Practical Adversarial Attacks on Stochastic Bandits via Fake Data Injection

Qirun Zeng[1] **Eric He**[2] **Richard Hoffmann**[2] **Xuchuang Wang**[3] **Jinhang Zuo**[4]
[1]University of Science and Technology of China [2]California Institute of Technology
[3]University of Massachusetts Amherst [4]City University of Hong Kong

Abstract

Adversarial attacks on stochastic bandits have traditionally relied on some unrealistic assumptions, such as per-round reward manipulation and unbounded perturbations, limiting their relevance to real-world systems. We propose a more practical threat model, Fake Data Injection, which reflects realistic adversarial constraints: the attacker can inject only a limited number of bounded fake feedback samples into the learner's history, simulating legitimate interactions. We design efficient attack strategies under this model, explicitly addressing both magnitude constraints (on reward values) and temporal constraints (on when and how often data can be injected). Our theoretical analysis shows that these attacks can mislead both Upper Confidence Bound (UCB) and Thompson Sampling algorithms into selecting a target arm in nearly all rounds while incurring only sublinear attack cost. Experiments on synthetic and real-world datasets validate the effectiveness of our strategies, revealing significant vulnerabilities in widely used stochastic bandit algorithms under practical adversarial scenarios.

1 Introduction

Multi-armed bandit (MAB) algorithms are widely used in online decision-making systems for their ability to balance exploration and exploitation using partial feedback. They form the backbone of many interactive applications, including personalized recommendation [1], online advertising [2], clinical trials [3], and adaptive routing [4]. As these algorithms are increasingly deployed in high-stakes, user-facing systems, growing concerns have emerged regarding their vulnerability to adversarial manipulation. A growing body of work [5, 6, 7, 8] has shown that MAB algorithms are vulnerable to adversarial attacks on feedback, where an attacker subtly perturbs the observed rewards to mislead the learning process. Remarkably, even with limited intervention, the attacker can steer the learner toward repeatedly selecting a targeted but suboptimal arm in the vast majority of rounds.

However, prior works on adversarial attacks against stochastic bandits [5, 6, 8] typically adopt a feedback-perturbation threat model, where the attacker observes the learner's chosen arm and the corresponding environment-generated reward in each round, then arbitrarily modifies that reward before it is revealed to the learner. While this model offers strong theoretical leverage, the assumption that an attacker can directly and continuously alter feedback from the environment is often unrealistic in practice. Consider, for example, a restaurant recommendation platform. An attacker seeking to promote a specific restaurant cannot alter the actual feedback submitted by real users in every round. A more feasible strategy is to create fake user accounts that submit biased reviews to influence the recommendation system. Similarly, in online advertising or click-through rate prediction, attackers commonly engage in click fraud or inject synthetic interactions to simulate user behavior—effectively injecting new data rather than modifying genuine observations. These practical scenarios motivate a shift toward more realistic and constrained threat models.

Preprint.

Building on these observations, we propose a more realistic and practically grounded threat model: the *Fake Data Injection* model. Instead of modifying genuine feedback, the attacker influences the learner indirectly by injecting a limited number of fabricated (arm, reward) pairs into its interaction history. These fake samples must conform to valid feedback ranges (e.g., binary clicks or 1–5 star ratings), and their injection is subject to constraints such as system-level detection or resource limits. The learner processes these fake interactions indistinguishably from real ones—updating estimates, counts, and decision logic accordingly.

This model captures practical attack surfaces overlooked by previous works. It removes the unrealistic assumption of per-round reward manipulation, enables feedback injection on arbitrary arms, and respects the bounded nature of real-world feedback. At the same time, it introduces new algorithmic challenges that cannot be addressed by existing techniques. Unlike the standard model—where the attacker perturbs rewards in real time—fake data injection raises fundamental questions about *when*, *how strongly*, and *how frequently* to inject samples to effectively influence the learner. The attacker must decide: (i) how many fake samples are required to suppress the selections of a non-target arm, (ii) how to achieve this using only bounded reward values, and (iii) how to distribute injections over time when batch size or injection frequency is constrained. These challenges call for new analytical tools and attack strategies that explicitly account for both *magnitude* constraints (on reward values) and *temporal* constraints (on when and how often data can be injected).

