf{X}_i$ is the original image of cold item, and $\mathbf{X_{hook}}$ is the hook item. Φ is the neural network that extracts embeddings or features from the image content. Our aim is to find perturbations δ that could increase the personalized preference predictions by optimization through all user content embeddings (INSA) or targeting a hook item (EXPA). The magnitude of perturbations can be adjusted by δ. To make sure that the output images are valid with respect to standard image encoding format, a clip

---

Algorithm 1 Adversarial Item Promotion Attack

Input:
X: cold item image, X_{hook}: hook item image
δ: adversarial perturbations, ϵ: L_∞ norm bound
Φ: neural network, θ: user content embedding
K: number of iterations, $\mathcal{A}$: attack to mount (INSA or EXPA)
Output:
X': adversarial product image

1: Initialize $\boldsymbol{x}'_0 \leftarrow X$,   ▷ $\boldsymbol{x}'_k$ represents adversarial image in iteration k
2: $\boldsymbol{\delta} \leftarrow \boldsymbol{0}$
3: **for** $k \leftarrow 1$ to K **do**
4:     **if** $\mathcal{A}$ is INSA **then**
5:         AlexRank: $\boldsymbol{\delta} \leftarrow \underset{\boldsymbol{\delta}}{\arg\max} \sum_{u \in \mathcal{U}} \sum_{j \in I_u^+} \frac{-\|\Phi_f(\boldsymbol{x}'_{k-1}+\boldsymbol{\delta})-\Phi_f(X_j)\|^2}{|I_u^+|}$
6:
7:         VBPR: $\boldsymbol{\delta} \leftarrow \underset{\boldsymbol{\delta}}{\arg\max} \sum_{u \in \mathcal{U}} \theta_u^T (E\Phi_f(\boldsymbol{x}'_{k-1}+\boldsymbol{\delta}))$
8:
9:         DVBPR: $\boldsymbol{\delta} \leftarrow \underset{\boldsymbol{\delta}}{\arg\max} \sum_{u \in \mathcal{U}} \exp(\theta_u^T \Phi_e(\boldsymbol{x}'_{k-1}+\boldsymbol{\delta}))$ ▷ Eq.(6)
10:     **else if** $\mathcal{A}$ is EXPA **then**
11:         $\boldsymbol{\delta} \leftarrow \underset{\boldsymbol{\delta}}{\arg\min} \; \|\Phi(\boldsymbol{x}'_{k-1}+\boldsymbol{\delta})-\Phi(X_{hook})\|_2$ ▷ Eq.(7)
12:     **else**
13:         break
14:     **end if**
15:     $\boldsymbol{\delta} \leftarrow \text{clip}(\boldsymbol{\delta}, -\epsilon, \epsilon)$ ▷ Make sure that the magnitude of perturbations are in pre-defined L_∞ norm range
16:     $\boldsymbol{x}'_k \leftarrow \boldsymbol{x}'_{k-1} + \boldsymbol{\delta}$
17:     $\boldsymbol{x}'_k \leftarrow \text{clip}(\boldsymbol{x}'_k + \boldsymbol{\delta}, 0, 1)$ ▷ Ensure perturbed image stays in valid image range
18: **end for**
19: $\boldsymbol{x}'_k \leftarrow \text{quantize}(\boldsymbol{x}'_k)$ ▷ Ensure $\boldsymbol{x}'_k$ is valid in the 8-bit image format
20: **return** $X' \leftarrow \boldsymbol{x}'_k$ is the adversarial item image

---

function restricts adversarial item image in range [0, 1], and a quantization function ensures that the output image can be saved in the 8-bit format. The resulting adversarial image X' is the summation of the original image and the clipped perturbations.

4.3 Semantic Attack (SEMA)

The low-knowledge attack model assume nothing beyond general knowledge needed to choose hook items (see Section 3.1). We propose a semantic attack (SEMA) uses the semantic content of the image, i.e., what is shown in the image, in order to achieve the promotion of items. The attack differs considerably from INSA and EXPA, which add perturbations to existing images without changing what the images depict.

Figure 3 c-SEMA illustrates the semantic attack that we will test here, which we call compositing semantic attack (c-SEMA). With c-SEMA, the attacker creates an adversarial image by editing the original image into the hook item image as an inset. Here, the c-SEMA attack is promoting a pair of shoes and the hook item is the jeans. A text (here, "match your jeans/coat") can be included to contribute to the impression that the composite image is a fair-play attempt to raise the interest of potential customers.

Table 2: Statistics of the data sets

	#Users	#Items	#Interactions
Amazon Men	34244	110636	254870
Tradesy.com	33864	326393	655409

Figure 3 n-SEMA shows another type of semantic attack, which we call natural semantic attack (n-SEMA). Here, the integration of the hook item is natural. n-SEMA images can be created in a photo studio or a professional photo editor. We do not test them here, since creation is time consuming and we are using a cold test set of 1000 item images. However, an unscrupulous merchant would have the incentive to invest the time to create n-SEMA images. One highly successful adversarial item image could already lead to increased buyers and increased profit.

The semantic attack is particularly interesting for two reasons. SEMA achieves the change in image embeddings needed to push an adversarial image close to a hook image in image space by manipulating the depicted content of the image. First, this means that there are no limits on the quality of a SEMA adversarial image. Contrast c-SEMA and n-SEMA with the INSA and EXPA photos in Figure 3. The item is visible in the image, and consumers who decide to purchase the product will not find that they have been misled. However, not all of the images are crisp, and stronger attacks introduce artifacts affecting the perceived image quality. Second, the impact of a SEMA image attack is not dependent on the algorithm used by the recommender systems. In fact, SEMA images can effectively attack any recommender system using visual features, and not just systems using neural embedding as studied here.

