Retrieval-Augmented Review Generation for Poisoning Recommender Systems

Shiyi Yang, Xinshu Li, Guanglin Zhou, Chen Wang, Xiwei Xu, *Senior Member, IEEE*, Liming Zhu, *Senior Member, IEEE*, Lina Yao, *Senior Member, IEEE*

***Abstract*—Recent studies have shown that recommender systems (RSs) are highly vulnerable to data poisoning attacks, where malicious actors inject fake user profiles, including a group of well-designed fake ratings, to manipulate recommendations. Due to security and privacy constraints in practice, attackers typically possess limited knowledge of the victim system and thus need to craft profiles that have transferability across black-box RSs. To maximize the attack impact, the profiles often remains imperceptible. However, generating such high-quality profiles with the restricted resources is challenging. Some works suggest incorporating fake textual reviews to strengthen the profiles; yet, the poor quality of the reviews largely undermines the attack effectiveness and imperceptibility under the practical setting.**

To tackle the above challenges, in this paper, we propose to enhance the quality of the review text by harnessing in-context learning (ICL) capabilities of multimodal foundation models. To this end, we introduce a demonstration retrieval algorithm and a text style transfer strategy to augment the navie ICL. Specifically, we propose a novel practical attack framework named RAGAN to generate high-quality fake user profiles, which can gain insights into the robustness of RSs. The profiles are generated by a jailbreaker and collaboratively optimized on an instructional agent and a guardian to improve the attack transferability and imperceptibility. Comprehensive experiments on various real-world datasets demonstrate that RAGAN achieves the state-of-the-art poisoning attack performance.

***Index Terms*—adversarial learning, recommender systems, poisoning attacks, shilling attacks, neural networks, deep learning**

I. Introduction

Recommender systems (RSs) mitigate information overload by mining users' historical interactions to help them discover desirable products. At the same time, they benefit businesses by increasing customer retention and boosting sales through product promotion [1]. As a result, RSs have been widely deployed in various platforms such as e-commerce (e.g., Amazon, Taobao, Yelp) and social media (e.g., Facebook, YouTube, LinkedIn) [2]. The great influences of RSs on individuals/organizations and their openness nature [3] provide both incentives and conveniences for unscrupulous parties to poison RSs. Specifically, attackers can inject a group of well-crafted fake user profiles into the training data of target RSs, including user-item interaction data, to mislead the RSs for certain malicious purposes (e.g., to promote their own products for profits). This kind of attacks is known as data poisoning attacks [4]. Many academic studies [5] and practical applications [6] have shown that real-world RSs are highly vulnerable to such attacks. The attacks undermine users' trust in RSs and lead to unfair competition among businesses.

To proactively identify potential risks before real attacks, gain insight into the robustness of existing RSs, and drive the development of effective defensive measures, increasing efforts have been devoted to studying how RSs can be attacked [1]–[20]. However, existing attack methods are *still far from practical*, which may *overestimate* the robustness of RSs in realistic scenarios [19]. As discussed in previous works [21], a practical attack should meet three criteria: 1) *Knowledge-restricted settings:* In real-world RSs, attackers usually have limited knowledge of the target systems, including their architectures, parameters, and training data, due to system complexity, frequent updates, and privacy protections [3], [6], [11]; 2) *Transferability:* Since the actual target RS is typically treated as a black-box [19], fake user profiles generated by the attack should have transferability (i.e., effective against different black-box RSs and training data); and 3) *Imperceptibility:* Malicious profiles are usually designed to closely resemble legitimate ones [15], making them more difficult to detect and hence extending both the duration and scope of their impact in practice.

Existing studies satisfy only a subset of the above outlined criteria, preventing them from forming a comprehensive and robust solution. Conventional poisoning methods [7], [16], [22] primarily focus on optimizing attack effectiveness for specific collaborative filtering (CF) models [23], but often neglect stealthiness, making them more detectable. To mitigate this, recent studies have leveraged generative adversarial networks (GANs) [24] to produce more inconspicuous perturbations. However, excessive emphasis on imperceptibility may compromise the transferability of attacks [10]. To address this trade-off, subsequent works [6], [11] have integrated transferability optimization into the GAN framework, thereby improving both objectives simultaneously. By producing profiles containing only fake numerical ratings, these attacks have semantic gaps with contemporary RSs enhanced by textual reviews [25]. To solve this, we propose to incorporate textual reviews, which is a publicly available natural feature and contains rich semantic information reflecting user behaviors and item characteristics [26], to reinforce attack profiles, as demonstrated in the previous work R-Trojan [21]. Nevertheless, the generated

Shiyi Yang is with University of New South Wales and CSIRO's Data61, Sydney, Australia, shiyi.yang1@unsw.edu.au.

Xinshu Li is with Macquarie University, Sydney, Australia, xinshu.li@mq.edu.au.

Guanglin Zhou is with University of Queensland, Brisbane, Australia, guanglin.zhou@uq.edu.au.

Chen Wang, Xiwei Xu, Liming Zhu, Lina Yao are with CSIRO's Data61 and University of New South Wales, Sydney, Australia, chen.wang, xiwei.xu, liming.zhu, lina.yao@data61.csiro.au.

reviews are of low quality, characterized by logical gaps, incoherence, factual inaccuracies, sparse content, and so on, rendering them noticeable and weakening the effectiveness of transferability optimization. Moreover, R-Trojan makes an unrealistic assumption that the full data can be accessed.

To tackle the above challenges, we propose a novel practical attack framework named RAGAN for generating high-quality fake user profiles within the limited accessible resources, including partial data. The attack profiles are constructed with numerical ratings and enhanced with semantically meaningful textual reviews. These reviews provide plausible justifications for the ratings, enhancing the imperceptibility of the actual attack, and convey persuasive content that influences user decisions, thereby improving the real attack's effectiveness.

RAGAN is a tailored GAN improved by multimodal foundation models (FMs), composed of a jailbreaker, an instructional agent and a guardian. The jailbreaker aims to produce high-quality attack profiles, where the ratings are produced through pattern learning, and the corresponding reviews are generated by harnessing in-context learning (ICL) capabilities of multimodal foundation models [27], [28]. To improve the quality of the reviews to strengthen the profiles' quality, we introduce a demonstration retrieval algorithm and a text style transfer strategy to augment the naive ICL. The instructional agent is introduced to optimize the profiles to improve the transferability of the attack by providing constructive feedback to the jailbreaker. To enhance the accuracy of the feedback under the limited knowledge, we adopt review-based RSs for building agents, due to review text can finely reflect user behaviors and characteristics of their preferred items, as compared to rating-only-based RSs [6], [29]. Similarly, the guardian is a review-based detector, which is designed to distinguish the generated attack profiles from the benign profiles as much as possible for optimizing the attack imperceptibility.

Our main contributions are summarized as follows:

- We propose a new attack framework named RAGAN, which can generate transferable and imperceptible fake user profiles under a practical setting.
- We propose a demonstration retrieval algorithm and a text style transfer strategy to harness the in-context learning for high-quality attack profile generation.
- We comprehensively demonstrate the effectiveness of semantically rich textual reviews in improving the quality of attack profiles, even within the limited resources.
- Extensive experimental results on real-world datasets show that RAGAN significantly outperforms state-of-the-art attack methods in terms of transferability and imperceptibility.

II. Related Works

RSs are highly vulnerable to data poisoning attacks, aka shilling attacks, due to their openness (i.e., the training data of RSs is usually publicly accessible) [5], [30]. Researchers have successfully performed shilling attacks against real-world RS such as YouTube, Google Search, Amazon and Yelp in experiments [6]. Large companies (e.g., Amazon, Sony, eBay) have also reported that they have suffered from such attacks in practice [6], [13]. In poisoning attacks, a number of well-crafted fake user profiles can be injected to the victim RS to promote/demote an attacker-chosen item. Considering that demoting a target item can be implemented by promoting other items [16], current works focus mainly on item promotions for simplicity [30]. The items in an attack profile have four different types [4]: selected items, filler items, unrated items and the target item. To be specific, the selected items are determined by the attacker and a rating function that reflect the characteristics of the attack, and are optionally included in the profiles. The filler items make the fake user profile resemble the real user profile, reducing the chance of being detected. The target item is chosen by the attacker for promotion or demotion, and the remaining items are unrated. Based on the modality of the generated profiles, attacks can be categorized into two types: 1) *unimodal attacks* that involve only numerical ratings or textual reviews, and 2) *multimodal attacks* that incorporate both ratings and review text, leading to higher-quality profiles with enhanced realism and manipulative power.

A. Unimodal Attacks on Recommender Systems

Traditional RSs often use CF methods such as KNN [31], AR [32] and MF [33] to model user-item interaction matrix and predict users' preference on items, where MF has been widely adopted because of its performance and flexibility [23]. Conventional shilling attacks [4] such as Random and Bandwagon [22] rely on global statistics and work mainly for the traditional CFs (e.g., user-based KNN) [6]. These methods are simple heuristics based and not transferable among different RSs [11] (e.g., for item-based KNN [34]). The lack of diversity in data generation makes them easy to be detected [6].

Some data poisoning attacks are then proposed to optimize for specific types of RSs, such as PGA [7] for the MF-based, Co-visitation Injection [8] for the AR-based and a black-box poisoning strategy designed for the graph-based [9]. These attacks typically outperform conventional shilling attacks on the RSs they are tailored for. As shown in Table I, [8] and [9] explore the transferability of poisoning attacks, with a focus on the effectiveness of the attacks among traditional black-box RSs. Traditional CF methods are gradually replaced by non-linear deep learning (DL) paradigms [23], due to their restricted abilities to capture complex structure of massive interaction data. The typical approaches are NCF [35] and LightGCN [36]. However, there is a gap between existing traditional algorithm-specific attacks [7]–[9] and the growing prevalence of DL-based paradigms. To fill this gap, Huang et al. [16] propose a data poisoning attack tailored for DL-based RSs. Nevertheless, they demonstrate the transferability of the attack under unrealistic assumptions, as shown in Table I.