Our Contributions. We develop a suite of attack strategies tailored to the fake data injection model, addressing both theoretical and practical challenges:

- We propose the *Least Injection* algorithm for the *unbounded* setting, showing that a single fake sample per non-target arm suffices to steer the learner toward a target arm with sublinear cost. A key technical tool is the *Exponential Suppression Lemma*, which ensures long-term suppression of non-target arms and guides the design of our subsequent algorithms.
- We extend this to the *bounded* setting via the *Simultaneous Bounded Injection (SBI)* algorithm, which replicates the effect of an unbounded sample using a batch of bounded fake data, while maintaining sublinear cost.
- To address stricter constraints, we propose the *Periodic Bounded Injection (PBI)* algorithm, which injects small batches at controlled intervals. We provide a new suppression analysis to guarantee its effectiveness under temporal and magnitude constraints.
- All attack algorithms are analyzed under both UCB and Thompson Sampling, with theoretical guarantees that match the standard threat model in terms of cost and effectiveness.
- We validate our methods on the real-world dataset, demonstrating that even sparse, bounded fake data can significantly bias bandit learners in practice.

Our work bridges the gap between theoretical models of adversarial bandits and practical data-driven attacks observed in real systems. It introduces new threat models, techniques, and insights that we hope will inspire more realistic evaluations of online learning algorithms in adversarial environments.

Related Work. Recent years have seen growing interest in understanding the vulnerability of bandit algorithms under adversarial attacks [5, 6, 9, 10, 11]. Jun et al. [5] initiated this line of work by designing effective attack strategies against UCB and ϵ-greedy algorithms in the stochastic bandit setting. Their work showed that an attacker can steer the learner toward a suboptimal arm while incurring only sublinear cost. Liu and Shroff [6] extended this to settings where the learning algorithm is unknown, and further to contextual bandits. Subsequent work has explored increasingly general and complex settings. Garcelon et al. [9] studied attacks on linear contextual bandits, where an adversary can perturb both the context vectors and rewards. More recently, Zuo [8] developed smoothed attack strategies that reduce detectability while maintaining effectiveness.

While prior work has shown that bandit algorithms can be manipulated with sublinear cost, most of these attacks rely on a strong feedback-perturbation threat model, where the attacker modifies the observed reward of the selected arm in every round—often with unbounded perturbations [5, 6, 8]. Even in more recent studies that consider bounded feedback [12, 13, 7], the attacker is still allowed to act continuously, and only needs to decide *when* to attack. In contrast, we introduce the *Fake Data Injection Threat Model*, where the attacker indirectly influences the learner by inserting fabricated (arm, reward) pairs into its history. This model captures realistic constraints found in systems like recommendation platforms, where attackers can create fake users but cannot modify real feedback. It

imposes bounded rewards, supports injection on arbitrary arms, and introduces constraints on both the number and frequency of injections, necessitating new algorithmic techniques.

2 Preliminaries

2.1 Stochastic Bandits

We consider the standard stochastic multi-armed bandit setting with the arm set $[K] \coloneqq \{1, 2, \ldots, K\}$, where each arm $k \in [K]$ is associated with an unknown reward distribution with mean μ_k. The reward distributions are assumed to be σ^2-sub-Gaussian, with σ^2 known. Without loss of generality, we assume the arms are ordered such that $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_K$. In each round $t = 1, 2, \ldots, T$, the learner selects an arm $a_t \in [K]$ and receives a stochastic reward r_t drawn from the distribution corresponding to arm a_t. In this paper, we consider two widely used algorithms for stochastic bandits: Upper Confidence Bound (UCB) and Thompson Sampling (TS).

Upper Confidence Bound (UCB). We consider the UCB algorithm as specified in Jun et al. [5], Zuo [8], and the prototype is the (α, ψ) algorithm of Bubeck and Cesa-Bianchi [14, Section 2.2]. In the first K rounds, the learner pulls each arm once to initialize reward estimates. For subsequent rounds $t > K$, the learner selects the arm with the highest UCB index $a_t = \arg\max_{a \in [K]} \left[\hat{\mu}_a(t) + 3\sigma\sqrt{\frac{\log t}{N_a(t)}}\right]$, where $\hat{\mu}_a(t)$ is the empirical mean reward of arm a and $N_a(t)$ is the number of times arm a has been selected up to round t.

Thompson Sampling (TS). We consider the Thompson Sampling algorithm specified in Zuo [8], and the prototype is the (α, ψ) algorithm of Agrawal and Goyal [15]. In the first K rounds, the learner pulls each arm once. For rounds $t > K$, a sample ν_a is drawn independently for each arm $a \in [K]$ from the distribution $\mathcal{N}(\hat{\mu}_a(t), 1/N_a(t))$, and the learner selects the arm with the largest sampled value $a_t = \arg\max_{a \in [K]} \nu_a$. In the absence of attacks, both UCB and TS are known to achieve sublinear regret by selecting suboptimal arms only $o(T)$ times.