5 EXPERIMENTAL SETUP

In this section, we first introduce data sets used for our experiments (Section 5.1) and introduce the details of evaluation setup including metrics (Section 5.2). Then, we describe implementation of experiments (Section 5.3).

5.1 Data

5.1.1 Data statistics. We perform our experiments on two data sets: Men's Clothing in Amazon.com and Tradesy.com, which are publicly available and widely used in visually-aware recommender system research. The statistics of the two data sets are described in Table 2. The Men's Clothing category is an important subset of the Amazon.com data set, where the effectiveness of visual features has been validated in previous work [20, 24, 32]. Tradesy.com is a C2C second-hand clothing trading website where users can buy and sell fashion items. The nature of the Tradesy.com inventory makes visually-aware cold item recommendation crucial, because of its "one-off" characteristics. For both datasets, one descriptive image is available for each item, and we follow the protocol of [24] and treat users' review histories as implicit feedback. For each user, one item is selected among all interacted items as the test item, so we have the same number of test items as the number of users.

5.1.2 Cold test item election. To validate the effectiveness of the attacks in cold start scenario, in each of Amazon Men and Tradesy.com data sets, we randomly select 1000 cold test items that no user has interacted with and leave them out as the cold test set. These cold

Figure 3: Examples of adversarial item images by different approaches. INSA attack is based on Equation 6. EXPA, c-SEMA and n-SEMA select a popular item, e.g., Levi's 501 jeans, as the hook item. c-SEMA applies simple co-depiction approach, and n-SEMA incorporates the target item in a more natural way. More details about the influence of ϵ on recommendation performance can be found in Section 6.2

items are excluded from the training process. Later, they are injected as cold-start items into the candidate item set before feeding the set into the visually-aware ranker.

5.2 Evaluation Metrics

The change in rank of cold-start items is measured by the prediction shift and the change in hit rate ($HR@N$), following the evaluation metrics for top-N recommender system robustness [33] and [5]. Equation 8 defines the average prediction shift Δ_{p_i} for item i and also the mean average prediction shift for a set of test items Δ_{set}. Our results will report $|\Delta_{\text{set}}|$ and the direction separately for clarity. $p'_{u,i}$ is the post-attack predictor score and $p_{u,i}$ is the original predictor score for item i. $\mathcal{I}_{\text{test}}$ represents the set of test items.

$$\Delta_{p_i} = \sum_{u \in \mathcal{U}} \frac{(p'_{u,i} - p_{u,i})}{|\mathcal{U}|} \quad \Delta_{\text{set}} = \sum_{i \in \mathcal{I}_{\text{test}}} \frac{\Delta_{p_i}}{|\mathcal{I}_{\text{test}}|}, \quad (8)$$

Equation 9 defines the average hit rate $HR_i@N$ for item i in terms of $H_{u,i}@N$ for item i for user u. The mean average hit rate $HR@N$ for test items averages $HR_i@N$ over the test set. $\Delta_{HR@N}$ is the change in mean average hit rate, where $HR'_i@N$ is the post-attack hit rate for item i. Our results will report $|\Delta_{HR@N}|$ and the direction separately for clarity.

$$HR_i@N = \sum_{u \in \mathcal{U}} \frac{H_{u,i}@N}{|\mathcal{U}|} \qquad HR@N = \sum_{i \in \mathcal{I}_{\text{test}}} \frac{HR_i@N}{|\mathcal{I}_{\text{test}}|}$$

$$\Delta_{HR@N} = \sum_{i \in \mathcal{I}_{\text{test}}} \frac{HR'_i@N - HR_i@N}{|\mathcal{I}_{\text{test}}|}$$

$$(9)$$

It is important to note that low metric values can still result in large impact due to the large number of users involved. For example, in Amazon Men, an increase of 0.01 on $HR@5$ means that adversarial cold items are pushed into the top-5 list of about 340 users.

5.3 Implementation

In the first stage, we use BPR to generate a candidate set of top 1000 items that are selected by personalized preference ranking. Note that we use a set, which means that the original rank order is not taken into account by the visually-aware ranker. Then we inject the ordinary test item in the top 1000 candidate set and get a set of 1001 items for each user. To compare before and after the attack, we inject one cooperative cold item or its corresponding adversarial cold item in the candidate set. So, for each cold-start item, we have two sets of 1002 items, which each include one test item and also include either one cooperative or one adversarial cold item. We use our three visual ranking models, AlexRank, VBPR, and DVBPR (see Section 3.2), to rank the 1002 items and evaluate with respect to both integrity and availability (see Section 3.3).

5.3.1 Model training. We implement BPR, AlexRank, VBPR, and DVBPR in PyTorch [36]. For the first stage model BPR, we set the number of factors to 64. Stochastic Gradient Desecent (SGD) is used to optimize BPR with learning rate 0.01 for Amazon Men and 0.5 for Tradesy.com, where the weight decay for L_2 penalty is set to 0.001 on both data sets. The feature dimension of AlexRank is 2048, and the embedding length of both VBPR and DVBPR is 100. A grid search of learning rate in $\{0.1, 0.01, 0.001, 0.005\}$ and weight decay in $\{0.001, 0.0001, 0.0001\}$ is conducted for both VBPR and DVBPR to select hyperparameters, and we select the model with best validation performance.

