Sim4Rec: Data-Free Model Extraction Attack on Sequential Recommendation

Yihao Wang[1], Jiajie Su[1], Chaochao Chen[1], Meng Han[1], Chi Zhang[2], Jun Wang[3]*

[1]Zhejiang University
[2]Independent Researcher
[3]OPPO Research Institute
{yihaowang, sujiajie, zjuccc, mhan}@zju.edu.cn, zhangchi900207@gmail.com, junwang.lu@gmail.com

Abstract

Model extraction attack shows promising performance in revealing sequential recommendation (SeqRec) robustness, e.g., as an upstream task of transfer-based attack to provide optimization feedback for downstream attacks. However, existing work either heavily relies on impractical prior knowledge or has impressive attack performance. In this paper, we focus on data-free model extraction attack on SeqRec, which aims to efficiently train a surrogate model that closely imitates the target model in a practical setting. Conducting such an attack is challenging. First, imitating sequential training data for accurate model extraction is hard without prior knowledge. Second, limited queries for the target model require the attack to be efficient. To address these challenges, we propose a novel adversarial framework **Sim4Rec** which includes two modules, i.e., *controllable sequence generation* and *reinforced adversarial distillation*. The former allows a sequential generator to produce synthetic data similar to training data through pre-training with controllable generated samples. The latter efficiently extracts the target model via reinforced adversarial knowledge distillation. Extensive experiments demonstrate the advancement of **Sim4Rec**.

1 Introduction

In the contemporary age of information abundance, sequential recommendation (SeqRec) holds a crucial position in various domains (Guy et al. 2010; Okura et al. 2017; Yue et al. 2021a; Su et al. 2023b,a), serving as a driver in guiding users to discover content they are interested in. However, as concerns about user privacy grow and various data privacy protection policies are enacted, e.g., CCPA[1], the security issues of SeqRec are increasingly in the spotlight.

Recently, adversarial and privacy attack methods on SeqRec have been widely researched (Deldjoo, Noia, and Merra 2021). Among these attack methods, an advanced technique named transfer-based attack achieves the promising performance (Tang, Wen, and Wang 2020; Lin et al. 2020; Zhang et al. 2020; Wu et al. 2021; Lin et al. 2022; Zhu et al. 2023b). As Fig.1 shows, transfer-based attack can be split into two steps, i.e., model extraction attack and downstream attack. In first step, they train a surrogate model as a replacement for the target (victim) model. In second step, they perform various downstream attacks on the surrogate model and optimize attack performance based on feedback from the surrogate model. Therefore, downstream attack performance heavily relies on the former model extraction attack which is necessary to be further studied. Nevertheless, as blue rectangle shown in Fig.1, most existing model extraction attacks on SeqRec either have impractical assumptions that the attacker has full or partial prior knowledge of the target model, i.e., training data, parameter, and architecture, or demonstrate unimpressive attack performance.

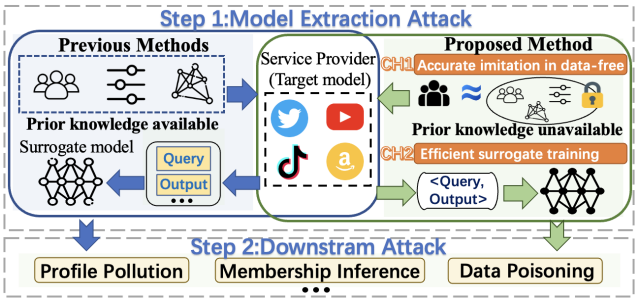


Figure 1: Compared to previous work in data-driven, the pipeline of model extraction attack in data-free setting.

In this paper, we focus on model extraction attack on SeqRec, considering a more stringent and practical *data-free* setting, where the attacker has no access to the target model's prior knowledge and limited budgets to query the target model. In such scenario, as illustrated by green rectangle in Fig.1, the attacker first generates synthetic data, querying the target model to obtain ranked lists as output. Then the tuple $\langle \text{Query}, \text{Output} \rangle$ is utilized to train a surrogate model that closely imitates the target model for attack.

Unfortunately, conducting remarkable data-free model extraction attack on SeqRec remains non-trivial due to the following challenges: **CH1: How to imitate sequential training data accurately in data-free setting.** To achieve accurate data-free model extraction, it is essential to generate synthetic sequences that closely approximate the distribution of training data. However, due to the tremendous discrete sample space and the lack of prior knowledge, the generation of sequences similar to training data remains a chal-

*Jun Wang is the corresponding author.

[1]https://oag.ca.gov/privacy/ccpa

lenge. **CH2: How to train the surrogate model efficiently with limited queries.** During the training of the surrogate model, the contribution of different samples in synthetic data to the improvement of extraction performance varies. Previous works ignore this distinction, making excessive queries on easy samples with minimal performance improvement, while such consumption is not practical in real-world setting. Thus how to make data-free model extraction attack efficient with limited queries remains unsolved.

To address challenges, we propose a data-free model extraction attack method **Sim4Rec** on SeqRec, aiming to perform accurate and efficient extraction with limited queries. To achieve that, we utilize two modules, i.e., controllable sequence generation and reinforced adversarial distillation. *Controllable sequence generation* reproduces samples and employs them to pre-train a sequential generator, which enables the generator to produce synthetic data similar to training data **(CH1)**. In this module, directly utilizing autoregressive generation from natural language processing (Carlini et al. 2021) may introduce significant bias between synthetic and training data. Thus we design two penalty factors to control sequence generation, i.e., (i) item-level penalty factor and (ii) sequence-level penalty factor. The former leads to synthetic sequences containing more low-frequency items, and the latter constrains the number of repeated items in one synthetic sequence. *Reinforced adversarial distillation* trains a surrogate model efficiently that closely imitates the target model via reinforced adversarial knowledge distillation **(CH2)**. In this module, the key to enhancing extraction efficiency lies in distilling **hard samples that contribute significantly to model discrepancy between the surrogate and target models**. To achieve that, we propose reinforced adversarial training to synthesize hard samples for efficient extraction. Firstly, we employ the former pre-trained sequential generator to create hard samples that enlarge model discrepancy. Secondly, the surrogate model minimizes model discrepancy based on these samples via knowledge distillation and backward reward for generator optimization. By iterating these two steps, the sequential generator could continuously produce hard samples guided by reward from surrogate model, and the surrogate model could simulate the target model efficiently through knowledge distillation on hard samples produced by the generator.