2.2 Previous Threat Model and Limitations

We begin by reviewing the standard threat model adopted in prior works on adversarial attacks against bandit algorithms [5, 6, 7, 8]. In each round t, the learner selects an arm a_t to play, and the environment generates a pre-attack reward r_t^0 drawn from the underlying distribution of arm a_t. The attacker then observes the tuple (a_t, r_t^0) and decides an attack value α_t. The learner can only receive the post-attack reward $r_t = r_t^0 - \alpha_t$. Define the cumulative attack cost as $C(T) = \sum_{t=1}^{T} |\alpha_t|$. The attacker's objective is to manipulate the learner into selecting a specific target arm for a linear number of rounds while incurring only sublinear attack cost. Formally, an attack is considered successful if the attacker can force the learner to select the target arm $T - o(T)$ times while ensuring that $C(T) = o(T)$. While many attack strategies have been proposed under the standard threat model, it exhibits several critical limitations when applied to practical settings.

First, the model assumes that the attacker can perturb the environment-generated reward in *every* round. This assumption is often unrealistic in real-world applications such as recommender systems. For example, consider an app that recommends restaurants to users and collects their feedback to improve future recommendations. An attacker may wish to bias the system toward recommending a particular restaurant, but it is infeasible to directly modify the feedback of all real users. In practice, a more common attack strategy is to create fake users who submit fabricated feedback. However, even this is constrained by operational or detection limits—adding fake users in every round is highly impractical. Thus, the assumption of per-round attack capability does not reflect realistic adversarial power. *Second*, the model restricts the attacker to modifying only the reward of the chosen arm in each round. In contrast, fake-user-based attacks offer more flexibility. A fake user can submit feedback on any item (i.e., any arm), regardless of what the learner selected in that round. This means fake data injection enables the attacker to fabricate feedback for arbitrary arms, not just the one currently played by the learner. As a result, the standard model underestimates the attacker's flexibility in practice and overestimates their ability to act at every timestep. *Third*, the threat model assumes that both the pre-attack reward and the attack values are unbounded. In many systems, user feedback is naturally bounded — e.g., binary click signals or discrete rating scores (e.g., 1 to 5 stars). Allowing arbitrarily large attack values could result in out-of-range or clearly invalid feedback, which

would either be filtered out by the system or easily flagged as suspicious. Therefore, attacks that rely on large reward perturbations are incompatible with these bounded-feedback environments.

3 New Threat Model: Fake Data Injection

To address practical limitations of the standard adversarial attack model, we introduce a new and more realistic threat model, which we call the Fake Data Injection Threat Model. This model captures how adversaries behave in real-world systems such as recommendation platforms, where direct manipulation of genuine user feedback is infeasible, and attacks are often carried out by injecting fabricated interactions (e.g., fake users with fake feedback).

In the Fake Data Injection model, the attacker does *not* interfere with the feedback received by the learner during normal interactions. Instead, the attacker is allowed to inject up to N^F *fake data samples*, denoted by $\{(a_i^F, r_i^F)\}_{i=1}^{N^F}$, into the learner's history. Each fake data point mimics a legitimate user interaction, where $a_i^F \in [K]$ is the selected arm and $r_i^F \in [\tilde{a}, \tilde{b}]$ is the corresponding reward for two given bounds $\tilde{a} \leq \tilde{b}$. The learner processes these injected samples in the same way as genuine observations—for example, updating the empirical mean $\hat{\mu}_{a_i^F}$, incrementing the pull count $N_{a_i^F}$, and advancing the internal time step t. We define the total attack cost as:

$$C^F(T) \coloneqq \sum_{i=1}^{N^F} |r_i^F - \mu_{a_i^F}|,$$

and consider an attack *successful* if it can mislead the learner into pulling a target arm for $T - o(T)$ rounds while ensuring $C^F(T) = o(T)$, analogous to prior work.

This new model resolves several key limitations of the previous threat model:

Limited access manipulation. Unlike the standard model—which assumes the attacker can modify the reward in every round—our model reflects the more plausible scenario where the attacker can only inject a limited number of fake interactions. For instance, in a restaurant recommendation app, an attacker cannot tamper with the feedback from real users but can register a finite number of fake accounts to submit biased reviews. It is unrealistic to assume the attacker can do this in every round without detection or resource exhaustion.

Flexible feedback across arms. The standard model restricts the attacker to modifying the reward for the arm chosen by the learner. In contrast, our model allows the attacker to fabricate data for *any* arm. This mirrors real-world attacks where fake users can submit reviews or feedback on arbitrary items, not just those recommended to them. For example, an attacker aiming to boost a target restaurant can flood the system with positive feedback for that restaurant—regardless of whether it was actually recommended in a specific round.

Bounded and plausible feedback. In our model, the fake rewards must lie within the valid feedback range $[\tilde{a}, \tilde{b}]$, consistent with many practical systems that collect binary clicks or scaled ratings (e.g., 1 to 5 stars). This avoids the unrealistic assumption of unbounded reward modifications, where a single large perturbation could dominate the learner's behavior. Our bounded injection design ensures the fake data remains indistinguishable from legitimate interactions.