5.3.2 AIP attacks. If not specifically mentioned, the maximal size of perturbations ϵ for AIP attacks is set to 32. In INSA, the number of epochs is set to 10 to control the attacking time, and our implementation takes about 2 hours to generate 1000 adversarial item images on a single NVIDIA RTX 2080Ti GPU. We use the Adam optimizer with the learning rate of 0.001 for DVBPR and 0.0001 for both VBPR and AlexRank. In EXPA, the hook items are "Levi's Men's 501 Original Fit Jean" in Amazon Men and a gray coat in Tradesy.com. These two products are most commonly interacted items in training data of these two data sets. Recall, however, that

hooks can be chosen without direct access to interaction statistics. We use a Adam optimizer with a learning rate of 0.01 in EXPA, and the number of iterations is set as 5000. In c-SEMA, we resize the hook item image and paste it on the right side of the cooperative item image as shown in Figure 3. To make the combination more natural, we also add a text description. More implementation details can be found in our released code.

6 EXPERIMENTAL RESULTS

In this section, we carry out experimental analysis of our attacks using two real-world data sets (Section 6.1) and also investigate the influence of hyperparameter selections (Section 6.2). Finally, we analyze and discuss classification-based attack (Section 6.3).

6.1 Attack Evaluation

Table 3: Absolute mean average prediction shifts of adversarial cold items $|\Delta_{\text{set}}|$ on Amazon Men (AM) and Tradesy.com (TC), where $\uparrow$ represents positive prediction shift (score increased) and $\downarrow$ represents negative prediction shift (score decreased). Positive shift means a successful attack.

	AlexRank			VBPR			DVBPR		
	INSA	EXPA	c-SEMA	INSA	EXPA	c-SEMA	INSA	EXPA	c-SEMA
AM	$\uparrow$16.13	$\uparrow$15.94	$\uparrow$11.1	$\uparrow$3.27	$\uparrow$3.16	$\uparrow$0.88	$\uparrow$13.54	$\uparrow$4.80	$\uparrow$4.82
TC	$\uparrow$26.89	$\downarrow$0.67	$\downarrow$3.44	$\downarrow$0.79	$\uparrow$1.60	$\uparrow$1.45	$\uparrow$3.64	$\uparrow$1.89	$\uparrow$1.19

We mount AIP attacks and assess their effects with respect to our metrics (cf. Section 5.1.2). Table 3 shows the absolute mean average prediction shift $|\Delta_{\text{set}}|$ for adversarial cold items vs. cooperative items. The upwards arrow represents a positive prediction shift, meaning that the attack has successfully promoted the item. We see that for nearly all combinations of AIP attack and visually-aware recommender system the attack is successful. The high-knowledge attack, INSA, achieves a larger shift than EXPA and c-SEMA. c-SEMA is surprisingly successful, given the very minimal amount of knowledge that it requires. Note that it is only meaningful to compare the size of the prediction shift for the same recommender system and the same data set. A negative mean average prediction shift does not necessarily mean that an attack is unsuccessful since it is the rank position and not the preference prediction score that translates into benefit for the attacker. For this reason, we go on to examine hit rate related metrics.

Table 4(a) presents results in terms of the mean average hit rate *HR*@5. The first two rows report the original situation: the hit rate for the cooperative (i.e., not adversarial) version of the cold items and the hit rate for the ordinary test items in the case that no adversarial items have been added to the candidate set. The rest of the table reports on attacks. Cases marked with ** indicate a statistically significant difference between the original situation and the case of the attack (item-level paired sample t-test $p < 0.01$).

First, we consider "Integrity", namely, the success of the attacks in pushing items. It can be observed in Table 4 that in nearly all cases the hit rate for the adversarial version of the cold items exceeds that of the cooperative version of the cold items, meaning that all AIP attacks are generally effective. For INSA, the impact of the attack is dramatic. The cooperative cold item makes it to one of the top-5 position in the lists of only 35 users (averaged over the three

recommender system), but after the attack, the adversarial cold item makes it into the top-5 position of over 20,000 users. Since INSA uses the most knowledge, it is not surprising that it is the most effective attack. However, even with much less knowledge, both EXPA and c-SEMA pose serious threats. For example, for Amazon Men, c-SEMA pushes cold items into the top-5 list of 582 users in the case of DVBPR. We also calculated *HR*@10 and *HR*@20 for all conditions. These are not reported here since the trends were overwhelming the same as for *HR*@5.

Next, we turn to discuss "Availability", namely, the extent to which promotion occurs at the expense of other items, which at scale can impede the functioning of the entire recommender system. In Table 4, we see that INSA has a strong impact on availability than EXPA and c-SEMA.

Notice that the different performance of EXPA and c-SEMA on Amazon Men and Tradesy.com is not solely attributable to the adversarial attack itself, since the selection of the hook item also has an impact. For Tradesy.com, the *HR*@5 for the selected hook item (which is a gray coat) is originally rather low, so after EXPA or c-SEMA, the rank cannot increase dramatically. In general, we observe that AIP attacks are more damaging to integrity than to availability. However, in real-world situations it would be important to study cases involving the simultaneous presence of multiple adversarial items.

In order to directly illustrate the magnitude of the impact of AIP attacks on integrity, we report the change in mean average hit rate $|\Delta_{HR@5}|$ in Table 5(a). Again, the expected large effect of INSA can be observed. Also, again, the c-SEMA attack is surprisingly effective, given the minimal knowledge it involves. Remember that the recommender system platforms we are concerned about are enormous, and even a small boost in average rank of the magnitude of that afforded by c-SEMA could translate in to a substantial increase in interactions and profit.