Our main contributions are as follows: (1) We propose a novel adversarial framework **Sim4Rec** for data-free model extraction attack on SeqRec, which efficiently extracts the target model with limited queries. (2) **Sim4Rec** utilizes controllable sequence generation and reinforced adversarial distillation to generate high-quality synthetic data that resembles training data and significantly contributes to model discrepancy, which facilitates accurate extraction attack. (3) Extensive experiments on three benchmark datasets demonstrate that **Sim4Rec** outperforms existing methods in extraction performance and improves downstream task outcomes.

2 Related Work

Data Synthesis for Recommender System. Data synthesis is an effective technology that has been widely used to solve class imbalance and privacy-preserving. (Wang et al. 2019) adopts Generative Adversarial Network (Goodfellow et al. 2014) to generate training data, relieving the data sparsity issue. (Liu et al. 2022) generates synthetic data for users based on their preferences for privacy-preserving. For sequential recommender systems, (Wang et al. 2021) proposes a counterfactual framework to enhance the performance of recommendation through data augmentation. (Dang et al. 2023) augments sequence data from the perspective of the time interval. The above methods all assume that real training data is available which contradicts the practical data-free setting.

Data-Free Model Extraction Attack. Model extraction attacks threaten recommenders and user privacy by transferring knowledge from the target to the surrogate model without permission. Several model extraction attacks have been proposed (Krishna et al. 2019; Zhou et al. 2020; Kariyappa, Prakash, and Qureshi 2021; Zhang, Chen, and Lyu 2022; Miura, Shibahara, and Yanai 2024), which can be classified into two types, i.e., *White-box* and *Black-box*. Data-free model extraction attack is the stringent scenario in *Black-box* that attacker has no prior knowledge of the target model, i.e., architecture, parameters, and training data. DaST (Zhou et al. 2020) is the first work of data-free model extraction in imagination classification, which adopts a generative adversarial network with multi-branch architecture and label-control loss to create synthetic data for surrogate model training. THIEVES (Krishna et al. 2019) shows that task-specific heuristics and random word sequences could achieve model extraction across a diverse set of NLP tasks. (Carlini et al. 2024) proposes the first model extraction on large language model. In contrast, there is limited work on data-free model extraction attack targeting recommendation (Zhu et al. 2023a). (Yue et al. 2021b) generates synthetic data heuristically, proving the possibility of model extraction on SeqRec. The above methods either face transfer challenges due to task differences or struggle to generate high-quality synthetic data for effective model extraction attack.

3 Preliminaries

3.1 Sequential Recommendation Framework

Sequential Recommendation (SeqRec) is widely applied which trains a personalized recommender system with the user's historical behavior sequences. Let U and V denote the set of users and items, respectively, where $u \in U$ denotes a user and $v \in V$ denotes an item. User u is constructed with a chronologically-ordered behavior sequence $u_{1:n} = [v_1, v_2, \ldots, v_n]$, where n denotes the sequence length and v_i is the i-th item that user u has interacted with. Generally, a sequential recommender f_θ encodes $u_{1:n}$ and obtains the hidden representation $\mathcal{H} = f_\theta(u_{1:n})$. The goal of SeqRec is to predict the next item v_{n+1} that user u would interact with, thus it can be formulated as finding the optimal encoder parameter θ^* that maximizes the probability:

$$\theta^* = \arg\max_\theta \log p(v_{n+1}|\mathcal{H}), \tag{1}$$

which can be calculated by minimizing the binary cross-entropy loss as follows:

$$\mathcal{L}_{rec} = CE(v_{pos}, v_{neg}, \mathcal{H}), \tag{2}$$

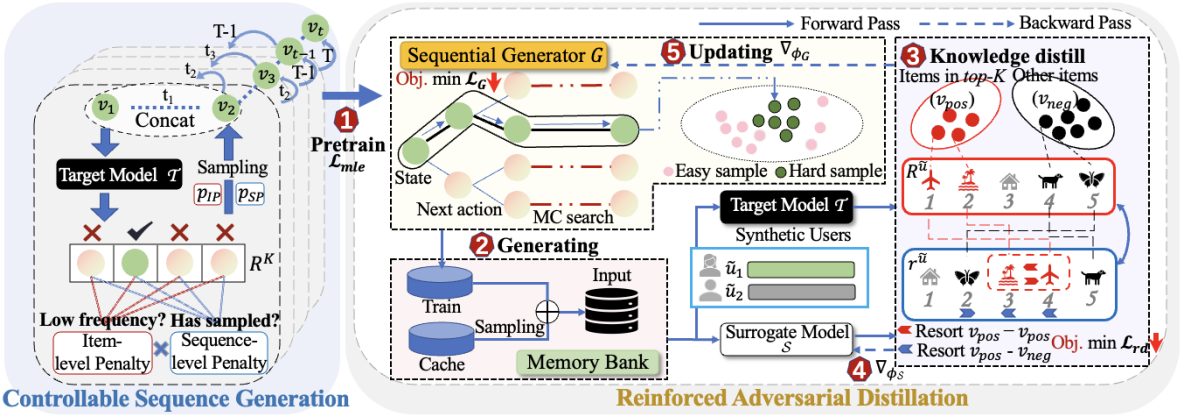


Figure 2: The framework of **Sim4Rec**.

where v_{pos} and v_{neg} denote the items that user has interacted with or not. Then the *top-K* items predicted by SeqRec could be recommended to user u.

3.2 Data-free Model Extraction Attack in SeqRec

We define the pipeline of data-free model extraction in SeqRec. Given a target model $\mathcal{T}$ and a surrogate model $\mathcal{S}$, $\mathbb{P}(\tilde{U})$ denotes the distribution of synthetic user sequences $\tilde{U}$, the objective of model extraction attack is to train an optimal surrogate model $\mathcal{S}^*$ that minimizes the model discrepancy:

$$\mathcal{S}^* = \arg\min_{\phi_\mathcal{S}} \mathcal{D}(\mathcal{T}(\tilde{u}), \mathcal{S}(\tilde{u})) \quad s.t. \tilde{u} \sim \mathbb{P}(\tilde{U}), \tag{3}$$

where $\phi_\mathcal{S}$ denotes parameters of $\mathcal{S}$, $\mathcal{D}$ is a distance function to measure model discrepancy. Different from data-driven model extraction, which has access to training data U, data-free model extraction solely relies on training the surrogate model through synthetic data $\tilde{U}$. Therefore, the similarity between synthesized sequences $\tilde{U}$ and training data U is the key to enhancing the performance of data-free extraction.