4 Attack Strategies

In this section, we develop attack strategies specifically designed for the fake data injection threat model. We begin by studying a simplified setting with unbounded feedback, which serves as a conceptual bridge from prior threat models: instead of directly altering the learner's observed rewards, the attacker injects fake data with unrestricted values. It allows us to highlight the core mechanisms and intuitions behind effective attack strategies. Building on these, we then consider constrained injection attacks, where both the *magnitude* of fake feedback and the *injection frequency* are limited. Our goal is to design strategies that balance these two constraints while still steering the learner toward suboptimal behavior efficiently. Without loss of generality, we assume that the target arm is arm K which has the lowest expected reward.[1]

[1]This represents the most challenging case for the attacker and can be easily extended to target any other arm.

4.1 Warm-up: Injection Attacks with Unbounded Feedback

We begin our study of fake data injection attacks by considering a relaxed setting in which the injected reward values r_i^F can take *arbitrary real values*, i.e., the bounded feedback constraint from Section 3 is removed. This setting closely mirrors the standard threat model, where the attacker can directly modify the observed reward as $r_t = r_t^0 - \alpha_t$, allowing unbounded perturbations α_t in each round. In this unbounded injection setting, we demonstrate that *injecting a single fake data point per non-target arm* is sufficient to mislead the learner into favoring the target arm. We formalize this insight through the Least Injection Algorithm, a simple yet effective one-shot attack strategy against the UCB algorithm, as shown in Algorithm 1.

Algorithm 1: Least Injection Algorithm on UCB

Input: Attack parameter $\delta_0 > 0$

for *round* $t = 1, 2, \dots$ **do**
  for *each non-target arm* $i \in [K-1]$ **do**
    if *arm* i *has not been attacked* ***and*** $N_i(t) = \lceil (\log T)/\delta_0^2 \rceil$ **then**
      Inject fake data sample:
      $(a_i^F, r_i^F) = \left(i, N_i(t) \cdot \left(\hat{\ell}_K(t) - \hat{\mu}_i(t)\right) + \hat{\ell}_K(t)\right)$
      $t \leftarrow t + 1$
    end
  end
end

The attack operates as follows. For each non-target arm i, we wait until it has been pulled $N_i(t) = \lceil (\log T)/\delta_0^2 \rceil$ times. At this point, we inject a single fake sample designed to reduce its empirical mean below a high-probability lower bound of the target arm. Specifically, we define the empirical lower confidence bound for target arm K as:

$$\hat{\ell}_K(t) \coloneqq \hat{\mu}_K(t) - 2\beta(N_K(t)) - 3\sigma\delta_0,$$

where $\beta(N) \coloneqq \sqrt{\frac{1}{2N} \log\left(\frac{\pi^2 K N^2}{3\delta}\right)}$, and $\delta_0 > 0$ is a tunable attack parameter. The injected sample on non-target arm i ensures that after the attack, the empirical mean of arm i satisfies:

$$\hat{\mu}_i(t+1) \leq \hat{\ell}_K(t), \tag{1}$$

thus making it unlikely to be selected in future rounds. The total number of injected fake data points is at most $K-1$, one per non-target arm. Define $\Delta_i := \mu_i - \mu_K$. We now provide the formal theoretical guarantee of Algorithm 1.

Theorem 4.1. *Suppose $T > 2K, \delta < 0.5$. With probability at least $1-\delta$, Algorithm 1 forces the UCB algorithm to select the target arm in at least*

$$T - \mathcal{O}\left((K-1)(\log T)/\delta_0^2\right)$$

rounds, using a cumulative attack cost of at most

$$C^F(T) = \sum_{i=1}^{K-1} |r_i^F - \mu_i| \leq \mathcal{O}\left(\sum_{i=1}^{K-1} (\Delta_i + 4\beta(1) + 3\sigma\delta_0) \cdot \frac{\log T}{\delta_0^2}\right).$$

Compared with the attack algorithm under the standard threat model in [5], the Least Injection Algorithm achieves a similar level of target-arm selection with comparable sublinear attack cost. Notably, the parameter δ_0 controls the trade-off between the number of non-target arm pulls and the attack cost: increasing δ_0 reduces the number of non-target pulls but increases the cost per injection. However, the marginal benefit diminishes once $\delta_0 > \sqrt{\log T}$, beyond which the cost grows without improving effectiveness. By selecting $\delta_0 = \Theta(\sqrt{\log T})$, the cumulative attack cost is minimized to $\widehat{\mathcal{O}}(K\sigma\sqrt{\log T})$, which matches the lower bound $\Omega(\sqrt{\log T})$ established in [8].

To prove Theorem 4.1, we introduce the following lemma, which plays a central role in our attack design and serves as a key building block for subsequent algorithms.