Figure 4 shows a 2-D visualization (using t-SNE [30]) of the image space defined by the DVBPR item embeddings. It allows us to directly observe the influences of different AIP attacks. The position of the original cold image (i.e., a cooperative image) is shown by ①. We can see that it is positioned next to items with which it is visually similar. The attacks move this image to the other positions. The hook item is a pair of jeans. We can see that EXPA, c-SEMA, and n-SEMA push the cold item to a cluster related to the hook item. Note that it is difficult to reason about the position of INSA, since it is optimized with respect to all user embeddings. Here the n-SEMA image is manually generated using photo-editing software, in particular, after cropping and rotation, the shoes to be promoted are edited into the hook image. However, recall that in the real world an image could easily be taken of a model wearing both items.

6.2 Influence of Hyperparameters

Choice of the hyperparameters can influence the impact of the attack. Here, we take a look at the two most important hyperparameters, embedding length and attack strength (ϵ).

To study the impact of embedding length, we gradually reduce the embedding length and measure *HR*@5. Specifically, we conduct experiments for different numbers of factors (for VBPR in

Table 4: Average $HR@5$ **of cooperative (adversarial) cold item and ordinary test item in AlexRank, VBPR, DVBPR and adversarial training-based AMR. Three attacks, INSA, EXPA, and c-SEMA, are evaluated on Amazon Men and Tradesy.com data sets.** ** **indicates that the increase/decrease over ordinary test/cooperative cold test are statistically significant (** $p < 0.01$ **).**

(a) AlexRank, VBPR, and DVBPR. **(b) AMR.**

			Amazon Men			Tradesy.com			Amazon Men	Tradesy.com
Attack method	**Dimension**	**Test set**	AlexRank	VBPR	DVBPR	AlexRank	VBPR	DVBPR	AMR	AMR
–	–	Cooperative cold test	0.0011	0.0017	0.0010	0.0003	0.0022	0.0011	0.0013	0.0013
	–	Ordinary test	0.0187	0.0191	0.0195	0.0338	0.0241	0.0234	0.0289	0.0381
INSA	Integrity	Adversarial cold test	0.8296**	0.2678**	0.7211**	0.9399**	0.0783**	0.2711**	0.7476**	0.2858**
	Availability	Ordinary test	0.0140**	0.0190**	0.0170**	0.0284**	0.0224**	0.0228**	0.0238**	0.0362**
EXPA	Integrity	Adversarial cold test	0.0036**	0.0150**	0.0646**	0.0002**	0.0048**	0.0384**	0.0185**	0.0039**
	Availability	Ordinary test	0.0188**	0.0198**	0.0192**	0.0338	0.0228**	0.0232**	0.0289**	0.0381**
c-SEMA	Integrity	Adversarial cold test	0.0017**	0.0027**	0.0181**	0.0000**	0.0048**	0.0005**	0.0007**	0.0047**
	Availability	Ordinary test	0.0187**	0.0198**	0.0194**	0.0338	0.0227**	0.0234	0.0289**	0.0381**

Table 5: $|\Delta_{HR@5}|$ **before and after AIP attacks: Cold items (attack is successful if** $HR@5$ **rises).**

(a) AlexRank, VBPR, and DVBPR. **(b) AMR.**

|	Amazon Men			Tradesy.com			Amazon Men	Tradesy.com
	AlexRank	VBPR	DVBPR	AlexRank	VBPR	DVBPR	AMR	AMR
INSA adversarial cold vs. cooperative cold	↑ 0.8285	↑ 0.2661	↑ 0.7201	↑ 0.9396	↑ 0.0761	↑ 0.2700	↑ 0.7463	↑ 0.2845
EXPA adversarial cold vs. cooperative cold	↑ 0.0025	↑ 0.0133	↑ 0.0636	↓ 0.0001	↑ 0.0026	↑ 0.0373	↑ 0.0172	↑ 0.0026
c-SEMA adversarial cold vs. cooperative cold	↑ 0.0006	↑ 0.0010	↑ 0.0171	↓ 0.0003	↑ 0.0026	↓ 0.0006	↓ 0.0006	↑ 0.0034
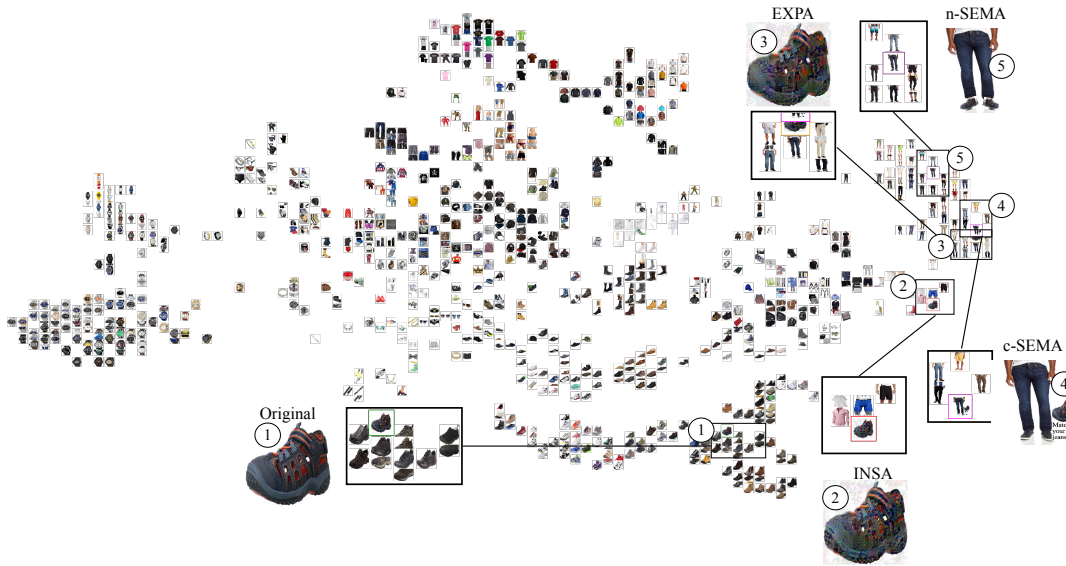
Figure 4: 2-D visualization of item embedding by pre-trained DVBPR model (#factors = 100) for randomly selected 1000 items in Amazon Men data set, cooperative item ① embedding and its corresponding adversarial item embedding. Attacks include generated images by INSA ②, EXPA ③, c- SEMA ④ and n- SEMA ⑤.