4 Method

In this subsection, we illustrate the framework of our proposed data-free model extraction attack. As shown in Fig.2, **Sim4Rec** has two main modules: 1) controllable sequence generation (CSG) and 2) reinforced adversarial distillation (ADV). *Controllable sequence generation* first pre-trains a sequential generator, aiming to produce synthesized data whose distribution is close to the real samples of training data. *Reinforced adversarial distillation* utilizes adversarial knowledge distillation to mine hard samples that is largely responsible for the model discrepancy, which enables the surrogate model to imitate the target model efficiently.

4.1 Controllable Sequence Generation

As discussed in Section 3.2, the similarity of synthesized data determines the attack performance. Therefore, to achieve an accurate model extraction attack, we propose to train a sequential generator G that produces synthesized data similar to training data without prior knowledge of the target model. Several researches (Carlini et al. 2021; Yue et al. 2021b) found that sequential model f_ϕ, given a prefix $[x_1, \ldots, x_{t-2}]$, could generate new sequences by iteratively following the procedure of sampling $\hat{x}_{t-1} \sim f_\phi(x_{t-1} \mid x_1, \ldots, x_{t-2})$ and feeding $\hat{x}_{t-1}$ back into the model to sample $\hat{x}_t \sim f_\phi(x_t \mid x_1, \ldots, \hat{x}_{t-1})$. Inspired by this, we first generate synthetic data autoregressively:

$$\begin{aligned} \hat{v}_t &\sim \mathcal{T}(v_t \mid v_1, \ldots, \hat{v}_{t-1}), \\ \tilde{x}_{1:t} &= v_1 \oplus \ldots \oplus \hat{v}_{t-1} \oplus \hat{v}_t, \end{aligned} \tag{4}$$

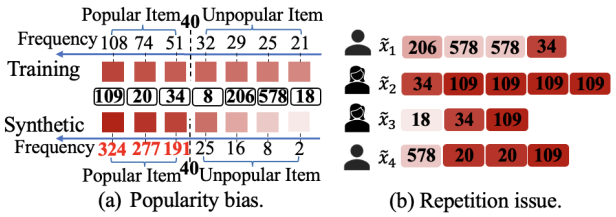
$\hat{v}_t$ denotes the sampled item, and $\oplus$ is the concatenate operation. Given an initial item $v_1 \in V$, through sampling and concatenating the output of the target model $\mathcal{T}$, the synthesized data $\tilde{x}_{1:t}$ is generated without prior knowledge.

However, directly adopting the above approach to produce synthetic data cannot effectively substitute for training data in SeqRec. As shown in Fig.3, we find that synthetic data generated by the autoregressive approach not only exhibits a substantial bias towards popular items but also holds serious repetition issues. These problems cause most synthetic data to resemble collections of popular items, which deviates from the actual distribution of training data. Therefore, to promote diversity and alleviate repetitions in sequence generation, we design two penalty factors based on item-level and sequence-level from the aspect of sampling strategy.

Item-level penalty factor. When input the prefix sequence $[v_1, \ldots, v_t](v_i \in V)$, the target model outputs *top-K* ranked list $R^K = [\overline{v}_1, \ldots, \overline{v}_K](\overline{v}_i \in V)$. Considering actual user sequences should encompass a certain proportion of long-tail (unpopular) items, we conduct a frequency analysis on the generated prefix sequences and retrieve the frequency $f_{\overline{v}_i}$ of each item in R^K. Subsequently, we calculate the sampling probability for each item $\overline{v}_i$ based on its frequency:

$$p_{IP} = \begin{cases} \frac{1}{f_{\overline{v}_i}+1}, & \text{if } f_{\overline{v}_i} < \varepsilon; \\ c \cdot \sqrt{\frac{\log f_{\overline{v}_i}}{f_{\overline{v}_i}+1}}, & \text{otherwise}, \end{cases} \tag{5}$$

where ε and c denote threshold and hyper-parameter, set to 3 and 0.5. This penalized sampling works by discounting the frequency of previously generated items. Through Eq.(5), the procedure of sampling can place more emphasis on long-tail items thereby enhancing diversity of synthesized data.



(a) Popularity bias. (b) Repetition issue.

Figure 3: The biases of autoregressive synthetic data.

Sequence-level penalty factor. In autoregressive sequence generation, repeated items occur within the same sequence, and in some instances, sequences resembling a mere concatenation of a single item which diminishes the semantic richness of the generated sequences. To address this, we design an Indicator function I to guide whether item $\overline{v}_i$ in R^K has been sampled in the input prefix sequence $[v_1, \ldots, v_t]$. Then we adjust the corresponding sampling probabilities based on the value of the indicator function:

$$p_{SP} = \frac{p_{IP}}{\theta} \quad \theta = 2 \text{ if } I(\overline{v}_i) \text{ is True else } 1, \tag{6}$$

we find $\theta = 2$ yields a good balance between truthful generation and lack of repetition. In the end, we adopt p_{SP} as the sampling probability of item $\overline{v}_i$ in R^K, selecting suitable item $\hat{v}$ in Eq.(4) to enhance diversity and alleviate repetition.

After generating synthetic data $\tilde{X}$, we pre-train sequential generator G using MLE (Dutilleul 1999) for E_{mle} epochs:

$$\mathcal{L}_{mle} = -\sum_{\tilde{x} \in \tilde{X}} \log P(\tilde{x} \mid \phi_G), \tag{7}$$

where ϕ_G denotes the parameters of the sequential generator, and $P(\tilde{x} \mid \phi_G)$ is the probability of generating $\tilde{x}$. By minimizing Eq.(7), we obtain the optimal parameters ϕ_G^*. Once pre-trained, the generator could synthesize user sequences similar to training data with diverse features.

4.2 Reinforced Adversarial Distillation

In real-world setting, the attacker needs to refrain from excessive access to the target model due to stealth reasons, thus limited queries for the target model require model extraction attack to be efficient. To achieve that, we propose reinforced adversarial distillation to automatically mine hard synthetic data and efficiently conduct model extraction attack.