Lemma 4.1 (Exponential Suppression of Non-Target Arms)**.** *Suppose $T > 2K, \delta < 0.5$. With probability at least $1-\delta$, for any non-target arm $i \in [K-1]$ that has been pulled $N_i(t)$ times, if a fake data point is injected according to Line 5 of Algorithm 1, then arm i will not be selected again until at least round* $\exp(N_i(t)\delta_0^2)$.

Proof Sketch. After the injection, the empirical mean of arm i is reduced such that its UCB index becomes significantly lower than that of the target arm. We analyze the evolution of the UCB indices and show that, unless arm i is pulled again (which it is not), its confidence bound tightens slowly while its empirical mean remains suppressed. By induction over subsequent rounds, we show that the UCB index of arm i remains lower than that of the target arm for an exponential number of rounds, specifically up to round $\exp(N_i(t)\delta_0^2)$. □

Remark. *Lemma 4.1 establishes a critical property of our attack strategy: once a non-target arm i has been pulled sufficiently and a properly chosen fake data point is injected, its UCB index becomes exponentially suppressed. More precisely, if the following two conditions are satisfied (1) $N_i(t) \geq (\log T)/\delta_0^2$ and (2) $\hat{\mu}_i(t+1) \leq \hat{\ell}_K(t)$, then arm i will not be selected again until after round T. This suppression effect is crucial: it guarantees that once the attack is applied to arm i, its influence on the learning process becomes negligible for the remaining rounds. The attacker can thus prevent further exploration of non-target arms using only a single injection per arm, ensuring that the learner increasingly concentrates on the target arm. This mechanism forms the backbone of all our attack strategies.*

In addition to attacking UCB, we extend the Least Injection Algorithm to target the Thompson Sampling algorithm. Specifically, the attacker injects a single fake data point into each non-target arm i when $N_i(t) = \lceil \log T/\delta_0^2 \rceil$, using a modified version of Line 5 in Algorithm 1. Due to space limitations, we defer the full algorithm and details to the appendix. We provide its theoretical guarantee below.

Theorem 4.2. *Suppose $T > 2K, \delta < 0.5$. With probability at least $1-2\delta$, the modified Least Injection Algorithm forces the Thompson Sampling algorithm to select the target arm in at least $T - \mathcal{O}((K-1)\log T/\delta_0^2)$ rounds, using a cumulative attack cost of at most* $\mathcal{O}\left(\sum_{i=1}^{K}\left(\Delta_i + 4\beta(1) + \sqrt{8\log\left(\frac{\pi^2 K}{3\delta}\right)} + 4\sqrt{\log T}\right)\frac{\log T}{\delta_0^2}\right)$.

Compared with the attack algorithm under the standard threat model studied in [8], our approach achieves a similar level of target-arm selection with matching attack cost. By setting the attack parameter $\delta_0 = \Theta(\sqrt{\log T})$, we obtain a total cost of $\mathcal{O}(\sqrt{\log T})$, which aligns with the known lower bound, demonstrating the near-optimality of our strategy under this relaxed fake data injection model.

4.2 Constrained Injection Attacks

We now turn to more realistic and constrained settings where injected fake data must lie within a bounded range. This reflects practical scenarios in which user feedback—such as clicks or ratings—is inherently limited (e.g., binary or on a fixed scale). To address the constraint on individual fake rewards, we first propose a natural extension of Algorithm 1. In this version, the influence of a single unbounded fake reward is approximated by injecting *a batch of bounded fake samples* simultaneously for each non-target arm.

In practice, however, attackers may face an additional constraint on the number of fake samples that can be injected at any given time, due to resource limitations or system-level detection thresholds. To address this more challenging setting, we introduce a periodic injection strategy that operates under two constraints: (1) each fake reward must be bounded; (2) only a limited number of fake samples can be injected at once. Our strategy carefully coordinates the frequency and timing of injections to maintain effective adversarial influence over the learner while satisfying both constraints. Despite these limitations, we show that the attacker can still successfully manipulate the learner's behavior with sublinear cost.

4.2.1 Simultaneous Bounded Injection

To ensure the attack remains realistic and stealthy, we consider a practical setting in which each injected fake reward must lie within a bounded range $[\tilde{a}, \tilde{b}]$. Under this constraint, we propose the

Simultaneous Bounded Injection (SBI) algorithm, which extends Algorithm 1 by replicating the effect of a single unbounded fake reward via the injection of multiple bounded fake samples. Specifically, for each non-target arm, the attacker injects a batch of rewards with the minimum value $\tilde{a}$ in a single round to achieve the same suppression of the empirical mean as in the unbounded setting.