$\{20, 50, 100\}$ and for DVBPR in $\{10, 30, 50, 100\}$) with same adversarial budget (i.e., same iterations and learning rates). Results are presented in Figure 5 for VBPR and DVBPR. We discovered that the embedding length is quite important, with evidence pointing towards systems using shorter embedding length being more vulnerable to AIP attacks. This finding is valuable since without this

knowledge a visually-aware recommender systems might use short embeddings to save storage.

To study the impact of attack strength we vary the magnitude of ϵ and measure $HR@5$. We carry out experiments with ϵ in $\{4, 8, 26, 32\}$ for INSA and EXPA on subset of 100 cold item images. Results are presented in Table 6 for all three visually-aware
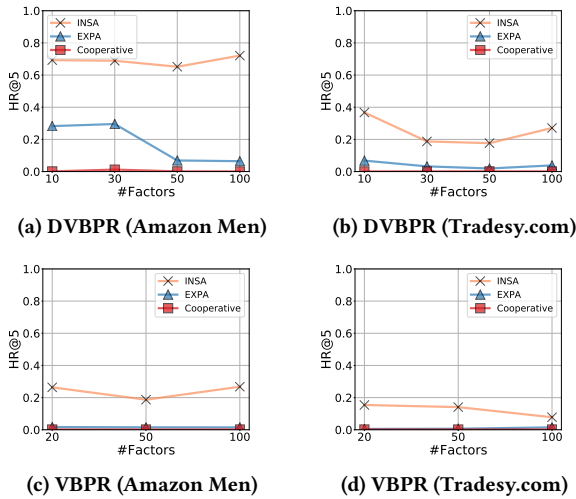
Figure 5: *HR*@5 **of cooperative cold items and adversarial cold items by INSA/EXPA with different number of factors in VBPR and DVBPR on Amazon Men's Clothing and Tradesy.com data set.**

recommender systems. We found that increasing ϵ from 4 to 32 leads to improved adversarial effects, but large perturbation size is not necessary in most cases for a successful attack. Comparably, DVBPR is more sensitive to the magnitude of ϵ than the other two approaches. Recall that EXPA has sensitivities to the choice of the hook item.

Figure 3 provides examples that correspond to different levels of ϵ. This figure confirms that it is not necessary for the adversarial modifications to be highly noticeable in an image in order for an attack to be effective.

6.3 Classifier-targeted Adversarial Images

In Section 1, we pointed out that previous work on the vulnerability of recommender systems that use images has focused on classifier-targeted adversarial examples, which are already well studied in the computer vision literature. In contrast, our EXPA and INSA approaches target the ranking mechanism, and attack the user content embedding. In this section, we confirm that our attack poses a greater threat than the classifier-targeted attack TAaMR [13]. The TAaMR attack works by calculating the most popular class of items using the data set on which the recommender system was trained. As such, it can be considered an insider attack, which, like our INSA, requires knowledge that would only be available inside the company running the recommender system platform. TAaMR also requires access to a visual feature extraction model, like our EXPA. It uses this model as a source of class definitions and to create images that are adversarial with respect to the most popular class. For our experiments we create class-targeted adversarial examples using two approaches from the computer vision literature, FGSM [18] and PGD [26], on pre-trained AlexNet [25]. We use the training data to calculate the most popular item classes. There are 'jersey, T-shirt, tee shirt' (9468 interactions) for Amazon Men and 'suit, suit of clothes' (19841 interactions) for Tradesy.com. The

inventory of classes is taken from ImageNet [12], as in [13]. Table 7 shows the change in mean average *HR*@5 for FGSM and PGD20 attack. Generally, PGD20 has larger impacts on the AlexRank, and FGSM is more effective on VBPR. Both attacks have little impact on DVBPR, because the architecture of AlexNet is fairly distinct from the CNN-F architecture [8] in DVBPR. Although it uses information comparable to that used by INSA and EXPA, the adversarial impact is on par with the c-SEMA attack (cf. Table 5). This experiment shows the importance of AIP attacks, which directly attacks the user content embeddings and thereby the ranker.

7 DEFENSE

7.1 Adversarial Training

AMR (Adversarial Multimedia Recommendation) [42] uses the preference prediction function from VBPR (cf. Equation 2) and adds on-the-fly adversarial information to the training process. Recall it was proposed to improve recommendation performance, but here we will study its potential for defending against AIP attacks. The optimization of AMR is implemented by mini-batch gradient descent. In each step, given a subset $\mathcal{D}_a$ of $\mathcal{D}_s$ (cf. Equation 3), AMR first perturbs the Θ to increase the loss:

$$\Theta' = \underset{\Theta}{\operatorname{argmax}} \sum_{(u,i,j)\in\mathcal{D}_a} -\ln\sigma(p_{u,i} - p_{u,j}) \tag{10}$$

In AMR, adversarial perturbations with respect to parameters are calculated by model gradients and added to current parameters in each step. Then it feeds forward the visual features, calculates combined loss, and back-propagates to update the parameters of the model. Specifically, $p'_{u,i}$ and $p'_{u,j}$ are calculated by the model with perturbed parameters Θ' (cf. Equation 2). Then, model parameters are updated by back-propagation based on the combined normal and adversarial loss:

$$\underset{\Theta}{\operatorname{argmin}} \sum_{(u,i,j)\in\mathcal{D}_a} -\ln\sigma(p_{u,i}-p_{u,j})-\lambda_{\text{adv}}\ln\sigma(p'_{u,i}-p'_{u,j})+\lambda_{\Theta}||\Theta||^2 \tag{11}$$

where λ_{adv} is the weight hyperparameter for adversarial loss. To train AMR, we adopt the same hyperparameters from VBPR and set $\lambda_{\text{adv}} = 1$.