Adversarial knowledge distillation. The goal of data-free model extraction, as defined in Eq.(3), is to improve extraction performance by minimizing model discrepancy. Different samples contribute variably to discrepancy, affecting the enhancement of extraction performance. Distillation based on these hard samples that substantially contribute to discrepancy yields significant improvements in extraction performance. Therefore, enhancing model extraction efficiency relies on distilling knowledge from hard samples, rather than wasting queries on easy samples. Several researches (Fang et al. 2019; Micaelli and Storkey 2019; Heo et al. 2019) proved the feasibility of mining hard samples via adversarial training. Inspired by this, we propose reinforced adversarial distillation to continuously generate hard samples for efficient data-free model extraction.

We treat the surrogate model training as a generative adversarial network (GAN), the pre-trained sequential generator as generator G, and the combination of the target and the surrogate models as discriminator D. Given a batch of synthesized data $\tilde{u} = [v_1, v_2, \ldots, v_T](v_i \in V)$ produced by G, the discriminator D aims to decrease the model discrepancy on $\tilde{u}$, and the generator G conversely synthesizes hard samples which increase the model discrepancy. Thus the value function of adversarial training is formulated as follows:

$$\max_G \min_D \mathcal{L}_{adv} = dis(\mathcal{T}(\tilde{u}), \mathcal{S}(\tilde{u})), \tag{8}$$

where dis is a distance function to measure the output difference between $\mathcal{T}(\tilde{u})$ and $\mathcal{S}(\tilde{u})$. For the target model $\mathcal{T}$, as a sequential recommender system, its output is *top-K* ranked list without probability values, which represents the hard-label scenario. Inspired by (Kang et al. 2020), we model dis with listwise ranking loss in this scenario. For synthetic user $\tilde{u}$, $R^{\tilde{u}}$ is a sorted list of all the sampled items recommended by the target model, including positive samples v_{pos} in the *top-K* ranked list and negative samples v_{neg} are not. K, N denotes $|v_{pos}|$ and $|v_{neg}|$ respectively. $r^{\tilde{u}}$ represents ranking scores on the sampled items predicted by the surrogate model. The permutation probability can be formulated as:

$$p\left(r^{\tilde{u}} \mid R^{\tilde{u}}_{K+N}\right) = \prod_{k=1}^{K} \frac{\exp\left(r^{\tilde{u}}_{R_k}\right)}{\sum_{i=k}^{K} \exp\left(r^{\tilde{u}}_{R_i}\right) + \sum_{j=1}^{N} \exp\left(r^{\tilde{u}}_{R_j}\right)}, \tag{9}$$

$r^{\tilde{u}}_{R_k}$ is the ranking score predicted by the surrogate model for the k-th item in $R^{\tilde{u}}$, Our purpose is to maximize the log-likelihood $\log p\left(r^{\tilde{u}} \mid R^{\tilde{u}}_{K+N}\right)$ based on the *top-K* ranked list recommended by the target model. Then we define the $\mathcal{L}_{rd}$ to distill the knowledge of the target model as follows:

$$\mathcal{L}_{rd} = -\sum_{\tilde{u} \in \tilde{U}} \log p(r^{\tilde{u}} \mid R^{\tilde{u}}_{K+N}). \tag{10}$$

We adopt $\mathcal{L}_{rd}$ to measure the model discrepancy as dis in Eq.(8). During training, minimizing $\mathcal{L}_{rd}$ equals to maximizing the likelihood of permutation probability which keeps the surrogate model similar to the target model. The loss function of sequential generator G is designed as:

$$\mathcal{L}_G = -dis(\mathcal{T}(\tilde{u}), \mathcal{S}(\tilde{u})). \tag{11}$$

Eq.(11) encourages the generator to produce hard samples which increase the model discrepancy between $\mathcal{T}$ and $\mathcal{S}$.

Discrete gradient optimization. However, because synthetic user sequences are discrete, gradients from Eq.(11) cannot be directly optimized backward to the generator. Following (Sutton et al. 1999; Yu et al. 2017), we employ reinforcement learning to address the challenge of discrete gradient backpropagation. We define the generation of synthesized data as a multi-step decision problem. Suppose that a synthetic sequence $\tilde{u}_{1:T} = [v_1, v_2, \ldots, v_T](v_i \in V)$ needs to be produced, and $\tilde{u}_{1:t}$ has been generated, taking the generated sequence $\tilde{u}_{1:t}$ as state s_t for the current moment t, the objective of the sequential generator G is to generate the sequence that maximizes the rewards from the start state s_1:

$$J(\phi_G) = \mathbb{E}\left[R_T \mid s_1, \phi_G\right] = \sum_{v_i \in V} G\left(v_i \mid s_1\right) Q_{\phi_G}(s_1, v_i), \tag{12}$$

where R_T is the reward for entire synthesized sequence $\tilde{u}_{1:T}$, ϕ_G represents parameters of G. $Q_{\phi_G}(s,a)$ is action-value function, which indicates the cumulative reward associated with selecting item v_i under the policy generator G, initiated from state s_1 and taking action a. For the generated sequence $\tilde{u}_{1:t-1}$, state $s = \tilde{u}_{1:t-1}$ and action $a = v_t$, the target and the surrogate models provide the recommended list accordingly. We adopt the value of model discrepancy dis as the reward, thereby maximizing the rewards is equivalent to minimizing $\mathcal{L}_G$. Formally, $Q_{\phi_G}(s,a)$ can be derived as:

$$Q_{\phi_G}\left(s = \tilde{u}_{1:t-1}, a = v_t\right) = dis(T(\tilde{u}_{1:t}), S(\tilde{u}_{1:t})). \quad (13)$$

Considering the attacker has limited resources to process tremendous data, we simplify the Monte Carlo search used in (Yu et al. 2017) by setting $t = T$ in the computation of Q_{ϕ_G} when $t < T$. Thus Eq.(13) is transformed as:

$$Q_{\phi_G}\left(s = \tilde{u}_{1:T-1}, a = v_T\right) = dis(T(\tilde{u}_{1:T}), S(\tilde{u}_{1:T})). \quad (14)$$