Algorithm 2: Simultaneous Bounded Injection on UCB

Input: Attack parameter δ_0, bounded reward range $[\tilde{a}, \tilde{b}]$

for *round* $t = 1, 2, \ldots$ **do**
  for *each non-target arm* $i \in [K-1]$ **do**
    if *arm* i *has not been attacked* ***and*** $N_i(t) = \lceil (\log T)/\delta_0^2 \rceil$ **then**
      $\tilde{n} \leftarrow \left\lceil \frac{\hat{\mu}_i(t) - \hat{\ell}_K(t)}{\hat{\ell}_K(t) - \tilde{a}} \cdot \left\lceil \frac{\log T}{\delta_0^2} \right\rceil \right\rceil$;
      Inject $\tilde{n}$ fake samples $(i, \tilde{a})$;
      $t \leftarrow t + \tilde{n}$
    end
  end
end

As shown in Algorithm 2, once a non-target arm i has been pulled $\lceil (\log T)/\delta_0^2 \rceil$ times, the attacker computes $\tilde{n}$ and injects $\tilde{n}$ fake samples with reward $\tilde{a}$. We make the following assumption to ensure that the suppression of arm i's empirical mean is always feasible:

Assumption 4.1. $\tilde{a} \leq \mu_K - 3\beta(1) - 3\sigma\delta_0$.

This assumption guarantees that it is always possible to reduce the empirical means of non-target arms below the lower confidence bound of the target arm. In practice, it can be relaxed to $\tilde{a} < \hat{\mu}_K(t) - 2\beta(N_K(t)) - 3\sigma\delta_0$ at the specific round t; we state the worst-case condition for generality.

We now present the theoretical guarantee of the SBI algorithm.

Theorem 4.3. *Suppose $T > 2K, \delta < 0.5$ and Assumption 4.1 hold. With probability at least $1 - \delta$, Algorithm 2 forces the UCB algorithm to select the target arm in at least*

$$T - \mathcal{O}\left(\sum_{i=1}^{K-1} \frac{\mu_i + \beta\left((\log T)/\delta_0^2\right) - \tilde{a}}{\mu_K - 3\beta(1) - 3\sigma\delta_0 - \tilde{a}} \cdot \frac{\log T}{\delta_0^2} \right)$$

rounds, using a cumulative attack cost of at most

$$O\left(\sum_{i=1}^{K-1} (\mu_i - \tilde{a}) \frac{\Delta_i + 4\beta(1) + 3\sigma\delta_0}{\mu_K - 3\beta(1) - 3\sigma\delta_0 - \tilde{a}} \frac{\log T}{\delta_0^2} \right),$$

Compared with Theorem 4.1, the attacker now injects $\tilde{n} = \left\lceil \frac{\hat{\mu}_i(t) - \hat{\ell}_K(t)}{\hat{\ell}_K(t) - \tilde{a}} \cdot \left\lceil \frac{\log T}{\delta_0^2} \right\rceil \right\rceil$ fake samples for each non-target arm instead of a single one. While this increases the total number of injected samples, the order of the attack cost remains $\mathcal{O}(\sqrt{\log T})$ when the attack parameter is set to $\delta_0 = \Theta(\sqrt{\log T})$. Thus, the SBI algorithm maintains asymptotic optimality under more realistic constraints.

As in Section 4.1, the SBI algorithm can also be extended to attack the Thompson Sampling algorithm. Due to space constraints, we defer the full algorithm and details to the appendix and present the main theorem below.

Theorem 4.4. *Suppose $T > 2K, \delta < 0.5$. With probability at least $1 - 2\delta$, the modified Simultaneous Bounded Injection forces the Thompson sampling algorithm to select the target arm in at least*

$$T - \mathcal{O}\left(\sum_{i=0}^{K-1} \frac{\mu_K - 3\beta(1) - \sqrt{8\log(\pi^2 K/(3\delta))} - 4\sqrt{\log T} - \tilde{a}}{\mu_i + \beta(\log T/\delta_0^2) - \tilde{a}} \frac{\log T}{\delta_0^2} \right) \tag{2}$$

with the attack cost:

$$\mathcal{O}\left(\sum_{i=1}^{K-1} (\mu_i - \tilde{a}) \frac{\Delta_i + 4\beta(1) + \sqrt{8\log(\pi^2 K/(3\delta))} + 4\sqrt{\log T}}{\mu_K - 3\beta(1) - \sqrt{8\log(\pi^2 K/(3\delta))} - 4\sqrt{\log T} - \tilde{a}} \frac{\log T}{\delta_0^2} \right) \tag{3}$$

4.2.2 Periodic Bounded Injection

The SBI algorithm above assumes that the attacker can inject all required fake samples within a single round. However, this assumption may not hold in practice. For example, in a restaurant recommendation system, injecting a large batch of fake (e.g., low-rating) reviews at once may trigger anomaly detection mechanisms, leading the system to filter or ignore the fake data. In contrast, injecting smaller amounts of fake feedback periodically—at a controlled rate—can be significantly less suspicious and more effective in practice.