Textual reviews, as a natural source of data containing rich semantic information, have been increasingly integrated into DL-based RSs to address the data sparsity and cold-start issues, enhance recommendation accuracy and improve model interpretability [25], [37]. A representative example is DeepCoNN [26], which jointly models user and item reviews to learn more expressive latent representations. Chiang et al. [20] fine-tune GPT-2 [38] through reinforcement learning to generate fake textual reviews for prediction shifts in review-based

TABLE I: Comparison of data poisoning attack approaches, where $|\mathcal{U}|$ and $|\mathcal{V}|$ denote the number of users and items, respectively, l is the length of recommendation list and p represents the maximum percentage of the data that the attack requires. Note that several methods (e.g., TrialAttack and RegUP) propose multiple attacks that require different levels of knowledge of the target RS, and we choose the attacks require the least knowledge for comparison due to they are closest to real-world settings.

Attack Method		Knowledge of Target Recommendation Systems			Attack Objectives		Multimodal Profiles?
		Training Data	RS Parameters	RS Architectures	Discuss transferability or not?	Discuss imperceptibility or not?	
Conventional	Random	$\|\mathcal{U}\| \cdot \|\mathcal{V}\|$	×	×	×	×	×
	Bandwagon	$\|\mathcal{U}\| \cdot \|\mathcal{V}\|$	×	×	×	×	×
Algorithm-Specific	MF-based	$\|\mathcal{U}\| \cdot \|\mathcal{V}\|$	✓	✓	×	×	×
	AR-based	$l \cdot \|\mathcal{V}\|$	×	×	✓	×	×
	Graph-based	$\|\mathcal{U}\| \cdot \|\mathcal{V}\|$	×	×	✓	✓	×
	DL-based	$\|\mathcal{U}\| \cdot \|\mathcal{V}\|$	×	✓	✓	✓	×
	Review-based	$\|\mathcal{U}\| \cdot \|\mathcal{V}\|$	×	×	✓	✓	×
GAN-based	DCGAN	$\|\mathcal{U}\| \cdot \|\mathcal{V}\|$	×	×	×	✓	×
	AUSH	$\|\mathcal{U}\| \cdot \|\mathcal{V}\|$	×	×	✓	✓	×
	TrialAttack	$p \cdot \|\mathcal{U}\| \cdot \|\mathcal{V}\|$	×	✓	×	✓	×
	RecUP	$p \cdot \|\mathcal{U}\| \cdot \|\mathcal{V}\|$	×	×	×	✓	×
	Leg-UP	$\|\mathcal{U}\| \cdot \|\mathcal{V}\|$	×	×	✓	✓	×
	R-Trojan	$\|\mathcal{U}\| \cdot \|\mathcal{V}\|$	×	×	✓	✓	✓
	RAGAN	$p \cdot \|\mathcal{U}\| \cdot \|\mathcal{V}\|$	×	×	✓	✓	✓

RSs. However, their method 1) requires that the feedback from RSs is periodically available, which is impractical for most RSs [6]; and 2) generates the review-only profiles, which is not applicable to rating-based DL methods. Since we focus on promoting attacks aligned with Top-N recommendation scenarios prevalent in real-world applications, such a shilling attack targeting rating predictions and inducing prediction shift falls outside the scope of this paper. In addition, these algorithm-specific attacks focus solely on attack performance optimization, making their imperceptibility limited.

Recently, some efforts have been made to leverage GAN for effective profile generation, e.g., DCGAN [10], AUSH [11], TrialAttack [18], RecUP [15] and Leg-UP [6], as illustrated in the table. DCGAN, TrialAttack and RecUP do not discuss the effectiveness of attacks on different black-box RSs. As pointed out in [3], it is impractical to assume that the entire training data of the target RS can be obtained by the attacker in the real world. The possible reasons are 1) service providers may restrict one's access to full data or even perturb these data to enhance user privacy and 2) the attacker's ability to collect massive data may be limited by his resources and/or the platform's data volume restriction. Consequently, these issues may make AUSH and Leg-UP suffer transferability degradation in practice due to their full data assumptions. In addition, these GAN-based methods are unable to transferable to review-based RSs, and are overestimated detection escape capabilities by assuming that all the data are available.

B. Multimodal Attacks on Recommender Systems

We propose R-Trojan [21] to mitigate the most above problems, which is a tailored transformer-powered GAN for generating attack profiles containing not only numerical ratings but also textual reviews. However, as illustrated in Table I, R-Trojan makes an unrealistic assumption of full access to the learning data, and relies on the fine-tuned GPT-2 [38] with simple prompts (i.e., topics + sentiments) to generate fake textual reviews. However, GPT-2-generated text suffers from issues such as factual inaccuracy, repetitive phrasing, semantic confusion and logical inconsistencies, leading to the low-quality profiles. To solve the problems, RAGAN introduces a more efficient and effective generation scheme. By harnessing powerful FMs [39] that pre-trains on extensive and diverse data and is built upon flexible and advanced transformers, it makes the reviews semantically meaningful, coherent and natural, thereby improving the transferability and imperceptibility of attack profiles under a more practical setting.

III. Problem Formulation

In this section, we first introduce necessary preliminaries and notations used for the following works, then present our threat model, and finally formulate our poisoning attack RAGAN as a bi-level optimization problem.

A. Preliminaries and Notations

We use $\mathcal{M} = \{m_{u,v} : u \in \mathcal{U}, v \in \mathcal{V}\}$ to denote the records of the user-item interaction matrix in the large-scale recommendation space, where $\mathcal{U}$ and $\mathcal{V}$ represent the set of real users and the item universe, respectively. An entry $m_{u,v}$ in $\mathcal{M}$ is $(r_{u,v}, \delta_{u,v})$, in which u is the row No. and v is the column No. $r_{u,v}$ is the numerical rating from user u on item v and $\delta_{u,v}$ is the corresponding review text. $\mathcal{V}_u = \{v \in \mathcal{V} : r_{u,v} \neq 0\}$ indicates the set of items that have been interacted by u (i.e., the user's profile). Similarly, $\mathcal{U}_v = \{u \in \mathcal{U} : r_{u,v} \neq 0\}$ represents the set of users that have rated v. Similar to the benign matrix $\mathcal{M}$, we adopt $\widetilde{\mathcal{M}} = \{(r_{\widetilde{u},v}, \delta_{\widetilde{u},v}) : \widetilde{u} \in \widetilde{\mathcal{U}}, v \in \mathcal{V}\}$ to indicate the fake matrix, where $\widetilde{\mathcal{U}}$ is the fake user pool, and $r_{\widetilde{u},v}$ and $\delta_{\widetilde{u},v}$ are, respectively, fake numerical ratings and fake textual reviews that will be generated and optimized by RAGAN. Thus, each row of $\widetilde{\mathcal{M}}$ is a fake user profile. The target item is represented by t. Moreover, $\mathcal{D} = \{(x_v, d_v, n_v) : v \in \mathcal{V}\}$ is used to denote the raw metadata of the items, where x_v is the image of the item v, d_v is the textual description of the item v, n_v is the title of the item v.

B. Threat Model

*1) **Attacker's Objective**:* As discussed in Section I, we consider two practical properties of an RS attack method.

1) *Transferability.* Let $h(t)$ to denote the hit ratio of a target item t, which is the percentage of normal users whose the recommendation lists include t after the attack, i.e., $h(t) = \sum_{u \in \widehat{\mathcal{U}}_t} \frac{\mathbb{I}(t \in \widehat{\mathcal{V}}_{u,:k})}{|\widehat{\mathcal{U}}_t|}$, where $\mathbb{I}(\cdot)$ is an indicator function, $\widehat{\mathcal{U}}_t = \{u \in \mathcal{U} : r_{u,t} = 0\}$ is the set of normal users (i.e., target users who have not yet interacted with t) and $\widehat{\mathcal{V}}_{u,:k}$ is the top-k recommendation list of the normal user u. Our first objective is to maximize $h(t)$ on the black-box RSs.
2) *Imperceptibility.* Our second attack objective is to make our attack as imperceptible as possible to maximize the attack impact and the number of affected users.

*2) **Attacker's Knowledge**:* We assume that the attacker is able to access to p *percent of the training data* $\mathcal{M}$ *of the victim RS*, where $p = 1$ denotes full knowledge of data and $0 < p < 1$ indicates partial knowledge of data. This is realistic considering the openness nature [5] of a RS. Although our attack makes use of item metadata to reinforce the attack, such information is often publicly available, for example, the images/descriptions/titles on Amazon can be collected by directly browsing or writing crawlers. In addition, since real-world RSs are usually complex and flexible (e.g., they may adopt ensemble schemes and may be updated frequently), it is difficult to gain accurate knowledge (e.g., model architectures, parameters and implementation details) of the victim RSs [6]. We therefore consider black-box settings for the victim RSs.

*3) **Attacker's Capability**:* To avoid being detected while conducting attacks under a budget, an attacker often injects a limited number of profiles [6] and interacts with only a few items in each profile [16]. As a result, an attacker's capabilities are restricted by the attack size (i.e., the number of injected fake user profiles $|\widetilde{\mathcal{U}}|$) and the profile size (i.e., the number of interactive items in each fake profile).

C. Formulate Attacks As An Optimization Problem

We formulate the attack as a bi-level optimization problem, as shown in Fig. 1(a), and solve for this problem to obtain high-quality fake user profiles, as inspired by [6], [21]. Our attack consists of a jailbreaker, an instructional agent and a guardian. *The lower-level* computes the optimal parameters of the agent (denoted by Θ) and the guardian (denoted by Φ) with the given benign matrix $p\mathcal{M}$ and fake matrix $\widetilde{\mathcal{M}}$. *The upper-level* optimizes $\widetilde{\mathcal{M}}$ to maximize the attack objectives based on model parameters obtained by solving the lower-level problem.

$$\begin{aligned} &\min_{\widetilde{\mathcal{M}}} \ \lambda \mathcal{L}_{\text{trans}}(\widehat{\mathcal{M}_\Theta}) + (1-\lambda)\mathcal{L}_{\text{imper}}(\widehat{\widetilde{\mathcal{M}}_\Phi}) \\ &\text{subject to } \Theta = \arg\min_{\Theta} \mathcal{L}_{\text{agent}}(\mathcal{M}^*, \widehat{\mathcal{M}^*_\Theta}) \\ &\quad \text{and } \Phi = \arg\max_{\Phi} \mathcal{L}_{\text{guardian}}(\mathcal{M}^*, \widehat{\mathcal{M}^*_\Phi}), \end{aligned} \tag{1}$$

where $\mathcal{M}^* = \text{concatenation}(p\mathcal{M}; \widetilde{\mathcal{M}})$, and $\widehat{\mathcal{M}^*_\Theta}$ and $\widehat{\mathcal{M}^*_\Phi}$ are predictions from the corresponding models with parameters Θ and Φ, respectively. $\mathcal{L}_{\text{agent}}$ and $\mathcal{L}_{\text{guardian}}$ denote the training objectives of the related modules. $\mathcal{L}_{\text{trans}}$ is the transferability objective defined on normal user's predictions $\widehat{\mathcal{M}_\Theta}$, $\mathcal{L}_{\text{imper}}$ is the imperceptibility objective defined on fake user's predictions $\widehat{\widetilde{\mathcal{M}}_\Phi}$ and λ is a configurable parameter to adjust the trade-off of two objectives when optimized simultaneously. For simplicity, we use $\mathcal{M}$ to represent the partially observed matrix $p\mathcal{M}$ in the following formulations.