In the end, the gradient of the objective function $J(\phi_G)$ can be formulated as:

$$\begin{aligned}
&\nabla_{\phi_G} J(\phi_G) \\
&= \mathbb{E}_{\tilde{u}_{1:t-1}\sim G}[\sum_{v_i \in V} \nabla_{\phi_G} G(v_t \mid \tilde{u}_{1:t-1}) Q_{\phi_G}(\tilde{u}_{1:t-1}, v_t)] \\
&\simeq \mathbb{E}_{v_t \sim G(v_t|\tilde{u}_{1:t-1})}[\nabla_{\phi_G} \log G(v_t \mid \tilde{u}_{1:t-1}) Q_{\phi_G}(\tilde{u}_{1:t-1}, v_t)].
\end{aligned} \quad (15)$$

Then update the generator by $\phi_G \leftarrow \phi_G + \gamma \nabla_{\phi_G} J(\phi_G)$, where γ is the learning rate. This reinforced adversarial training allows for continuously generating hard samples and narrowing the model discrepancy, significantly enhancing the efficiency of model extraction attack.

Catastrophic forgetting prevention. To address the performance decline caused by catastrophic forgetting of GAN, we set a memory bank after the generator, composed of train, cache, and input pool. The capacity of the input pool is $M_{bank} * b$ (batch-size), when the generator produces b latest synthetic sequences to the train pool, $(M_{bank} - 1) * b$ sequences are selected from the cache pool which stores generated sequences. Subsequently, two data sets are merged and fed into the input pool for knowledge distillation.

After iterative optimization, high-quality synthetic data is generated, which is not only similar to training data but also greatly contributes to surrogate model training, facilitating an accurate and efficient data-free model extraction attack.

5 Experiments

In this section, we conduct experiments to address the following research questions. **RQ1**: How is the effectiveness of **Sim4Rec** compared with existing model extraction methods? **RQ2**: How does the query budget influence the attack performance? **RQ3**: How does each designed module in **Sim4Rec**, i.e., *controllable sequence generation* and *reinforced adversarial distillation*, contributes to the performance? **RQ4**: How does the setting of hyper-parameters values impact performance? **RQ5**: How does the extracted surrogate model impact the performance of downstream tasks?

Datasets	#Users	#Items	#Actions	Avg. length	Density
ML-1M	6,040	3,416	1.0M	163.5	4.79%
Steam	334,730	13,047	3.7M	10.6	0.08%
Beauty	40,226	54,542	0.4M	8.8	0.02%

Table 1: Statistics of datasets.

5.1 Experimental Setup

Datasets and models. We evaluate our method on three benchmark datasets, i.e., **MovieLens-1M (ML-1M)**, **Steam**, and **Beauty**. Two representative sequential recommenders, i.e., RNN-based **NARM** (Li et al. 2017), Transformer-based **SASRec** (Kang and McAuley 2018), are employed as the target model, which outputs top-100 recommended items. We reserve the last two items in each sequence for validation and testing, and the remaining items used for training. Details of datasets are shown in Table 1.

Evaluation protocols. Following (Yue et al. 2021b), We evaluate our method from three metrics. (1) **Recall at top K (Recall@K)** is to evaluate the effectiveness of ranking; (2) **Normalized Discounted Cumulative Gain at top K (NDCG@K)** is to measure the ranking quality; (3) **Agreement in top K (Agr@K)** denoted as $\text{Agr@K} = \frac{|R_{\mathcal{T}}^K \cap R_{\mathcal{S}}^K|}{K}$ is to evaluate the output similarity between the target and surrogate models, where $R_{\mathcal{T}}^K$ and $R_{\mathcal{S}}^K$ are the *top-K* ranked list of the target and surrogate models, respectively.

Comparison methods. As discussed in related work, there is currently limited research on data-free model extraction attack in SeqRec, thus we compare our method with following baselines: (1) **Random** samples items uniformly from the discrete item space to form sequences. (2) **DFKD** (Wang et al. 2023) synthesizes continuous data by leveraging the target model's parameters for knowledge distillation. (3) **DFME** (Yue et al. 2021b) generates autoregressive synthetic data for data-free model extraction without prior knowledge.

Implementation details. We utilize Gated Recurrent Unit (GRU) as the architecture of the sequential generator. Adam optimizer with learning rate $\gamma = 0.01$, weight decay $\eta = 0.01$, and batch size $b = 128$ are adopted. We set the allowed sequence lengths of ML-1M, Steam, and Beauty to $\{200, 50, 50\}$, the hyper-parameters to E_{mle} = 30 and M_{bank} = 100.

5.2 Model Extraction Performance (RQ1-RQ2)

Model comparison (RQ1). As the attacker has no access to prior knowledge of the target model, we comprehensively evaluate the extraction performance in two scenarios. The first is the target and surrogate models have the same architecture, i.e., **N-N** (NARM to NARM) and **S-S** (SASRec to SASRec). The second is the target and surrogate models use different architectures, i.e., **N-S** (NARM to SASRec) and **S-N** (SASRec to NARM). We evaluate all methods on three datasets, results are reported in Table 2 with fixed query budgets Q = 1k. From Table 2, we observe that: (1) Comparing attack performance between two scenarios on three datasets, the same architecture achieves higher ex-