To model this scenario, we introduce a more restrictive and realistic setting where:

1. The attacker can inject at most f fake samples in any single round (batch size constraint);
2. There must be a delay of at least R_i rounds between consecutive injections on the same arm i (cooldown constraint).

To address this setting, we propose the Periodic Bounded Injection (PBI) algorithm, shown in Algorithm 3. Given a maximum batch size f, the algorithm adaptively schedules periodic injections to suppress the empirical mean of non-target arms while respecting both constraints.

Algorithm 3: Periodic Bounded Injection on UCB

Input: Attack parameter δ_0, reward bound $[\tilde{a}, \tilde{b}]$, max batch size f

for *round* $t = 1, 2, \ldots$ **do**
  for *each non-target arm* $i \in [K-1]$ **do**
    if *arm* i *has not been attacked* ***and*** $N_i(t) = \lceil (\log T)/\delta_0^2 \rceil$ **then**
      $\tilde{n}_i \leftarrow \left\lceil \frac{\hat{\mu}_i(t) - \hat{\ell}_K(t)}{\hat{\ell}_K(t) - \tilde{a}} \cdot \left\lceil \frac{\log T}{\delta_0^2} \right\rceil \right\rceil$;
      $R_i \leftarrow \min_{1 \le c \le \lceil \frac{\tilde{n}_i}{f} \rceil} \frac{1}{c} \exp\left(\left(\frac{\hat{\mu}_K(t) - 2\beta(N_K(t)) - \tilde{\mu}_i(t_i(c))}{3\sigma} \right)^2 \cdot (N_i(t) + fc) \right) - t - f$;
      // $\tilde{\mu}_i(t_i(c))$ represents the estimated cumulative empirical mean of arm i after the i-th injection
      next$_i \leftarrow t$;
    end
    if $\tilde{n}_i > 0$ ***and*** *next*$_i \le t$ **then**
      Inject f fake samples $(i, \tilde{a})$;
      $\tilde{n}_i \leftarrow \tilde{n}_i - f$;
      $t \leftarrow t + f$;
      next$_i \leftarrow$ next$_i + f + R_i$;
    end
  end
end

The PBI algorithm distributes the injection of fake samples across multiple rounds rather than injecting them all at once. Once a non-target arm i reaches the designated pull threshold ($\lceil (\log T)/\delta_0^2 \rceil$), the attacker computes both the total number of fake samples $\tilde{n}_i$ required to suppress the empirical mean of arm i, and a waiting interval R_i, which ensures that the fake samples can be injected periodically without allowing arm i to regain a high UCB index. The notation $\tilde{\mu}_i(t+f)$ represents the estimated value of $\hat{\mu}_i(t+f)$ with f fake data injections starting from round t. At each interval of $R_i + f$ rounds, a batch of f fake samples is injected until the total $\tilde{n}_i$ is exhausted. This strategy effectively balances *stealthiness* and *attack efficacy*, making it robust against detection in practical systems with bounded feedback and rate-limited injection constraints.

The analysis of cumulative attack cost for PBI is deferred to the appendix, as it is similar to that of Theorem 4.3: the total number of fake samples injected remains the same. What distinguishes PBI is how *suppression is maintained across time*, which is guaranteed by the following lemma.

Lemma 4.2. *The choice of R_i in Algorithm 3 ensures that once a batch of f fake data samples is injected into non-target arm i, the arm will not be selected again for at least the next R_i rounds.*

Proof Sketch. This result builds on a modified version of the exponential suppression lemma (Lemma 4.1). Rather than suppressing a non-target arm with a single large injection, we ana-

lyze the suppression effect of a partial injection of f bounded fake samples. We show that the first batch induces the weakest suppression, so it suffices to compute R_i based on this worst-case scenario. By ensuring that the UCB index remains below that of the target arm during this interval, we guarantee that arm i is not selected within the next R_i rounds. After the c-th period of injection, we have $\hat{\mu}_i \leq \hat{\mu}_K - 2\beta(N_K(t)) - 3\sigma\sqrt{(\log(t+(f+R)c))/N_i(t+(f+R)c)}$. And its UCB index will remain lower than arm K's until at least round $t+(f+R)c$. □

We also extended the PBI algorithm to attack the Thompson Sampling algorithm. Due to space limitations, we defer the detailed algorithm and corresponding results to the appendix.