IV. RAGAN

In this section, we present the practical attack framework RAGAN for high-quality fake user profile generation. RAGAN enhances the quality (i.e., transferablity and imperceptibility) of the profiles within the limited accessible resources by introducing publicly available textual reviews and further reinforce the profiles by improving the quality of the textual reviews. Fig. 1 shows an overview of RAGAN. RAGAN is a tailored GAN that is composed of a jailbreaker, an instructional agent and a guardian, as shown in the figure. The design of each module is elaborated as follows.

A. Jailbreaker

Jailbreaker aims to generate transferable and imperceptible fake user profiles. Given that the review text is increasingly incorporated to improve the RSs [37] and informative textual reviews play an important role in explaining the reasons for the numerical ratings convincingly [21] and influencing users' decisions [40], the profiles are created with fake numerical ratings and enhanced with their corresponding fake textual reviews. To this end, we introduce a behavior pattern learning scheme to produce fake numerical ratings. For the generation of the fake textual reviews, we propose a demonstration retrieval algorithm and a text style transfer strategy to harness the in-context learning capabilities of the multimodal FMs.

*1) **Fake Numerical Rating Generation**:* The behavior pattern learning scheme is composed of three critical components to ensure the quality of the rating parts of the profiles, as shown in Fig. 1(b).

Personalized Template Sampling. Some works (e.g., [15]) create attack profiles from scratch (i.e., noise), resulting in low-quality profiles being produced. To address this, inspired by [6], [11], we use benign user profiles containing real rating patterns as templates. Unlike these previous methods that randomly sample from $\mathcal{M}$, we prioritize user profiles based on three standards: 1) preference is given to profiles that have interacted with items similar to the target item (e.g., within the same category), as they better reflect realistic behaviors given that benign users typically do not interact with items arbitrarily (e.g., guitarists tend not to purchase drum-related products [21]) and are more susceptible to influence due to the users share similar tastes; 2) profiles exhibiting a rich interaction history are favored to enhance personalization; and 3) profiles containing the target item are explicitly excluded to align with target normal users. As such, these standards collectively improve the effectiveness and imperceptibility of the attack. Let $\mathcal{P}$ to denote the profile templates,

$$\mathcal{P} = (\mathcal{E}\mathcal{M}_{r_{\widehat{\mathcal{U}}_t,:}})_{0:|\widetilde{\mathcal{U}}|,:}, \tag{2}$$

where the rating part of $\mathcal{M}$, represented as $\mathcal{M}_r$, $\mathcal{M}_r \in \mathbb{R}^{|\mathcal{U}|\times|\mathcal{V}|}$. $\widehat{\mathcal{U}}_t$ indicates the normal user set to align with 3),

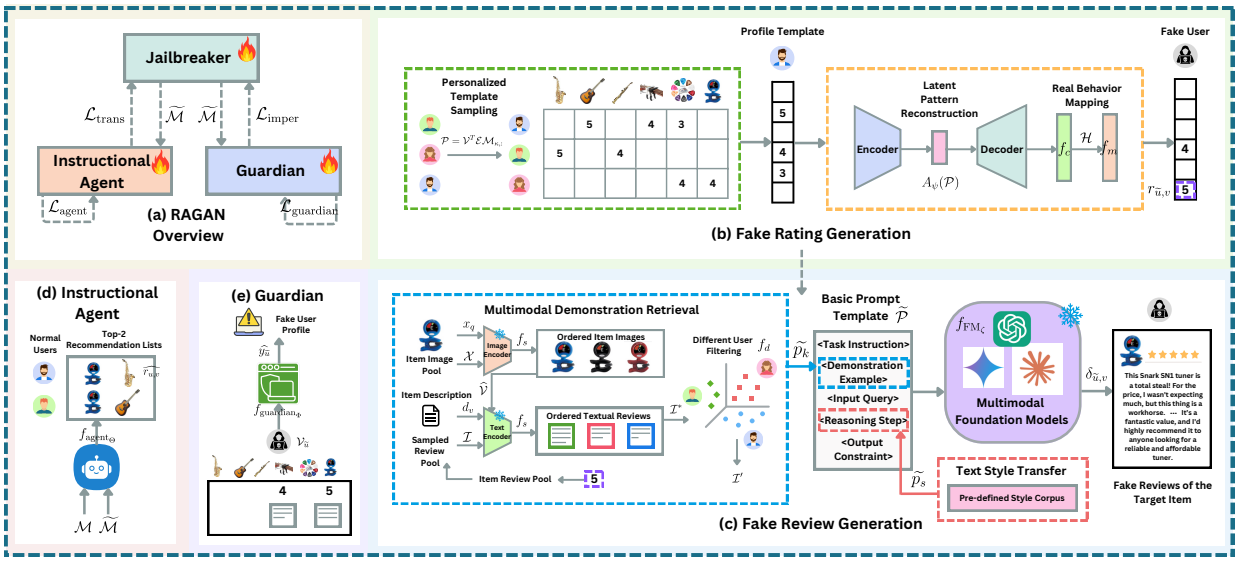


Fig. 1: The practical attack framework **RAGAN** consists of three modules: 1) a **Jailbreaker** that is responsible for generating fake user profiles containing both numerical ratings and textual reviews, where the quality of the profiles is strengthened through in-context learning enhanced with multimodal demonstration retrieval and text style transfer; 2) an **Instructional Agent** that is designed to improve attack transferability; and 3) a **Guardian** that is aimed at enhancing the imperceptibility of the profiles.

and $\mathcal{E}$ denotes a permutation matrix for sorting the matrix to match 1) and 2), where $\mathcal{E} \in \mathbb{R}^{|\widetilde{\mathcal{U}}_t| \times |\widetilde{\mathcal{U}}_t|}$.

Latent Pattern Reconstruction. To capture complex user-item associations from the templates in the latent space, RAGAN uses neural network paradigms with non-linearity [41]–[44] for the pattern reconstruction. Formally, the architecture of the paradigm is

$$A_\psi(\mathcal{P}) = f_l(\cdots f_2(f_1(\mathcal{P}))\cdots), \tag{3}$$

where $\mathcal{A}$ denotes the paradigm. There are various possible implementations for the paradigm. Here, we adopt AutoEncoder [23] (with parameters ψ) to sufficiently learn features at various levels of abstraction. AutoEncoder is built with an encoder and a decoder, where the encoder extracts the feature representations (i.e., user preferences) from the input and the decoder regenerates the rating vectors based on the representations. We use multi-layer perceptrons (MLPs) to build the function $f_j(\cdot)$, $j = 1, 2, ..., l$, where $f_j(x) = \text{Dropout}(\text{ReLU}(\text{BatchNorm}(W_j x + b_j)))$, and W_j and b_j are learnable parameters. For the encoder, we set the dimension of layers to half the dimension of previous layers; and for the decoder, which is a symmetric structure of an encoder, i.e., the size of layers is set to twice the size of the previous layers.

To align the output of the AutoEncoder with the real rating approach in RSs (e.g., Likert-scale [21] that is the most commonly used), RAGAN rescales the output within a certain range (e.g., [1, 5]), as formulated below.

$$f_c(\mathcal{A}_\psi(\mathcal{P})) = \frac{r_{u,v}^{\max} - r_{u,v}^{\min}}{1 + e^{-(W_c \mathcal{A}_\psi(\mathcal{P}) + b_c)}} + r_{u,v}^{\min} \tag{4}$$

where $f_c(\cdot)$ is the reconstruction function, $r_{u,v}^{\max}$ is the maximum rating of $\mathcal{M}$ (e.g., 5), $r_{u,v}^{\min}$ is the minimum rating of $\mathcal{M}$ (e.g., 1) and W_c as well as b_c are learnable parameters.

Real Behavior Mapping. As discussed in previous sections, RAGAN aims to maximize $h(t)$. Thus, for each attack profile, the rating of the target item is increased to the maximum rating of $\mathcal{M}$, i.e., $r_{u,v}^{\max}$, so as to promote the target item to as many normal users as possible to improve the attack effectiveness. For the filler items, an rounding-off operation $f_o(\cdot)$ is first used to discretize their ratings to match the input characteristics of the RSs. Given that the recommendation space in the real world is very large, each benign user generally interacts with only a few items in the space [21], such as a customer cannot purchase all the products on Amazon. Hence, to align with the real behaviors to avoid being discovery, RAGAN then masks most harmful and useless ratings of the generated profiles via a masking matrix $\widetilde{\mathcal{E}}$. Formally, the overall operation $f_m(\cdot)$ is

$$f_m(\mathcal{H}) = \widetilde{\mathcal{E}} \circ f_o(\mathcal{H}), \tag{5}$$

where $\mathcal{H} = f_c(\mathcal{A}_\psi(\mathcal{P})) \in \mathbb{R}^{|\widetilde{\mathcal{U}}| \times |\mathcal{V}|}$. $\widetilde{\mathcal{E}} \in \mathbb{R}^{|\widetilde{\mathcal{U}}| \times |\mathcal{V}|}$ performs hadamard product, where each entry is 1 or 0 representing the filler item to be retained or not. Let $\mathcal{P} = (p_{\widetilde{u},v})_{|\widetilde{\mathcal{U}}| \times |\mathcal{V}|}$, $\widetilde{\mathcal{E}} = (\widetilde{e}_{\widetilde{u},v})_{|\widetilde{\mathcal{U}}| \times |\mathcal{V}|}$, $\mathcal{H} = (h_{\widetilde{u},v})_{|\widetilde{\mathcal{U}}| \times |\mathcal{V}|}$, and $\mathcal{F}$ to denote the profile size. For each malicious user $\widetilde{u}$'s profile, when the absolute difference between $h_{\widetilde{u},v}$ and $p_{\widetilde{u},v}$ belongs to the first $\mathcal{F} - 1$ smallest differences and $p_{\widetilde{u},v} \neq 0$, $\widetilde{e}_{\widetilde{u},v} = 1$; otherwise, $\widetilde{e}_{\widetilde{u},v} = 0$. Thus, the behavior patterns of real users can be preserved to the maximum extent, which further improves diversity (e.g., picky users vs. nicey users) and thus the profiles' stealthiness.