Option	Method	ML-1M				Steam				Beauty			
		N@10	R@10	Agr@1	Agr@10	N@10	R@10	Agr@1	Agr@10	N@10	R@10	Agr@1	Agr@10
N-N	*Target*	0.622	0.812	-	-	0.627	0.846	-	-	0.354	0.515	-	-
	Random	0.599	0.809	0.350	0.568	0.620	**0.846**	0.372	0.587	0.274	0.413	0.191	0.326
	DFKD	0.603	0.810	0.512	0.708	0.616	0.839	0.392	0.597	0.272	0.395	0.201	0.347
	DFME	0.607	0.806	0.510	0.689	0.613	0.830	0.403	0.611	0.258	0.362	0.207	0.364
	Sim4Rec	**0.610**	**0.812**	**0.519**	**0.710**	**0.622**	0.845	**0.423**	**0.642**	**0.290**	**0.416**	**0.230**	**0.398**
S-S	*Target*	0.607	0.823	-	-	0.616	0.836	-	-	0.338	0.489	-	-
	Random	0.450	0.699	0.181	0.332	0.545	0.775	0.201	0.206	0.177	0.333	0.001	0.036
	DFKD	0.524	0.781	0.312	0.514	0.564	0.796	0.324	0.491	0.291	0.398	0.041	0.108
	DFME	0.549	0.790	0.328	0.502	0.591	0.818	0.367	0.502	0.298	0.409	0.052	0.125
	Sim4Rec	**0.600**	**0.828**	**0.556**	**0.723**	**0.612**	**0.834**	**0.468**	**0.645**	**0.307**	**0.458**	**0.219**	**0.242**
N-S	*Target*	0.622	0.812	-	-	0.627	0.846	-	-	0.354	0.515	-	-
	Random	0.544	0.765	0.251	0.427	**0.609**	**0.832**	0.363	0.525	0.255	0.396	0.110	0.220
	DFKD	0.547	0.783	0.327	0.563	0.603	0.791	0.316	0.511	0.211	0.389	0.174	0.287
	DFME	0.563	0.779	0.322	0.549	0.604	0.820	0.323	0.512	0.180	0.371	0.178	0.334
	Sim4Rec	**0.567**	**0.788**	**0.335**	**0.585**	0.607	0.828	**0.368**	**0.561**	**0.284**	**0.418**	**0.183**	**0.369**
S-N	*Target*	0.607	0.823	-	-	0.616	0.836	-	-	0.338	0.489	-	-
	Random	0.338	0.567	0.038	0.158	0.375	0.557	0.035	0.126	0.140	0.288	0.002	0.034
	DFKD	0.461	0.731	0.189	0.372	0.486	0.732	0.048	0.154	0.198	0.350	0.032	0.066
	DFME	0.473	0.727	0.171	0.354	0.511	0.760	0.076	0.187	0.235	0.415	0.038	0.075
	Sim4Rec	**0.596**	**0.819**	**0.459**	**0.644**	**0.609**	**0.833**	**0.400**	**0.579**	**0.299**	**0.456**	**0.057**	**0.192**

Table 2: Comparison results of extraction performance under Q= 1k, with original performance of the *Target* model.

traction performance than the different, and denser datasets demonstrate better performance than sparse ones. (2) The naive Random shows the worst performance in all scenarios because the synthetic data is just a combination of items, with no sequential features. (3) DFKD and DFME achieve a better result than Random, but these two methods still perform poorly in some settings, i.e., {S-S, Beauty} and {S-N, Steam}. (4) Our method **Sim4Rec** achieves the best performance in most scenarios, which indicates that controllable sequence generation effectively enables the generator to produce synthetic data that closely simulates training data, which results in impressive model extraction performance. (5) Comparing **Sim4Rec** performance in N-S with S-N, S-N yields better extraction results in most scenarios. The phenomenon could be attributed that the target model (SASRec) generates high-quality synthetic data through our method, consequently improving the extraction performance, despite the simple structure of the surrogate model Narm. This also proves that **Sim4Rec** has better attack performance when targeting transformer-based sequential models like SASRec.

Impact of query budget (RQ2). To evaluate efficiency and stability of **Sim4Rec**, we employ different query budgets and present results on Beauty in Fig.4. The results show that: (1) With query budgets increasing, Agr@10 of all methods becomes higher. Notably, **Sim4Rec** outperforms all baselines when query budgets vary from Q = {0.5k, 1k, 2k, 3k, 5k}, indicating advancement and stability. (2) **Sim4Rec** achieves better efficiency with fewer queries for considerable attack performance. For instance, **Sim4Rec** achieves 0.374 Agr@10 under {N-S, 2k}, while Q = 5k is required for DFME to achieve comparable performance. This can be explained by reinforced adversarial distillation, continuously searching hard samples that cause substantial model discrepancy for surrogate model training, thus efficiently extracting the target model. (3) Compared to other methods, **Sim4Rec** is still effective with a mini query budget. As the results under {S-N, 0.5k}, the performance of other methods drops close to zero, **Sim4Rec** still achieves 0.109 Agr@10.

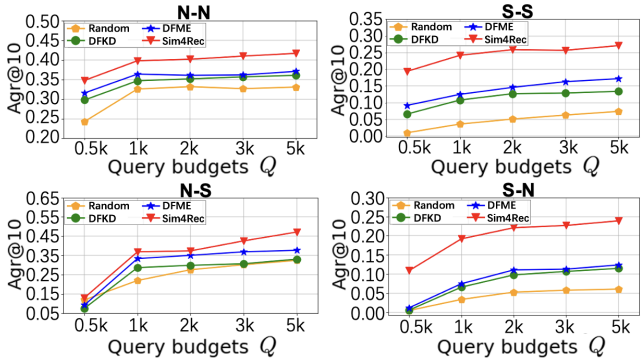


Figure 4: Comparison results under different query budgets.

5.3 In-depth Model Analysis (RQ3-RQ4)

Ablation study (RQ3). To investigate contributions of two modules introduced in **Sim4Rec**, we conduct ablation study on following variants: (i) ***w/o IP*** removes Item-level penalty factor. (ii) ***w/o SP*** removes Sequence-level penalty factor. (iii) ***w/o CSG*** removes controllable sequence generation for pre-training. (iv) ***w/o ADV*** removes reinforced adversarial distillation, training the surrogate model with classical knowledge distillation. Results are presented in Table 3, we further visualize the distribution of synthetic data with four circumstances via t-SNE in Fig.5 to show two modules' contributions. Note that we utilize DFME (Yue et al. 2021b) to represent ***w/o CSG & w/o ADV*** in Fig.5(a). From these re-

Method	ML-1M				Steam				Beauty				Avg. decline
	N-N	S-S	N-S	S-N	N-N	S-S	N-S	S-N	N-N	S-S	N-S	S-N	
Sim4Rec	**0.710**	**0.723**	**0.585**	**0.644**	**0.642**	**0.645**	**0.561**	**0.579**	**0.398**	**0.242**	**0.369**	**0.192**	-
w/o IP	0.647	0.675	0.430	0.632	0.620	0.625	0.551	0.563	0.351	0.122	0.154	0.131	0.066 ↓
w/o SP	0.696	0.692	0.481	0.625	0.631	0.621	0.531	0.557	0.343	0.174	0.209	0.138	0.049 ↓
w/o CSG	0.607	0.529	0.356	0.449	0.587	0.515	0.311	0.476	0.377	0.102	0.142	0.074	0.147 ↓
w/o ADV	0.653	0.667	0.467	0.592	0.604	0.612	0.537	0.542	0.320	0.165	0.263	0.093	0.065 ↓

Table 3: Ablation study on Sim4Rec with Agr@10, Avg. decline denotes the average decline of the performance.