Table 4(b) presents the *HR*@5 of AMR under AIP attacks and Table 5(b) presents the mean average hit rate changes. Although AMR increases the general performance by including adversarial information into training process, the *HR*@5 jumps noticeably when AIP attacks are applied, which means AMR is vulnerable to AIP attacks. Our finding here is consistent with recent research in the machine learning community [6, 44], which shows that achieving adversarial robustness is non-trivial.

7.2 Defense by Image Compression

In the computer vision literature, simple defenses have been shown to be effective against adversarial images that cause neural classifiers to misclassify [10, 14, 19, 47]. Here, we evaluated two common defenses: JPEG compression and bit depth reduction in order to test whether they are effective against AIP attacks. These are known to be able to erase the effect of image perturbations. We carried out an evaluation by applying progressively stronger versions of the

Table 6: Average $HR@5$ of a subset of 100 adversarial cold test items by INSA/EXPA with different magnitude of ϵ in DVBPR and VBPR for Amazon Men's Clothing and Tradesy.com data set. (cf. Figure 3 for adversarial item images with different ϵ)

			Amazon Men				Tradesy.com			
	Attack	Cooperative cold	$\epsilon=4$	$\epsilon=8$	$\epsilon=16$	$\epsilon=32$	Cooperative cold	$\epsilon=4$	$\epsilon=8$	$\epsilon=16$	$\epsilon=32$
AlexRank	INSA	0.0023	0.7856	0.8180	0.8258	0.8297	0.0054	0.9280	0.9351	0.9381	0.9400
	EXPA		0.0036	0.0035	0.0036	0.0036		0.0020	0.0000	0.0000	0.0000
VBPR	INSA	0.0018	0.2932	0.3159	0.3498	0.3599	0.0025	0.0734	0.0755	0.0815	0.0842
	EXPA		0.0008	0.0075	0.0135	0.0167		0.0005	0.0009	0.0009	0.0009
DVBPR	INSA	0.0007	0.0049	0.0391	0.3706	0.7039	0.0018	0.0015	0.0194	0.1116	0.2731
	EXPA		0.0021	0.0088	0.0313	0.0554		0.0025	0.0076	0.0291	0.0397

Table 7: $|\Delta_{HR@5}|$ before and after TAaMR attack: Cold items (attack is successful if $HR@5$ rises) and ordinary test items (successful attacks cause a drop in $HR@5$) .

		Amazon Men			Tradesy.com		
		AlexRank	VBPR	DVBPR	AlexRank	VBPR	DVBPR
Integrity dimension	FGSM adversarial cold vs. cooperative cold	↓ 0.0010	↑ 0.0014	↑ 0.0070	↓ 0.0002	↑ 0.0183	↓ 0.0008
	PGD20 adversarial cold vs. cooperative cold	↑ 0.0100	↓ 0.0011	↓ 0.0005	↑ 0.0019	↓ 0.0014	↓ 0.0004
Availability dimension	FGSM ordinary test vs. ordinary test	↑<< 0.0001	↓<< 0.0001	↑ 0.0016	↓<< 0.0001	↓<< 0.0001	↑<< 0.0001
	PGD20 ordinary test vs. ordinary test	↓<< 0.0001	↑<< 0.0001	↑ 0.0016	↑<< 0.0001	↓<< 0.0001	↑<< 0.0001

Table 8: Visualization of the level at which a defense is successful at lowering the $HR@5$ of an adversarial cold item equal or less the average $HR@5$ of a cooperative cold item. For JPEG compression, levels are specified as compression percents and for bit depth reduction levels are specified as number of bits with which the image is encoded. (▲: Amazon.com; ★: Tradesy.com)

		JPEG compression					Bit depth reduction					
		90	70	50	30	10	7	6	5	4	3	2
AlexRank	INSA											
	EXPA										▲	
VBPR	INSA											
	EXPA	▲★									▲★	
DVBPR	INSA						★					
	EXPA					▲						▲★

8 CONCLUSION AND OUTLOOK

This paper has investigated the vulnerability at the core of Top-N recommenders that use images to address cold start. We have shown that Adversarial Item Promotion (AIP) attacks allow unscrupulous merchants to artificially raise the rank of their products when a visually-aware recommender system is used for candidate ranking. Our investigation has led us to conclude that AIP attacks are a potential threat with clear practical implications. Compared with existing profile injection attacks [5, 27, 33, 35] and poisoning attacks [9, 15, 16, 29, 42] that promote items by injecting fake profiles, AIP only needs to modify the descriptive image of the item. Effective AIP attacks are easy to mount, as demonstrated by the minimal scale attack that we have studied here (cf. attack model in Section 3.1). In short, our work reveals that the promise of hybrid recommender systems to provide a higher degree of robustness [33] is not an absolute, and that we must proceed with caution when using images to address cold start.