*2) **Fake Textual Review Generation**:* Multimodal FMs have shown remarkable capabilities in language and vision understanding, reasoning and generation, revolutionizing a wide range of fields [45]. In-context learning empowers FMs to generalize to unseen downstream tasks purely through prompt-based conditioning, eliminating the need for parameter updates. Compared to fine-tuning and re-training approaches, which may be impractical due to restricted access to the model parameters (e.g., closed source models like GPT-4 and Claude 3) or constraints on computational resources required for large models such as LLaMA 3.3, such an effective and efficient paradigm gains increasing popularity [28]. As such, we carefully craft prompts to effectively harness the ICL capabilities

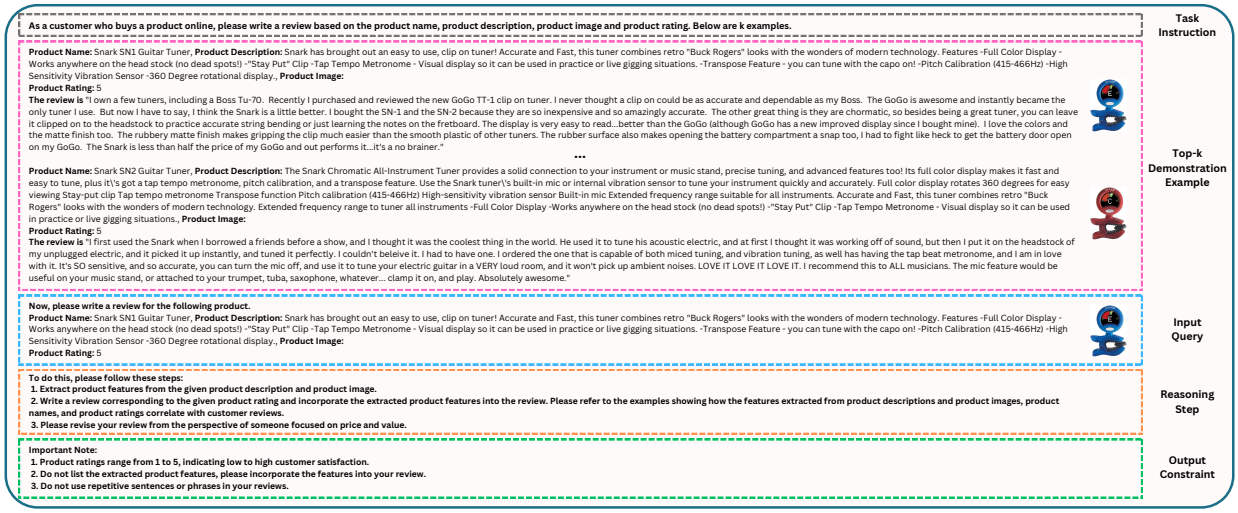


Fig. 2: An example of a multimodal prompt within the well-crafted prompt template that is fed into a foundation model for fake review generation, where top-k demonstration examples are obtained by our multimodal demonstration retrieval algorithm and our text style transfer strategy is integrated into the chain-of-thought reasoning process.

of the FMs to improve the quality of the corresponding fake textual reviews. Enhancing review quality can strengthen the optimization procedure of the attack effectiveness, resulting in more effective attack profiles.

Basic Prompt Template. We prepare the prompt template $\widetilde{\mathcal{P}}$ to provide instructions, goals, demos, queries, constraints and other contexts, so that the model knows 'what needs to be done' and 'how to respond to', as shown in Fig. 2. The main components are elaborated below.

1) *Task Instruction* $\widetilde{p}_t$. To make the generated reviews adapt to specific scenarios, we introduce role information [46] into context, allowing the FMs to respond with domain-specific knowledge. We then give detailed instructions, which guides the model to understand what tasks need to be done and generate desired responses.
2) *Demonstration Example* $\widetilde{p}_d$. Given that ICL often learns the pattern hidden in the demonstrations and accordingly makes predictions [28], the demonstrations play an important role in outputting expectantly. As such, we adopt few-shot ICL. As the figure shows, to sufficiently take advantage of the *internal knowledge* of the FMs, product names are introduced into the demos. To enrich the demos to improve the quality of the reviews, *external resources* across multiple modalities (e.g., product descriptions and images) are incorporated. For aligning the review with the given rating, product ratings is added.
3) *Input Query* $\widetilde{p}_i$. We present the task-specific queries corresponding to the items and ratings of the obtained fake user profiles. As shown in the example, we maintain the same product attributes and presentation order as demonstrations to fully leverage the language and vision understanding and reasoning capabilities of the FMs to improve the effectiveness of the generation.
4) *Reasoning Step* $\widetilde{p}_r$. To enrich user and item features in the generated reviews and enhance the alignment between the reviews and the ratings to ensure the quality, we adopt chain-of-thought (CoT) [47] to instruct the model how to respond to precisely.
5) *Output Constraint* $\widetilde{p}_o$. We give some constraints on the response to facilitate subsequent optimizations. Specifically, we define a rating range to further strengthen the alignment and impose additional format limits informed by empirical observations to reduce undesirable model behaviors that undermine review qualities.

Hence, the adversarial prompt template can be formulated:

$$\widetilde{\mathcal{P}} = \widetilde{p}_t \oplus \widetilde{p}_d \oplus \widetilde{p}_i \oplus \widetilde{p}_r \oplus \widetilde{p}_o, \tag{6}$$

where $\oplus$ denotes the integration of textual strings.

Based on our observations, when demonstrations are overly generic, the generated reviews tend to lack specificity in item characteristics and diversity in user behaviors, for instance, a review like "The Behringer A500 is a great amplifier for the price. It is powerful and has a lot of features.". In reality, users focus on different aspects of an item such as ease of use, affordability, flexibility, or scalability, based on their individual preferences. Moreover, users exhibit varied personalities (e.g., lenient or picky), thinking patterns, and cultural backgrounds, all of which shape their reviewing styles. The *richer the item features* described in textual reviews, the more users with similar tastes can be identified, thereby *increasing the promotional effect*. In addition, *the greater the variation in item features across different reviews of the target item* and *the more divergent the user features across profiles*, the *less detectable* the attack becomes [6]. Hence, the general reviews largely limit the effectiveness and stealthiness of the attack. To address the limitations, we propose a multimodal demonstration retrieval algorithm and a text style transfer strategy to enrich review quality and disguise.

Multimodal Demonstration Retrieval. We present an algorithm to retrieve review texts containing rich semantics and diverse behavior patterns from the learning corpus as top-k demonstrations. Given that the ICL performance is sensitive not only to the selection of demo samples, but also to the order of demonstrations [28], the algorithm prioritizes the reviews that are the most relevant to the query to enhance the generation quality of reviews.

For each rated item of each fake user $\widetilde{u}$'s profile (i.e., the target item and the filler items marked by Eq. (5)), k appropriate and related multimodal demonstrations are retrieved, as shown in Fig. 1(c). First, the algorithm obtains a set of items that are most similar to the rated item. Given that the visual appearances of items provide rich feature elements and crucial signals about user preferences [48], we propose to use the item images to calculate the similarity. For each item image $x_v, x_v \in \mathcal{D}$, the similarity score between it and the query image x_q (i.e., the rated item's image) is obtained by a function $f_s(\cdot)$,

$$s_v = f_s(x_q, x_v), \tag{7}$$

where s_v represents the similarity score and $f_s(\cdot)$ is implemented via lightweight cosine similarity, i.e., $f_s(x_q, x_v) = \frac{f_e(x_q)\cdot f_e(x_v)}{||f_e(x_q)||||f_e(x_v)||}$. $f_e(\cdot)$ refers to a pre-trained vision encoder for feature vector extraction. We use CLIP-ViT[1] to build it here. As a result, an ordered list of items $\widehat{\mathcal{V}}$ can be produced using the sequence of the similarity scores $\{s_v\}$. Formally, $\widehat{\mathcal{V}} = \text{argsort}_v(\{s_v\})$, where the higher the score, the higher the ranking of the item. Note that the first one of $\widehat{\mathcal{V}}$ is the rated item itself.

To ensure the alignment between the numerical ratings and the textual reviews, the algorithm then selects the reviews from the matrix $\mathcal{M}$ with the same rating of the attack item (i.e., $r_{\widetilde{u},v}$). As such, such demos have high similarity to the input query $\widetilde{p}_i$. Formally, $\mathcal{I} = \{\delta_{u,v} : r_{u,v} = r_{\widetilde{u},v}, r_{u,v} \in \mathcal{M}\}$, where $\mathcal{I}$ is the set of the filtered reviews.

Afterwards, the algorithm sorts the selected reviews in $\mathcal{I}$ based on the obtained similar items and the semantic richness. The algorithm measures the similarity between the reviews and the textual item descriptions $d_v, d_v \in \mathcal{D}$ (including material, volume, shape, weight, colour, functionality, usage scenarios and so on) to get the semantic richness scores. For each user $u, u \in \mathcal{U}$'s textual review $\delta_{u,v}$, $\delta_{u,v} \in \mathcal{I}$, the richness score $c_{u,v}$ is computed through

$$c_{u,v} = f_s(d_v, \delta_{u,v}), \tag{8}$$

where $f_s(\cdot)$ is implemented via the cosine similarity. Thereinto, $f_e(\cdot)$ is constructed with a pre-trained textual encoder, sentence transformer (a.k.a. SBERT[2]). If d_v is empty, the algorithm uses the lexical diversity to indicate the richness, that is, the number of different words in the textual review. The algorithm thus ranks the reviews for each $v, v \in \widehat{\mathcal{V}}$ based on $c_{u,v}$. Let $\mathcal{I}_v$ denote the sequence of v's ordered review samples and $\mathcal{I}^*$ represent the list of all the reviews,

$$\begin{aligned} \mathcal{I}_v &= \{\delta_{u,v}\} \\ \text{subject to } u &= \underset{u}{\text{argsort}}(\{c_{u,v}\}), \end{aligned} \tag{9}$$

where the higher the score, the richer the item features of the review text. Hence, $\mathcal{I}^* = \text{concatenation}(\{\mathcal{I}_v\})$.