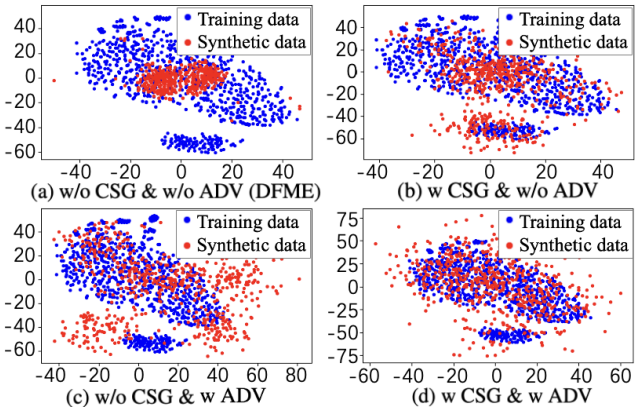


Figure 5: Synthetic data embeddings on Steam.

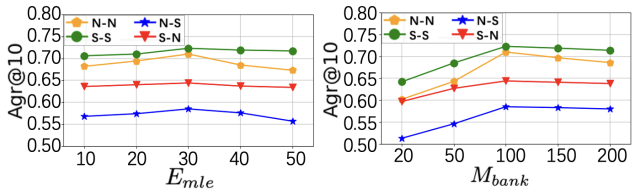


Figure 6: Effects of E_{mle} and M_{bank} on ML-1M.

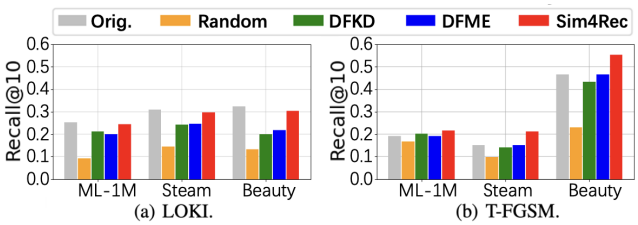


Figure 7: Comparison of data poisoning attack on different model extraction methods under S-S, the vertical axis is Recall@10 of the target item, Orig. denotes the performance of transfer-based attack on their original surrogate model.

sults, we find that: (1) Item-level and Sequence-level penalty factors relieve popularity bias and repetition issue in sequence generation, which can be demonstrated by the performance of ***w/o IP*** and ***w/o SP***. (2) Fig.5(b) shows that CSG makes the synthetic data similar to training data instead of forming a centralized distribution like DFME. (3) The weak performance of ***w/o CSG*** suggests that CSG is crucial for model extraction. Simultaneously, as Fig.5(c) shows, using only ADV leads to synthetic data with a distribution diverging from the training data. (4) Through Fig.5(d) and performance gap between ***w/o ADV*** and **Sim4Rec**, ADV improves extraction performance by unifying synthetic data.

Parameter analysis (RQ4). We study hyper-parameters, i.e., the epoch of pre-training E_{mle} and memory bank size M_{bank}. We vary E_{mle} in $\{10, 20, 30, 40, 50\}$ and report results in Fig.6. From the results we conclude that insufficient epochs of pre-training prevent the generator from producing synthetic data similar to training data, while excessive epochs lead to overfitting of the generator to the pre-training data, hindering subsequent adversarial training and resulting in a decline in extraction performance. Then we vary M_{bank} in $\{20, 50, 100, 150, 200\}$. The results indicate that an optimal memory bank capacity, neither too large nor too small, is crucial for effectively addressing catastrophic forgetting, ultimately leading to improved model extraction performance.

5.4 Impact on Downstream Task (RQ5)

Model extraction is employed as upstream task for transfer-based attack, thus we evaluate **Sim4Rec** downstream effect, i.e., data poisoning attack (Li et al. 2016). Two transfer-based attacks are utilized for evaluation: **LOKI** and **T-FGSM**. The original surrogate model in LOKI and T-FGSM is replaced with the one extracted by methods in Section 5.2, aiming to explore the impact on downstream attacks. From results in Fig.7: (1) Downstream attack performance correlates positively with model extraction due to increased similarity of the surrogate model, enhancing transferability of poison samples and consequently improving attack effectiveness. (2) For LOKI, whose surrogate model is learned with training data, **Sim4Rec** achieves comparable performance with Orig. among all methods. This demonstrates our method still achieves remarkable attack performance even in a more stringent setting. (3) For T-FGSM, **Sim4Rec** also contributes to an improved attack performance.

6 Conclusion

We propose **Sim4Rec** for effective data-free model extraction attack on SeqRec, including *controllable sequence generation (CSG)* and *reinforced adversarial distillation (ADV)*. *CSG* pre-trains generator to create training-like samples, *ADV* efficiently extracts the target model via adversarial distillation. Copious experiments show **Sim4Rec** advancement.

Acknowledgments

This work was supported in part by the National Key R&D Program of China (No. 2022YFB4501504).

References

Carlini, N.; Paleka, D.; Dvijotham, K. D.; Steinke, T.; Hayase, J.; Cooper, A. F.; Lee, K.; Jagielski, M.; Nasr, M.; Conmy, A.; et al. 2024. Stealing part of a production language model. *arXiv preprint arXiv:2403.06634*.

Carlini, N.; Tramer, F.; Wallace, E.; Jagielski, M.; Herbert-Voss, A.; Lee, K.; Roberts, A.; Brown, T.; Song, D.; Erlingsson, U.; et al. 2021. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, 2633–2650.

Dang, Y.; Yang, E.; Guo, G.; Jiang, L.; Wang, X.; Xu, X.; Sun, Q.; and Liu, H. 2023. Uniform sequence better: Time interval aware data augmentation for sequential recommendation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, 4225–4232.

Deldjoo, Y.; Noia, T. D.; and Merra, F. A. 2021. A survey on adversarial recommender systems: from attack/defense strategies to generative adversarial networks. *ACM Computing Surveys (CSUR)*, 54(2): 1–38.

Dutilleul, P. 1999. The MLE algorithm for the matrix normal distribution. *Journal of statistical computation and simulation*, 64(2): 105–123.