Future work should dive more deeply into connection between adversarial items and user experience with the recommender system. One aspect is the relevance of adversarial items to users. Like any cold start item, users click an adversarial cold start item because it piques their interest. As an adversarial item accumulates more clicks, and enters more users' personalized lists, the main issue may be not be relevance, but rather fair competition with other potentially relevant items.

Another aspect related to user experience is the impact of image quality. If users have sensitivities that cause them to avoid products with images affected by perturbations, then attackers would need to back off to weaker attacks that make perturbations unnoticeable. In this case, defenses such as adversarial training could be more effective. More work is needed to understand approaches such as SEMA, which do not involve trading off image quality and attack strength. Alternatively, approaches that make adversarial images effective yet non-suspicious, such as [22, 48], can also be studied.

Future work must develop effective defenses against AIP attacks. An approach that easily comes to mind is the use a gatekeeper classifier to flag adversarial images at the moment that merchants

defense to the 100-item subset of our larger test set that was previously selected. We do not evaluate SEMA, since the semantic attack does not involve perturbations and if these defenses would destroy the effectiveness of SEMA they would destroy the usefulness of all images to the recommender system.

In Table 8, we visualize the level of strength of defense that must be applied to the adversarial image in order for its rank to be lowered to the average $HR@5$ of a cooperative image. If the defenses presented effective protection against adversarial item images then we would expect the ▲ and ★ to appear consistently to the far left in the boxes. This is clearly not the case. We see in Table 8 that INSA is more difficult to defeat than EXPA, which is expected because it leverages insider knowledge. However, EXPA is clearly not easy to beat across the board. It is important to note that this test is a strong one. If these defenses would be applied in practice, they would need to be applied to all images and not just adversarial images. Image content becomes indistinguishable as compression increases, and an image 10% the size of the original image or encoded with only 2-3 bits can be expected to contain little to no item information.

upload them. It is clear that for SEMA such a classifier would be difficult to build, since SEMA attacks are created in a natural manner and are indistinguishable from cooperative images. For INSA and EXPA, a gateway filter could be built if the exact specifications of the adversarial attack, including the parameter settings, were known. However, we need to be aware that in the worst case scenario where the information of the gatekeeper is available (i.e., white-box scenario), variants on INSA and EXPA can still bypass such a classifier by constructing new loss functions [7].

We have shown in our paper (cf. Section 7.1) that simply incorporating on-the-fly adversarial information into model training cannot guarantee a robust recommender. In addition, adversarial training requires strict hypothesis about the attack strength (ϵ) [31], and it also needs vast computational resources in practice [46]. Therefore, building a robust visually-aware recommender system is non-trivial and needs more research attention.

Future work must look at the impact of multipliers. If a single item has multiple descriptive images, attacks are more likely to go unnoticed, in particular semantic attacks that require no perturbations. Further, multiple merchants (or fake merchant profiles) could collaborate in a collusion attack. Finally, we note that although, here, we have focused on e-commerce, entertainment recommender systems are vulnerable: an adversarial signal could be embedded into a thumbnail or the content itself. In sum, AIP attacks constitute an important, practical risk of using images in recommender systems and serious challenges remain to be addressed.

ACKNOWLEDGMENTS

REFERENCES

[1] Marco Barreno, Blaine Nelson, Anthony D Joseph, and J Doug Tygar. 2010. The security of machine learning. *Machine Learning* 81, 2 (2010), 121–148.

[2] Alexandra Berzon, Shane Shifflett, and Justin Scheck. 2019. Amazon has ceded control of its site. The result: Thousands of banned, unsafe or mislabeled products. https://www.wsj.com/articles/amazon-has-ceded-control-of-its-site-the-result-thousands-of-banned-unsafe-or-mislabeled-products-11566564990 Accessed: 2020-10-19.

[3] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. 2013. Evasion attacks against machine learning at test time. In *ECML-PKDD*. Springer, 387–402.

[4] Christian Bracher, Sebastian Heinz, and Roland Vollgraf. 2016. Fashion DNA: Merging content and sales data for recommendation and article mapping. In *KDD Fashion Workshop*.

[5] Robin Burke, Michael P O'Mahony, and Neil J Hurley. 2015. Robust collaborative recommendation. In *Recommender systems handbook*. Springer, 961–995.

[6] Nicholas Carlini, Anish Athalye, Nicolas Papernot, Wieland Brendel, Jonas Rauber, Dimitris Tsipras, Ian Goodfellow, Aleksander Madry, and Alexey Kurakin. 2019. On evaluating adversarial robustness. *arXiv preprint arXiv:1902.06705* (2019).

[7] Nicholas Carlini and David Wagner. 2017. Adversarial examples are not easily detected: Bypassing ten detection methods. In *ACM Workshop on Artificial Intelligence and Security*. 3–14.

[8] Ken Chatfield, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Return of the devil in the details: Delving deep into convolutional nets. *BMVC*.

[9] Konstantina Christakopoulou and Arindam Banerjee. 2019. Adversarial attacks on an oblivious recommender. In *RecSys*. 322–330.

[10] Nilaksh Das, Madhuri Shanbhogue, Shang-Tse Chen, Fred Hohman, Siwei Li, Li Chen, Michael E Kounavis, and Duen Horng Chau. 2018. Shield: Fast, practical defense and vaccination for deep learning using jpeg compression. In *KDD*. 196–204.

[11] James Davidson, Benjamin Liebald, Junning Liu, Palash Nandy, Taylor Van Vleet, Ullas Gargi, Sujoy Gupta, Yu He, Mike Lambert, Blake Livingston, et al. 2010. The YouTube video recommendation system. In *RecSys*. 293–296.

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale image database. In *CVPR*. 248–255.