The algorithm finally filters out the reviews from the same users and then obtains top-k ordered reviews. Thus, a new review list $\mathcal{I}'$ for demo examples is created,

$$\mathcal{I}' = f_d(\mathcal{I}^*)_{:k}, \tag{10}$$

[1]https://openai.com/index/clip/
[2]https://sbert.net/

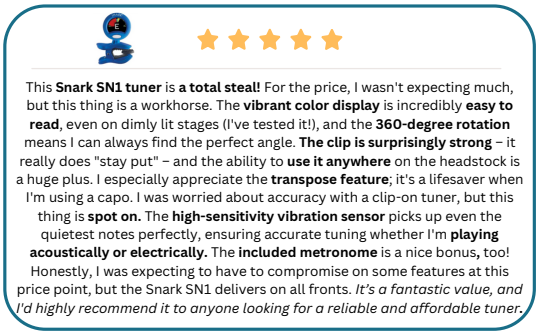


Fig. 3: An example of reviews for a target item in the fake user profiles generated by RAGAN.

where $f_d(\cdot)$ denotes the function to capture the reviews from different users for demonstrating diverse user features.

Let $\widetilde{p}_k$ denote the set of the multimodal demos. Based on our designed template, $\widetilde{p}_k = \{(x_v, n_v, d_v, r_{u,v}, \delta_{u,v}) : \delta_{u,v} \in \mathcal{I}', r_{u,v} \in \mathcal{M}, x_v, n_v, d_v \in \mathcal{D}\}$, which serves as in-context demo examples for FMs, aiming to enhance their understanding and performance on analogous tasks (i.e., review generation here).

Text Style Transfer. Text style transfer is the task of rewriting text to incorporate additional stylistic elements while preserving the overall semantics and structure [49]. To further enrich the user features, we adopt a strategy to transfer the style of the reviews. The strategy first establishes a comprehensive text style transfer prompt corpus encompassing ten distinct categories, as elaborated in the appendix. Then, the strategy randomly selects a candidate from the corpus as $\widetilde{p}_s$, such as 'Please revise your review from the perspective of someone focused on price and value.'. The strategy finally incorporates $\widetilde{p}_s$ into the reasoning steps of the prompt template $\widetilde{\mathcal{P}}$, as shown in Fig. 2.

Consequently, for each rating $r_{\widetilde{u},v}$ obtained from $f_m(\mathcal{H})$, the corresponding review is obtained via

$$\delta_{\widetilde{u},v} = f_{\text{FM}_\zeta}(\widetilde{\mathcal{P}} \oplus \widetilde{p}_k \oplus \widetilde{p}_s), \tag{11}$$

where FM is with frozen parameters ζ. As can be seen from the example in Fig. 3, the generated textual reviews contain rich item characteristics (bond fonts, e.g., excellent design and versatile functionality) and diverse user behaviors (italic fonts, e.g., cost-effective).

B. Instructional Agent

It is not easy to access the accurate knowledge of the victim RSs and their detailed recommendation results in real-world scenarios [21]. As such, we introduce an instructional agent to measure how effective the attack is and then optimize the Jailbreaker to improve the attack transferability, as inspired by [6], [21]. Existing works [3], [6], [13], [18] often use rating-only-based RSs to evaluate the effectiveness, which is unsuitable for our profiles containing both numerical ratings and textual reviews. We thus adopt the review-based RS [25] to build the agent, as shown in Fig. 1(d), where the semantically rich textual reviews are introduced to enrich user and item features for accurately predicting user preferences, so that the valuable and effective instructions can be provided to the Jailbreaker within the limited accessible data.

Algorithm 1 RAGAN Optimization Procedure

Input: p percent of user-item interaction matrix $\mathcal{M}$, and item metadata $\mathcal{D}$
Output: fake matrix $\widetilde{\mathcal{M}}$

for number of training epochs **do**
  Solve for the Lower-level Problems:
  Obtain $\widetilde{\mathcal{M}}$ with Eq. (2) - Eq. (11)
  for number of training epochs **do**
    Optimize the Instructional Agent Θ on $p\mathcal{M} + \widetilde{\mathcal{M}}$ to minimize $\mathcal{L}_{\text{agent}}$
  end for
  Optimize the Guardian Φ on $p\mathcal{M} + \widetilde{\mathcal{M}}$ to maximize $\mathcal{L}_{\text{guardian}}$
  Solve for the Upper-level Problem:
  Calculate $\mathcal{L}_{\text{trans}}$ on Θ with Eq. (12) and Eq. (14)
  Calculate $\mathcal{L}_{\text{imper}}$ on Φ with Eq. (13) and Eq. (15)
  Optimize the Jailbreaker to minimize Eq. (1)
end for
Obtain optimal $\widetilde{\mathcal{M}}$ from the optimal Jailbreaker
return $\widetilde{\mathcal{M}}$

The agent performs a function $f_{\text{agent}_\Theta}(\cdot)$ with the trainable parameters Θ to predict the preference of the user u on v, which is denoted as $\widehat{r_{u,v}}$ and can be formulated below.

$$\widehat{r_{u,v}} = f_{\text{agent}_\Theta}(\delta_u, \delta_v, q_u, q_v), \tag{12}$$

where δ_u and δ_v are, respectively, the set of historical reviews from u and on v, and q_u and q_v indicate the user ID embeddings and the item ID embeddings, separately. $f_{\text{agent}_\Theta}(\cdot)$ is generally constructed with feature modeling and preference prediction components [25]. For example, in the architecture of the previous work DeepCoNN++ [21], a TextCNN [37] is first used to extract features from the user and the item reviews. The features are then combined with the embeddings. Afterwards, the latent representations of the user network and the item network are concatenated and mapped to a shared feature space. Finally, a MLP followed by a Sigmoid is adopted for effectively conducting implicit recommendations.

C. Guardian

The guardian with the learnable parameters Φ plays a minimax game with the jailbreaker empowered by the agent, as inspired by the vanilla GAN [24]. Therefore, the module distinguishes the fake user profiles from the benign user profiles as much as possible to enhance the imperceptibility of the attack, as shown in Fig. 1(e). Let $f_{\text{guardian}_\Phi}(\cdot)$ denote the function to identify malicious/benign behavior patterns from fake/benign user $\widetilde{u}/u$. In the case of the fake user profile $\mathcal{V}_{\widetilde{u}}$,

$$\widehat{y_{\widetilde{u}}} = f_{\text{guardian}_\Phi}(\mathcal{V}_{\widetilde{u}}), \tag{13}$$

where $\widehat{y_{\widetilde{u}}}$ denotes the prediction of $\widetilde{u}$'s profile. There are two kinds of predicted results: one is normal and another is abnormal. Based on the previous detector [21], $f_{\text{guardian}_\Phi}(\cdot)$ can be built through a TextCNN, an Encoder and a MLP, where the TextCNN enhanced by the Encoder are used to capture features from the profiles, and the MLP followed by a Sigmoid is leveraged to perform detections based on the obtained feature representations. The benign user profile's prediction $\widehat{y_u}$ can be produced via the similar operations.

TABLE II: Statistics of Datasets

Dataset	#Users	#Items	#Reviews	Sparsity
Amazon Musical Instruments	1,429	900	10,261	99.20%
Amazon Automotive	2,928	1,835	20,473	99.62%
Yelp	1,599	1,318	30,120	98.57%

D. Learning

As discussed in Section III-C, we obtain high-quality attack profiles by solving a bi-level optimization problem.

The Lower-level Problems. We adopt binary cross-entropy (BCE) [35] for $\mathcal{L}_{\text{agent}}$ to adapt to the implicit top-$\mathcal{K}$ recommendation task. Θ will be obtained when $\mathcal{L}_{\text{agent}}$ is minimum that is closest to the actual situation. Similarly, given that the guardian performs a binary classification task, inspired by [6], we adopt negative BCE for $\mathcal{L}_{\text{guardian}}$. Since this module plays the minimax game with the jailbreaker, Φ will be obtained when $\mathcal{L}_{\text{guardian}}$ is maximum, ensuring that the guardian achieves its best ability to distinguish between benign and attack profiles.

The Upper-level Problem. $\mathcal{L}_{\text{trans}}$ aims to maximize $h(t)$. If t is in the recommendation lists of the normal users, it is not necessary to optimize much. But if not, it is to minimize the prediction rating gap between t and the predicted items that are in the recommended lists, so that the target item can be promoted to as many normal users as possible. We design the loss as follows.

$$\mathcal{L}_{\text{trans}} = \log\Big(\sum_{u \in \widehat{\mathcal{U}}_t} \sum_{v \in \widehat{\mathcal{V}}_{u,:k}} (e^{\widehat{r_{u,v}}} - e^{\widehat{r_{u,t}}}) + 1\Big), \tag{14}$$

where $\widehat{r_{u,t}}$ is the prediction of the target normal user u on the target item t from $f_{\text{agent}_\Theta}(\cdot)$ and $e^{(\cdot)}$ is used to amplify the rating gap and $\log(\cdot)$ is used to shrink the overall sum of the gap to a range to avoid this attack objective overly dominating the optimization direction. To make sure the loss is positive, we add 1 to the sum. $\mathcal{L}_{\text{imper}}$ is optimized by tricking the guardian that the fake profiles are from real users, formally,

$$\mathcal{L}_{\text{imper}} = \frac{1}{\widetilde{\mathcal{U}}} \sum_{\widetilde{u} \in \widetilde{\mathcal{U}}} \log(1 - \widehat{y_{\widetilde{u}}}). \tag{15}$$

$\mathcal{L}_{\text{imper}}$ is weighted by λ and combined with $\mathcal{L}_{\text{trans}}$ to form the final loss for optimizing the parameters of the generation module, as detailed in Section III-C. Inspired by the works [6], [21], the optimization procedure of RAGAN is detailed in Algorithm 1, where Adam is used as an optimizer.

V. Experiment

A. Experiment Setup

1) Datasets Selection: We use three real-world datasets from different scenarios [46], [50] to evaluate RAGAN, as detailed in Table II. For Yelp[3], we randomly select a subset to avoid exceeding the hardware limit. The datasets vary in size and sparsity, which is suitable for a comprehensive evaluation of RAGAN. We adopt the leave-one-out method [35] to select the training set and the test set, with a split ratio of 9:1.