Fang, G.; Song, J.; Shen, C.; Wang, X.; Chen, D.; and Song, M. 2019. Data-free adversarial distillation. *arXiv preprint arXiv:1912.11006*.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. *Advances in neural information processing systems*, 27.

Guy, I.; Zwerdling, N.; Ronen, I.; Carmel, D.; and Uziel, E. 2010. Social media recommendation based on people and tags. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, 194–201.

Heo, B.; Lee, M.; Yun, S.; and Choi, J. Y. 2019. Knowledge distillation with adversarial samples supporting decision boundary. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, 3771–3778.

Kang, S.; Hwang, J.; Kweon, W.; and Yu, H. 2020. DE-RRD: A knowledge distillation framework for recommender system. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 605–614.

Kang, W.-C.; and McAuley, J. 2018. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*, 197–206. IEEE.

Kariyappa, S.; Prakash, A.; and Qureshi, M. K. 2021. Maze: Data-free model stealing attack using zeroth-order gradient estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13814–13823.

Krishna, K.; Tomar, G. S.; Parikh, A. P.; Papernot, N.; and Iyyer, M. 2019. Thieves on sesame street! model extraction of bert-based apis. *arXiv preprint arXiv:1910.12366*.

Li, B.; Wang, Y.; Singh, A.; and Vorobeychik, Y. 2016. Data poisoning attacks on factorization-based collaborative filtering. *Advances in neural information processing systems*, 29.

Li, J.; Ren, P.; Chen, Z.; Ren, Z.; Lian, T.; and Ma, J. 2017. Neural attentive session-based recommendation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 1419–1428.

Lin, C.; Chen, S.; Li, H.; Xiao, Y.; Li, L.; and Yang, Q. 2020. Attacking recommender systems with augmented user profiles. In *Proceedings of the 29th ACM international conference on information & knowledge management*, 855–864.

Lin, C.; Chen, S.; Zeng, M.; Zhang, S.; Gao, M.; and Li, H. 2022. Shilling Black-Box Recommender Systems by Learning to Generate Fake User Profiles. *IEEE Transactions on Neural Networks and Learning Systems*.

Liu, F.; Cheng, Z.; Chen, H.; Wei, Y.; Nie, L.; and Kankanhalli, M. 2022. Privacy-preserving synthetic data generation for recommendation systems. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1379–1389.

Micaelli, P.; and Storkey, A. J. 2019. Zero-shot knowledge transfer via adversarial belief matching. *Advances in Neural Information Processing Systems*, 32.

Miura, T.; Shibahara, T.; and Yanai, N. 2024. Megex: Data-free model extraction attack against gradient-based explainable ai. In *Proceedings of the 2nd ACM Workshop on Secure and Trustworthy Deep Learning Systems*, 56–66.

Okura, S.; Tagami, Y.; Ono, S.; and Tajima, A. 2017. Embedding-based news recommendation for millions of users. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, 1933–1942.

Su, J.; Chen, C.; Lin, Z.; Li, X.; Liu, W.; and Zheng, X. 2023a. Personalized behavior-aware transformer for multi-behavior sequential recommendation. In *Proceedings of the 31st ACM International Conference on Multimedia*, 6321–6331.

Su, J.; Chen, C.; Liu, W.; Wu, F.; Zheng, X.; and Lyu, H. 2023b. Enhancing hierarchy-aware graph networks with deep dual clustering for session-based recommendation. In *Proceedings of the ACM Web Conference 2023*, 165–176.

Sutton, R. S.; McAllester, D.; Singh, S.; and Mansour, Y. 1999. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12.

Tang, J.; Wen, H.; and Wang, K. 2020. Revisiting adversarially learned injection attacks against recommender systems. In *Proceedings of the 14th ACM Conference on Recommender Systems*, 318–327.

Wang, C.; Sun, J.; Dong, Z.; Zhu, J.; Li, Z.; Li, R.; and Zhang, R. 2023. Data-free Knowledge Distillation for Reusing Recommendation Models. In *Proceedings of the 17th ACM Conference on Recommender Systems*, 386–395.

Wang, Q.; Yin, H.; Wang, H.; Nguyen, Q. V. H.; Huang, Z.; and Cui, L. 2019. Enhancing collaborative filtering with generative augmentation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 548–556.

Wang, Z.; Zhang, J.; Xu, H.; Chen, X.; Zhang, Y.; Zhao, W. X.; and Wen, J.-R. 2021. Counterfactual data-augmented sequential recommendation. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, 347–356.

Wu, C.; Lian, D.; Ge, Y.; Zhu, Z.; and Chen, E. 2021. Triple adversarial learning for influence based poisoning attack in recommender systems. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 1830–1840.

Yu, L.; Zhang, W.; Wang, J.; and Yu, Y. 2017. Seqgan: Sequence generative adversarial nets with policy gradient. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31.

Yue, W.; Wang, Z.; Zhang, J.; and Liu, X. 2021a. An overview of recommendation techniques and their applications in healthcare. *IEEE/CAA Journal of Automatica Sinica*, 8(4): 701–717.

Yue, Z.; He, Z.; Zeng, H.; and McAuley, J. 2021b. Black-box attacks on sequential recommenders via data-free model extraction. In *Proceedings of the 15th ACM Conference on Recommender Systems*, 44–54.

Zhang, H.; Li, Y.; Ding, B.; and Gao, J. 2020. Practical data poisoning attack against next-item recommendation. In *Proceedings of The Web Conference 2020*, 2458–2464.

Zhang, J.; Chen, C.; and Lyu, L. 2022. IDEAL: Query-Efficient Data-Free Learning from Black-Box Models. In *The Eleventh International Conference on Learning Representations*.

Zhou, M.; Wu, J.; Liu, Y.; Liu, S.; and Zhu, C. 2020. Dast: Data-free substitute training for adversarial attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 234–243.

Zhu, Z.; Fan, R.; Wu, C.; Yang, Y.; Lian, D.; and Chen, E. 2023a. Model Stealing Attack against Recommender System. *arXiv preprint arXiv:2312.11571*.

Zhu, Z.; Wu, C.; Fan, R.; Lian, D.; and Chen, E. 2023b. Membership Inference Attacks Against Sequential Recommender Systems. In *Proceedings of the ACM Web Conference 2023*, 1208–1219.