[13] Tommaso Di Noia, Daniele Malitesta, and Felice Antonio Merra. 2020. TAaMR: Targeted adversarial attack against multimedia recommender systems. In *International Workshop on Dependable and Secure Machine Learning*.

[14] Gintare Karolina Dziugaite, Zoubin Ghahramani, and Daniel M Roy. 2016. A study of the effect of jpg compression on adversarial images. *arXiv preprint arXiv:1608.00853*.

[15] Minghong Fang, Neil Zhenqiang Gong, and Jia Liu. 2020. Influence function based data poisoning attacks to top-n recommender systems. In *WWW*. 3019–3025.

[16] Minghong Fang, Guolei Yang, Neil Zhenqiang Gong, and Jia Liu. 2018. Poisoning attacks to graph-based recommender systems. In *ACSAC*. 381–392.

[17] João Gomes. 2017. Boosting recommender systems with deep learning. In *RecSys*. 344–344.

[18] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *ICLR*.

[19] Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens Van Der Maaten. 2018. Countering adversarial images using input transformations. *ICLR*.

[20] Ruining He and Julian McAuley. 2016. VBPR: Visual bayesian personalized ranking from implicit feedback. In *AAAI*. 144–150.

[21] Vignesh Jagadeesh, Robinson Piramuthu, Anurag Bhardwaj, Wei Di, and Neel Sundaresan. 2014. Large scale visual recommendations from street fashion images. In *KDD*. 1925–1934.

[22] Ameya Joshi, Amitangshu Mukherjee, Soumik Sarkar, and Chinmay Hegde. 2019. Semantic Adversarial Attacks: Parametric Transformations That Fool Deep Classifiers. In *ICCV*. 4773–4783.

[23] Yannis Kalantidis, Lyndon Kennedy, and Li-Jia Li. 2013. Getting the look: Clothing recognition and segmentation for automatic product suggestions in everyday photos. In *ICMR*. 105–112.

[24] Wang-Cheng Kang, Chen Fang, Zhaowen Wang, and Julian McAuley. 2017. Visually-aware fashion recommendation and design with generative image models. In *ICDM*. 207–216.

[25] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *NeurIPS*. 1097–1105.

[26] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. 2017. Adversarial examples in the physical world. In *ICLR*.

[27] Shyong K Lam and John Riedl. 2004. Shilling recommender systems for fun and profit. In *WWW*. 393–402.

[28] Joonseok Lee and Sami Abu-El-Haija. 2017. Large-scale content-only video recommendation. In *International Conference on Computer Vision Workshops (ICCVW)*. 987–995.

[29] Bo Li, Yining Wang, Aarti Singh, and Yevgeniy Vorobeychik. 2016. Data poisoning attacks on factorization-based collaborative filtering. In *NeurIPS*. 1885–1893.

[30] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research* 9, Nov (2008), 2579–2605.

[31] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards deep learning models resistant to adversarial attacks. In *ICLR*.

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

[33] Bamshad Mobasher, Robin Burke, Runa Bhaumik, and Chad Williams. 2007. Toward trustworthy recommender systems: An analysis of attack models and algorithm robustness. *ACM Transactions on Internet Technology* 7, 4 (2007), 23.

[34] James Neve and Ryan McConville. 2020. ImRec: Learning reciprocal preferences using images. In *RecSys*. 170–179.

[35] Michael P O'Mahony, Neil J Hurley, and Guenole CM Silvestre. 2002. Promoting recommendations: An attack on collaborative filtering. In *DEXA*. 494–503.

[36] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in PyTorch. In *NIPS Autodiff Workshop*.

[37] Bobby Prévost, Jonathan Laflamme Janssen, Jaime R Camacaro, and Carolina Bessega. 2018. Deep inventory time translation to improve recommendations for real-world retail. In *RecSys*. 195–199.

[38] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian personalized ranking from implicit feedback. In *UAI*. 452–461.

[39] Sara Sabour, Yanshuai Cao, Fartash Faghri, and David J Fleet. 2016. Adversarial manipulation of deep representations. In *ICLR*.

[40] Sandhya Sachidanandan, Richard Luong, and Emil S Joergensen. 2019. Designer-driven add-to-cart recommendations. In *RecSys*. 525–525.

[41] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. In *ICLR*.

[42] Jinhui Tang, Xiaoyu Du, Xiangnan He, Fajie Yuan, Qi Tian, and Tat-Seng Chua. 2020. Adversarial training towards robust multimedia recommender system. *IEEE Transactions on Knowledge and Data Engineering* 32, 5 (2020), 855–867.

[43] Jiaxi Tang, Hongyi Wen, and Ke Wang. 2020. Revisiting adversarially learned injection attacks against recommender systems. In *RecSys*. 318–327.

[44] Florian Tramer, Nicholas Carlini, Wieland Brendel, and Aleksander Madry. 2020. On adaptive attacks to adversarial example defenses. *NeurIPS* (2020).

[45] Jizhe Wang, Pipei Huang, Huan Zhao, Zhibo Zhang, Binqiang Zhao, and Dik Lun Lee. 2018. Billion-scale commodity embedding for e-commerce recommendation in Alibaba. In *KDD*. 839–848.

[46] Cihang Xie and Alan Yuille. 2020. Intriguing properties of adversarial training at scale. In *ICLR*.

[47] Weilin Xu, David Evans, and Yanjun Qi. 2017. Feature squeezing: Detecting adversarial examples in deep neural networks. In *NDSS*. 849–853.

[48] Zhengyu Zhao, Zhuoran Liu, and Martha Larson. 2020. Towards large yet imperceptible adversarial image perturbations with perceptual color distance. In *CVPR*. 1039–1048.