[3]https://www.yelp.com/dataset

TABLE III: HR@10 and NDCG@10 of different attacks against various victim RSs on real-world datasets. We use bold fonts and asterisk symbols to denote the best performance and second best performance methods, respectively.

Victim RS	Dataset	Metric	Attack Method										
			Random	Bandwagon	PGA	DCGAN	AUSH	DLA	RecUP	Leg-UP	TrialAttack	R-Trojan	RAGAN
WRMF	Musical	HR	0.2649	0.2968	0.3904	0.3599	0.3781	0.4383	0.4020	0.4681	0.4804	0.5530*	**0.6313**
		NDCG	0.1082	0.1211	0.1822	0.1957	0.1845	0.2208	0.1970	0.2374	0.2418	0.3617*	**0.4254**
	Automotive	HR	0.1137	0.1348	0.1563	0.1695	0.1698	0.2094	0.1730	0.2305	0.2284	0.2693*	**0.3185**
		NDCG	0.0451	0.0533	0.0772	0.0818	0.0796	0.0981	0.0781	0.1232	0.1145	0.1677*	**0.2262**
	Yelp	HR	0.0939	0.0861	0.1082	0.1017	0.1231	0.1276	0.1399	0.1360	0.1464	0.1852*	**0.2494**
		NDCG	0.0635	0.0563	0.0801	0.0731	0.0799	0.0893	0.0969	0.1087	0.0998	0.1496*	**0.2011**
NCF	Musical	HR	0.1495	0.1509	0.1633	0.1807	0.2068	0.2525	0.2104	0.2438	0.2714	0.3382*	**0.4209**
		NDCG	0.0633	0.0575	0.0677	0.0816	0.0944	0.1305	0.0802	0.1112	0.1503	0.2009*	**0.2526**
	Automotive	HR	0.1043	0.1102	0.1192	0.1383	0.1650	0.1962	0.1827	0.2038	0.2187	0.2679*	**0.3095**
		NDCG	0.0474	0.0515	0.0604	0.0854	0.0971	0.1009	0.0935	0.1109	0.1353	0.1594*	**0.1925**
	Yelp	HR	0.0784	0.0667	0.0848	0.0965	0.0991	0.1159	0.1082	0.1328	0.1250	0.1729*	**0.2040**
		NDCG	0.0472	0.0337	0.0561	0.0603	0.0553	0.0752	0.0598	0.0844	0.0851	0.1162*	**0.1469**
LightGCN	Musical	HR	0.0602	0.0718	0.0893	0.0972	0.1110	0.1509	0.1023	0.1437	0.1357	0.1858*	**0.2569**
		NDCG	0.0260	0.0370	0.0413	0.0466	0.0556	0.0786	0.0517	0.0698	0.0732	0.0983*	**0.1331**
	Automotive	HR	0.0340	0.0343	0.0589	0.0433	0.0634	0.1102	0.0735	0.1088	0.1009	0.1536*	**0.2149**
		NDCG	0.0189	0.0174	0.0359	0.0238	0.0324	0.0599	0.0387	0.0632	0.0681	0.0846*	**0.1363**
	Yelp	HR	0.0136	0.0117	0.0246	0.0317	0.0415	0.0602	0.0434	0.0544	0.0512	0.0926*	**0.1308**
		NDCG	0.0059	0.0061	0.0141	0.0197	0.0265	0.0262	0.0252	0.0354	0.0359	0.0560*	**0.0961**

2) Baseline Attack Methods: We compare RAGAN with typical and state-of-the-art poisoning attacks including Random, Bandwagon [4], PGA [7], DCGAN [10], AUSH [11], DLA [16], RecUP [15], Leg-UP [6], TrialAttack [18] and R-Trojan [21]. The hyper-parameters of the baselines are set as suggested in the original papers.

3) Targeted Recommender Systems: We consider three rating-only-based victim RSs, as existing works do [6], [11], [21]: WRMF [51], NCF [35] and LightGCN [36]. Moreover, to evaluate the attack effectiveness on review-based RSs under both black-box and white-box settings, we use DeepCoNN [26] and our surrogate model as victim RSs. The hyper-parameters of the victim RSs are set as suggested in the original papers.

4) Evaluation Metrics: Following the existing works [6], [11], [15], [16], [18], we use two widely-used ranking metrics to evaluate attack transferability: hit ratio (HR@k $\uparrow$) and normalized discounted cumulative gain (NDCG@k $\uparrow$), where HR@k measures whether the target item is appeared in the top-k recommendation list, while NDCG@k indicates the ranking position of the target item in the list [52]. k is set to 10 here. We also introduce the standard sentence perplexity score $\downarrow$ [53], a commonly-adopted metric for evaluating attack imperceptibility, as textual reviews primarily serve to enhance the stealthiness of the generation profiles [21].

5) Configuration of RAGAN: To maintain a fair comparison, we set the attack size to 3% of the population as default, which can clearly show the performance differences [6], and the profile size equals the average number of ratings per user in the data set. For RAGAN, we set l=6, the number of in-context examples to 3, training epochs to 20, batch size to 256, learning rate to 0.001, λ to 0.5 to balance the two attack goals. To improve the attack efficiency while reducing the attack costs, we adopt 'gemini-1.5-flash' for $f_{\text{FM}_\zeta}(\cdot)$, 'vit_large_patch14_224_clip_laion2b' for CLIP-ViT and 'all-MiniLM-L6-v2' for SBERT, where the hyper-parameters are configured as the default values provided by Hugging Face[4]. For other hyper-parameters and the data pre-processing scheme, we follow the previous work R-Trojan [21] to ensure the effectiveness of the evaluation.

[4]https://github.com/huggingface/transformers

B. Attack Transferability

1) Overall Transferability: Table III and Table IV illustrate the impressive transferability of RAGAN, underscoring the critical role of high-quality textual reviews in strengthening the effectiveness of the attack profiles across diverse victim RSs. From Table III, we can find that LightGCN is more robust than NCF and WRMF, possibly due to its unique graph convolutions. For the review-based RSs, as shown in Table IV, DeepCoNN++ is more vulnerable than DeepCoNN, due to its internal knowledge can be accurately obtained by RAGAN. Furthermore, the results on Yelp exhibit less variability, which we attribute to its larger scale and higher density, making it more resistant to the perturbations.

2) RAGAN vs. Attack Baselines: As shown in Table III and Table IV, RAGAN achieves the state-of-the-art performance among all evaluated methods against various victim RSs on real-world datasets, with the highest HR@10 and NDCG@10. By refining the quality of textual reviews, RAGAN demonstrates improved attack transferability compared to R-Trojan. R-Trojan greatly outperforms other baselines, with its key distinction being the incorporation of textual reviews, demonstrating that even basic utilization of the reviews can significantly improve the attack performance.

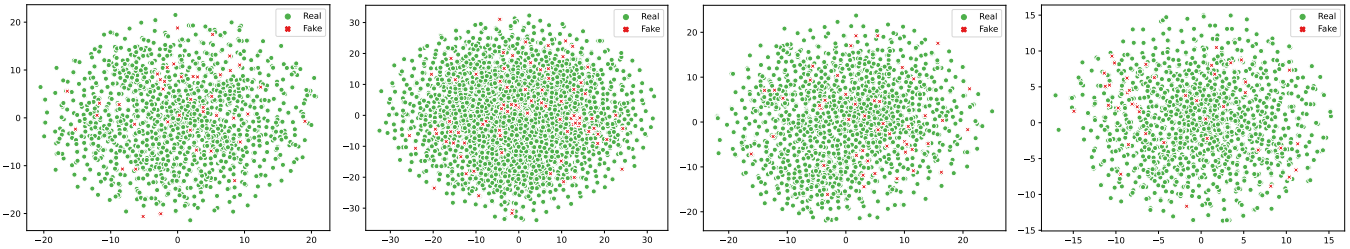
Among the remaining baselines, Leg-UP, TrialAttack and DLA exhibit comparable performance, likely due to their use of surrogate RSs (Leg-UP and TrialAttack), which enhance attack transferability, and DLA's optimization for DL-based RSs. RecUP and AUSH achieve moderate performance, outperforming DCGAN due to their specialized GAN architectures for shilling attacks. In contrast, PGA, which is tailored for MF models, fails to effectively transfer to DL-based RSs such as NCF and LightGCN. Finally, conventional attack methods (i.e., Random and Bandwagon) demonstrate the lowest transferability among all baselines.

C. Attack Imperceptibility

1) Overall Imperceptibility: Since we are the first to incorporate textual reviews into fake user profiles to enhance

TABLE IV: HR@10 and NDCG@10 of R-Trojan and RAGAN with different attack sizes against various review-based RSs on three real-world datasets, where the attacks are performed under the black-box and white-box settings, respectively.

Dataset	Attack	DeepCoNN						DeepCoNN++					
		0.5%		1%		3%		0.5%		1%		3%	
		HR	NDCG	HR	NDCG	HR	NDCG	HR	NDCG	HR	NDCG	HR	NDCG
Musical	None	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	R-Trojan	0.1219	0.0356	0.1916	0.0560	0.3991	0.1170	0.1938	0.0583	0.2663	0.0783	0.5835	0.2132
	RAGAN	**0.3316**	**0.0972**	**0.4049**	**0.1191**	**0.5624**	**0.2018**	**0.3752**	**0.1114**	**0.4521**	**0.1353**	**0.7438**	**0.2334**
Automotive	None	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	R-Trojan	0.0810	0.0287	0.1230	0.0420	0.2763	0.0939	0.1220	0.0386	0.2026	0.0823	0.4523	0.1576
	RAGAN	**0.1783**	**0.0522**	**0.2192**	**0.0638**	**0.4621**	**0.1963**	**0.2293**	**0.0704**	**0.3330**	**0.1275**	**0.6312**	**0.2245**
Yelp	None	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	R-Trojan	0.0547	0.0160	0.0861	0.0252	0.2003	0.0586	0.0835	0.0252	0.1574	0.0471	0.3297	0.1050
	RAGAN	**0.0835**	**0.0245**	**0.1159**	**0.0341**	**0.2597**	**0.0760**	**0.1645**	**0.0504**	**0.2545**	**0.0788**	**0.4495**	**0.1394**



(a) RAGAN on Musical (b) RAGAN on Automotive (c) RAGAN on Yelp (d) RAGAN on Partial Musical

Fig. 4: Visualization of RAGAN's fake user profiles and real user profiles on real-world datasets. The attack profiles overlap with normal profiles, reflecting a global semantic alignment and demonstrating the imperceptibility of RAGAN.

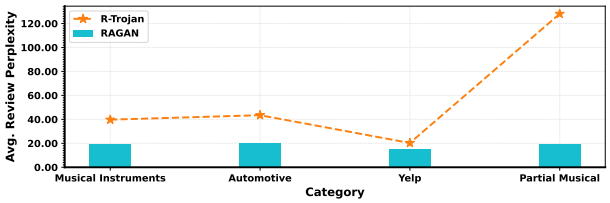


Fig. 5: Average text perplexity of generated reviews from different attacks with varying knowledge on real-world datasets.

attack effectiveness, existing detectors, which primarily focus on rating-only profiles (e.g., [6], [19]), are inadequate for identifying RAGAN. To demonstrate the imperceptibility of the attack, we employ t-SNE [6] to visualize the real user's and RAGAN's profile representations generated by the guardian module on real-world datasets, which involves rating and review information. As shown in Fig. 4, the representations of fake user profiles are scattered within the distribution of real user profiles on real-world datasets. The overall proximity between attack profiles and normal profiles in the embedding space can be attributed to the imperceptibility optimizations, which enforces high semantic similarity. Meanwhile, the dispersed nature of the attack profiles may stem from our style transfer strategy, which enhances the diversity and realism of the generated behaviors. Hence, RAGAN can effectively mimic real users, enabling highly inconspicuous attacks.

2) Review Quality: To further assess the imperceptibility of the attack, we conduct an additional evaluation on the generation quality of the adversarial reviews. Following the works [46], [53], we adopt the text perplexity score computed by GPT-Neo as the metric. A lower perplexity indicates that the perturbed text remains closer to natural human language, thus appearing less suspicious and detectable. As shown in Fig. 5, RAGAN achieves a lower average perplexity than R-Trojan, demonstrating its superiority in generating high-quality review text. These reviews are not only fluent and coherent but also semantically meaningful and stealthy, thereby enhancing their deceptive effectiveness. The performance gap may be attributed to the stronger ICL and generative abilities of RAGAN's underlying multimodal FMs powered by the retrieval demonstration algorithm and the text style transfer strategy, compared to R-Trojan's plain GPT-2.

D. Ablation Studies

We remove some pivotal components of RAGAN and investigate the performance changes. Table V illustrates the performance of ablation studies of RAGAN. We give results on two representative datasets, which are Amazon Musical Instruments and Yelp, due to space limitations.

To validate the effectiveness of our review quality enhancement component, we conduct ablation experiments by removing the multimodal demonstration retrieval algorithm. Since our previous work [21] has empirically demonstrated the effectiveness of textual reviews for strengthening fake user profiles, we focus solely on assessing how review quality affects attack performance. By comparing the cases of $\widetilde{p}_k = \times$ and $\widetilde{p}_k = \checkmark$ (i.e., zero-shot vs. few-shot learning), we observe that improving the quality of fake textual reviews contributes notably to the transferability of the attack. The main reason may be that the reviews encapsulate rich item features, which can help focus on a broader set of users with similar taste preferences and thus enhance item exposure rates. Moreover, as shown in Table III, Table IV and Table V, RAGAN without $\widetilde{p}_k$ achieves comparable performance to R-Trojan across various victim RSs. This is primarily attributed to the use of powerful foundation models with extensive corpus, which possess strong capabilities in understanding, reasoning and generation, thereby producing more semantically meaningful

TABLE V: Ablation and parameter studies of RAGAN on real-world datasets.

Dataset	Component			WRMF		NCF		LightGCN		DeepCoNN		DeepCoNN++	
	λ	p	$\widetilde{p}_k$	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10
Amazon	0.5	1	×	0.5581	0.3137	0.3527	0.2078	0.1967	0.1079	0.4064	0.1191	0.6081	0.1901
	0	1	✓	0.4673	0.2619	0.2605	0.1316	0.1407	0.0741	0.3019	0.0882	0.4369	0.1316
	0.5	0.5	✓	0.5784	0.3650	0.3904	0.2271	0.2199	0.1160	0.5210	0.1772	0.7163	0.2291
	0.5	1	✓	**0.6313**	**0.4254**	**0.4209**	**0.2526**	**0.2569**	**0.1331**	**0.5624**	**0.2018**	**0.7438**	**0.2334**
Yelp	0.5	1	×	0.1859	0.1340	0.1593	0.0750	0.1023	0.0650	0.2267	0.0666	0.3303	0.1099
	0	1	✓	0.1276	0.0938	0.1250	0.0659	0.0654	0.0459	0.0848	0.0248	0.1587	0.0468
	0.5	0.5	✓	0.2383	0.1523	0.1865	0.1362	0.1159	0.0836	0.2383	0.0701	0.4268	0.1431
	0.5	1	✓	**0.2494**	**0.2011**	**0.2040**	**0.1469**	**0.1308**	**0.0961**	**0.2597**	**0.0760**	**0.4495**	**0.1394**

TABLE VI: Examples of fake ratings with the corresponding fake textual reviews from different attacks on real-world datasets.

Dataset	Attack	Fake Review Text Generated by Attack Methods	Fake Rating
Amazon	R-Trojan	**Excellent Guitar Stand Black** I've had several stands for my electric guitars and none of them had a black coating on them. The only thing I wanted was black. A non-skid coat like Fender, suede or something would have been perfect. I'm glad I did.	5.0
	RAGAN	**This Musician's Gear Tubular Guitar Stand** is a lifesaver! At my age, getting my guitar in and out of its case can be a bit of a struggle. This stand is a perfect solution. It's so easy to use; just set the guitar in place—the soft rubber tubing is gentle on the instrument, and the neck rest keeps it secure. The stand is very stable, and the sturdy, wide base gives me peace of mind that my guitar won't tip over. I also appreciate how easily it folds up for storage – it's wonderfully compact when not in use. It's simple to set up and put away, no complicated assembly required. For the price, it's a fantastic value. I highly recommend it to anyone, especially those of us who value ease of use and a secure place to keep their prized instrument.	5.0
Yelp	R-Trojan	**Average RA Sushi Bar Restaurant** My SO and I visited RA Sushi for happy hour (Saturday) around 6:30. The happy hour prices were very reasonable, including 1/2 price beer, and the sushi was pretty great. It was the best sushi we've found in the area. We ordered the tuna poke, the Hawaiian, and the Godzilla. The Hawaiian was amazing - I could have easily made this my meal if I didn't have to. The Hawaii was good, but was also a lot better - much more flavorful and fresh. The sushi was pretty fresh and it looked like there was really really nothing wrong with it. It was only $5 cheaper, but I don't know if we would have gone there for dinner. I would definitely go back.	3.0
	RAGAN	My recent visit to **RA Sushi** was a bit of a mixed bag. The restaurant itself is pleasant enough, easy to navigate, and the seating was comfortable. I appreciated the large print on the menu, making it simple to read. My order, a simple California roll and miso soup, was accurately taken. The soup was delicious, perfectly warm and flavorful. However, the sushi rice was a little bit on the dry side for my taste. The service was friendly enough, but a little slow; we had to ask for our check twice. Overall, it was a decent experience, but nothing exceptional. The food was good, but not outstanding, and the service was adequate but not overly attentive. For a casual meal, it's acceptable, but I wouldn't rush back, especially if you're looking for particularly speedy service. The bill was reasonable.	3.0

review text than the GPT-2 backbone used in R-Trojan. Moreover, as shown in Table V, removing the instructional agent (i.e., $\lambda = 0$) results in a significant drop in attack performance against different victim models, indicating its vital role in optimizing and enhancing the attack transferability.

E. Attack Performance with Partial Knowledge

To evaluate the effectiveness of RAGAN in more practical settings, we conduct experiments with the assumption that the attacker can access only 50% of the user-item interaction records, i.e., $p = 0.5$. Due to space limitations, we focus on representative datasets, reporting mainly on Amazon Musical Instruments and additionally on Yelp for a more comprehensive evaluation. For the other datasets, the trends are similar. From Fig. 4(d) and Table V, we can observe that RAGAN is still effective against various black-box RSs on real-world datasets, and its profiles remain similar to normal user profiles, with high-quality reviews generated. These results indicate that RAGAN can generate transferable and imperceptible fake user profiles, even when only partial knowledge of the training data of the victim RSs is accessed. This weak model in Table V is significantly superior to all the baselines learned with the full data in Table III and Table IV, and the same advantage is also shown in Fig. 5, demonstrating the effectiveness of the semantically rich textual reviews in reinforcing the attack. These findings highlight that existing RSs are vulnerable to semantically informed threat models in realistic scenarios.

F. Examples of Adversarial Attack Profiles

We provide examples to demonstrate the quality of the attack profiles generated by R-Trojan and RAGAN (note that R-Trojan is the first and only method to incorporate reviews into the profiles in the literature). Due to space constraints, we randomly select one item from the profiles on each type of dataset. As illustrated in Table VI, the reviews in R-Trojan are incoherent, factually inaccurate, semantically confused, logically inconsistent, sparse in content and misaligned with the ratings. In contrast, the reviews in RAGAN are more semantically meaningful, richer in valuable features, more coherent and natural, and better aligned with the ratings, resulting in higher-quality profiles. This observation also sheds light on why RAGAN is effective and why its profiles are both transferable and imperceptible.

VI. Conclusion

Many existing attack methods are overly idealized and fail to reflect realistic adversarial scenarios, which can result in overly optimistic assessments of the robustness of RSs in practice. To mitigate this problem, in this paper, we propose a novel poisoning attack framework, RAGAN, for generating high-quality fake user profiles including carefully designed ratings and textual reviews. RAGAN achieves significant improvements over existing profiles, particularly in the coherence and semantic richness of the reviews, as well as in the alignment between ratings and reviews. As a result, RAGAN demonstrates state-of-the-art transferability and imperceptibility in a more practical setting, as shown in the comprehensive experiments on real-world datasets. RAGAN enables us to delve into the vulnerabilities of current RSs, revealing their inherent security risks under specific input features (e.g., well-crafted ratings and/or reviews), and guides us to secure them in actual deployment scenarios.

Appendix

Text Style Transfer Prompt Corpus

To support nuanced and controllable text style transfer, we construct a comprehensive prompt corpus organized into ten major categories: *sentiment, formality, persona, emotion, complexity, audience, creativity, comparison, genre, and dialect*. Each category captures a distinct dimension of stylistic variation. For instance, sentiment-based prompts guide

the tone of opinion expression (e.g., positive, negative, or neutral), while formality-based prompts regulate the register from casual to highly formal. Persona-based prompts simulate different user perspectives such as experts, first-time buyers, parents, or teenagers. Emotion-based prompts capture a wide range of affective tones including joy, disappointment, anger, fear, and gratitude. Complexity-based prompts control linguistic sophistication, ranging from plain, accessible language to technical or complex styles. Audience-specific prompts adapt the review for particular demographic groups like families, professionals, students, or seniors. Creativity-focused prompts encourage stylistic diversity through humor, satire, storytelling, or poetic forms. Comparative prompts introduce contrastive framing, such as before-and-after contrasts or upgrades over time. Genre-based prompts reframe content into different communicative styles, including advertisement, news report, dialogue, or social media post. Finally, dialect prompts adjust for linguistic variation across English dialects (e.g., British vs. American) and regional colloquialisms to reflect diverse cultural norms. Hence, this taxonomy enables fine-grained control over stylistic modifications.

REFERENCES

[1] R. Cohen, O. Sar Shalom, D. Jannach, and A. Amir, "A black-box attack model for visually-aware recommender systems," in *WSDM*, 2021.

[2] J. Chen, W. Fan, G. Zhu, X. Zhao, C. Yuan, Q. Li, and Y. Huang, "Knowledge-enhanced black-box attacks for recommendations," in *KDD*, 2022.

[3] H. Zhang, C. Tian, Y. Li, L. Su, N. Yang, W. X. Zhao, and J. Gao, "Data poisoning attack against recommender system using incomplete and perturbed data," in *KDD*, 2021.

[4] I. Gunes, C. Kaleli, A. Bilge, and H. Polat, "Shilling attacks against recommender systems: a comprehensive survey," *Artificial Intelligence Review*, 2014.

[5] M. Si and Q. Li, "Shilling attacks against collaborative recommender systems: a review," *Artificial Intelligence Review*, 2020.

[6] C. Lin, S. Chen, M. Zeng, S. Zhang, M. Gao, and H. Li, "Shilling black-box recommender systems by learning to generate fake user profiles," *TNNLS*, 2022.

[7] B. Li, Y. Wang, A. Singh, and Y. Vorobeychik, "Data poisoning attacks on factorization-based collaborative filtering," *NIPS*, 2016.

[8] G. Yang, N. Z. Gong, and Y. Cai, "Fake co-visitation injection attacks to recommender systems." in *NDSS*, 2017.

[9] M. Fang, G. Yang, N. Z. Gong, and J. Liu, "Poisoning attacks to graph-based recommender systems," in *ACSAC*, 2018.

[10] K. Christakopoulou and A. Banerjee, "Adversarial attacks on an oblivious recommender," in *RecSys*, 2019.

[11] C. Lin, S. Chen, H. Li, Y. Xiao, L. Li, and Q. Yang, "Attacking recommender systems with augmented user profiles," in *CIKM*, 2020.

[12] M. Fang, N. Z. Gong, and J. Liu, "Influence function based data poisoning attacks to top-n recommender systems," in *WWW*, 2020.

[13] J. Tang, H. Wen, and K. Wang, "Revisiting adversarially learned injection attacks against recommender systems," in *ACM RecSys*, 2020.

[14] W. Fan, T. Derr, X. Zhao, Y. Ma, H. Liu, J. Wang, J. Tang, and Q. Li, "Attacking black-box recommendations via copying cross-domain user profiles," in *ICDE*. IEEE, 2021.

[15] X. Zhang, J. Chen, R. Zhang, C. Wang, and L. Liu, "Attacking recommender systems with plausible profile," *TIFS*, 2021.

[16] H. Huang, J. Mu, N. Z. Gong, Q. Li, B. Liu, and M. Xu, "Data poisoning attacks to deep learning based recommender systems," in *NDSS*, 2021.

[17] L. Chen, Y. Xu, F. Xie, M. Huang, and Z. Zheng, "Data poisoning attacks on neighborhood-based recommender systems," *TETT*, 2021.

[18] C. Wu, D. Lian, Y. Ge, Z. Zhu, and E. Chen, "Triple adversarial learning for influence based poisoning attack in recommender systems," in *KDD*, 2021.

[19] M. Zeng, K. Li, B. Jiang, L. Cao, and H. Li, "Practical cross-system shilling attacks with limited access to data," *AAAI*, 2023.

[20] H.-Y. Chiang, Y.-S. Chen, Y.-Z. Song, H.-H. Shuai, and J. S. Chang, "Shilling black-box review-based recommender systems through fake review generation," in *KDD*, 2023.

[21] S. Yang, L. Yao, C. Wang, X. Xu, and L. Zhu, "Review-incorporated model-agnostic profile injection attacks on recommender systems," in *ICDM*. IEEE, 2023.

[22] B. Mobasher, R. Burke, R. Bhaumik, and J. J. Sandvig, "Attacks and remedies in collaborative recommendation," *IEEE Intelligent Systems*, 2007.

[23] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning based recommender system: A survey and new perspectives," *CSUR*, 2019.

[24] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, 2020.

[25] N. Sachdeva and J. McAuley, "How useful are reviews for recommendation? a critical review and potential improvements," in *SIGIR*, 2020.

[26] L. Zheng, V. Noroozi, and P. S. Yu, "Joint deep modeling of users and items using reviews for recommendation," in *WSDM*, 2017.

[27] G. Zhou, Z. Han, S. Chen, B. Huang, L. Zhu, S. Khan, X. Gao, and L. Yao, "Adapting large multimodal models to distribution shifts: The role of in-context learning," *arXiv preprint arXiv:2405.12217*, 2024.

[28] M. Luo, X. Xu, Y. Liu, P. Pasupat, and M. Kazemi, "In-context learning with retrieved demonstrations for language models: A survey," *arXiv preprint arXiv:2401.11624*, 2024.

[29] Z. Zhao, W. Fan, J. Li, Y. Liu, X. Mei, Y. Wang, Z. Wen, F. Wang, X. Zhao, J. Tang *et al.*, "Recommender systems in the era of large language models (llms)," *TKDE*, 2024.

[30] Z. Wang, M. Gao, J. Yu, H. Ma, H. Yin, and S. Sadiq, "Poisoning attacks against recommender systems: A survey," *arXiv preprint arXiv:2401.01527*, 2024.

[31] R. M. Bell and Y. Koren, "Improved neighborhood-based collaborative filtering," in *SIGKDD Workshop*. sn, 2007, pp. 7–14.

[32] C.-H. Lee, Y.-H. Kim, and P.-K. Rhee, "Web personalization expert with combining collaborative filtering and association rule mining technique," *Expert Systems with Applications*, 2001.

[33] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, 2009.

[34] B. Mobasher, R. Burke, R. Bhaumik, and C. Williams, "Toward trustworthy recommender systems: An analysis of attack models and algorithm robustness," *TOIT*, 2007.

[35] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *WWW*, 2017.

[36] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "Lightgcn: Simplifying and powering graph convolution network for recommendation," in *SIGIR*, 2020.

[37] Z. Xu, H. Zeng, and Q. Ai, "Understanding the effectiveness of reviews in e-commerce top-n recommendation," in *ICTIR*, 2021.

[38] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *NIPS*, 2020.

[39] J. Chua, Y. Li, S. Yang, C. Wang, and L. Yao, "Ai safety in generative ai large language models: A survey," *arXiv preprint arXiv:2407.18369*, 2024.

[40] R. Mohawesh, S. Xu, S. N. Tran, R. Ollington, M. Springer, Y. Jararweh, and S. Maqsood, "Fake reviews detection: A survey," *IEEE Access*, 2021.

[41] P. Wu, N. Moustafa, S. Yang, and H. Guo, "Densely connected residual network for attack recognition," in *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*. IEEE, 2020, pp. 233–242.

[42] S. Yang, H. Guo, and N. Moustafa, "Hunter in the dark: Discover anomalous network activity using deep ensemble network," in *2021 IEEE 21st International Conference on Software Quality, Reliability and Security (QRS)*. IEEE, 2021, pp. 829–840.

[43] S. Yang, P. Wu, and H. Guo, "Dualnet: Locate then detect effective payload with deep attention network," in *2021 IEEE Conference on Dependable and Secure Computing (DSC)*. IEEE, 2021, pp. 1–8.

[44] S. Yang, "Deep neural networks for network intrusion detection," Master's thesis, University of New South Wales (Australia), 2021.

[45] Y. Huang, Y. Gu, J. Xu, Z. Zhu, Z. Chen, and X. Ma, "Securing reliability: A brief overview on enhancing in-context learning for foundation models," *arXiv preprint arXiv:2402.17671*, 2024.

[46] S. Yang, Z. Hu, X. Li, C. Wang, T. Yu, X. Xu, L. Zhu, and L. Yao, "Drunkagent: Stealthy memory corruption in llm-powered recommender agents," *arXiv preprint arXiv:2503.23804*, 2025.

[47] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou *et al.*, "Chain-of-thought prompting elicits reasoning in large language models," *NeurIPS*, 2022.

[48] S. Yang, C. Wang, X. Xu, L. Zhu, and L. Yao, "Attacking visually-aware recommender systems with transferable and imperceptible adversarial styles," in *CIKM*, 2024, pp. 2900–2909.
[49] E. Reif, D. Ippolito, A. Yuan, A. Coenen, C. Callison-Burch, and J. Wei, "A recipe for arbitrary text style transfer with large language models," *arXiv preprint arXiv:2109.03910*, 2021.
[50] J. McAuley, C. Targett, Q. Shi, and A. Van Den Hengel, "Image-based recommendations on styles and substitutes," in *SIGIR*, 2015.
[51] Y. Hu, Y. Koren, and C. Volinsky, "Collaborative filtering for implicit feedback datasets," in *ICDM*, 2008.
[52] F. Wu, M. Gao, J. Yu, Z. Wang, K. Liu, and X. Wang, "Ready for emerging threats to recommender systems? a graph convolution-based generative shilling attack," *Information Sciences*, 2021.
[53] J. Zhang, Y. Liu, Q. Liu, S. Wu, G. Guo, and L. Wang, "Stealthy attack on large language model based recommendation," *ACL*, 2024.