5 Experiments

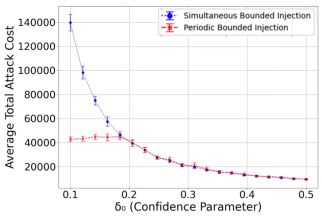

Figure 1: Attack Costs vs δ_0

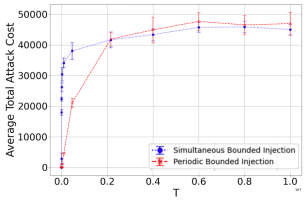

Figure 2: Attack Costs vs T

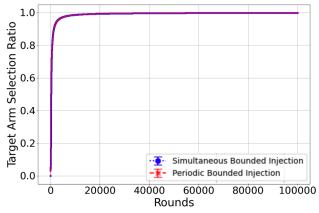

Figure 3: Target Arm Selection Ratios

We evaluate our attack strategies in a realistic setting using the MovieLens 25M dataset [16], which reflects the practical motivations of the Fake Data Injection model. Due to space constraints, we report results for the SBI and PBI algorithms on the UCB learner; results for other settings are moved to the appendix. We consider $K = 10$ arms and simulate user interaction traces with stochastic rewards derived from movie rating distributions. The time horizon is set to $T = 100{,}000$. For PBI, the per-round injection limit is set to $f = 5$. We vary the confidence parameter δ_0 and the horizon T to evaluate their effect on attack cost and effectiveness. Figure 1 plots the average attack cost as a function of δ_0. As expected, increasing δ_0 reduces the number of required fake samples for suppressing non-target arms, leading to lower attack costs. PBI consistently incurs lower cost than SBI for small δ_0 due to its more conservative and distributed injection schedule. Figure 2 shows the total attack cost versus the time horizon T. When T is small, some arms are not pulled often enough to trigger full injection, resulting in a lower realized cost. As T increases, the cost gradually converges to the predicted values. Across all settings, PBI performs comparably or better than SBI. Figure 3 tracks the target arm selection ratio over time. Both SBI and PBI are highly effective, with the learner converging to the target arm in nearly all rounds after early exploration. These results confirm that fake data injection attacks remain highly effective under realistic constraints. In particular, the PBI strategy achieves strong empirical performance while often incurring lower attack cost than SBI. This underscores the practical advantage of temporally distributed attacks.

6 Concluding Remarks

This work introduces a practical and realistic threat model, Fake Data Injection, for adversarial attacks on stochastic bandits. In contrast to prior models that assume per-round, unbounded reward perturbations, our framework captures real-world constraints such as bounded feedback, limited injection capability, and the attacker's inability to modify genuine user data. Within this model, we develop a suite of effective attack strategies that successfully manipulate both UCB and TS algorithms using only sublinear-cost injections. Our theoretical analysis and experimental results demonstrate that even sparse, bounded fake interactions can significantly bias stochastic bandit algorithms.

Despite these results, several limitations remain and open avenues for future work. We assume a passive learner that processes all feedback without defense, which may not hold in robust or adversarial-aware systems. Our work also focuses on stochastic bandits; extending to contextual or reinforcement learning settings remains an open challenge. Additionally, real-world attackers may face detection risks or adaptive filtering by the system—scenarios not captured in our current framework.

Future work should explore defense mechanisms such as anomaly detection or arm-level auditing, and investigate the dynamic interplay between attackers and adaptive learners. Addressing these limitations will be crucial for building secure online learning systems in adversarial environments.

References

[1] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670, 2010.

[2] Wei Chen, Yajun Wang, Yang Yuan, and Qinshi Wang. Combinatorial multi-armed bandit and its extension to probabilistically triggered arms. *Journal of Machine Learning Research*, 17 (50):1–33, 2016.

[3] Sofía S Villar, Jack Bowden, and James Wason. Multi-armed bandit models for the optimal design of clinical trials: benefits and challenges. *Statistical science: a review journal of the Institute of Mathematical Statistics*, 30(2):199, 2015.

[4] Shuai Li, Baoxiang Wang, Shengyu Zhang, and Wei Chen. Contextual combinatorial cascading bandits. In *International conference on machine learning*, pages 1245–1253. PMLR, 2016.

[5] Kwang-Sung Jun, Lihong Li, Yuzhe Ma, and Jerry Zhu. Adversarial attacks on stochastic bandits. *Advances in neural information processing systems*, 31, 2018.

[6] Fang Liu and Ness Shroff. Data poisoning attacks on stochastic bandits. In *International Conference on Machine Learning*, pages 4042–4050. PMLR, 2019.

[7] Jinhang Zuo, Zhiyao Zhang, Zhiyong Wang, Shuai Li, Mohammad Hajiesmaili, and Adam Wierman. Adversarial attacks on online learning to rank with click feedback. *Advances in Neural Information Processing Systems*, 36:41675–41692, 2023.

[8] Shiliang Zuo. Near optimal adversarial attacks on stochastic bandits and defenses with smoothed responses. In *International Conference on Artificial Intelligence and Statistics*, pages 2098–2106. PMLR, 2024.

[9] Evrard Garcelon, Baptiste Roziere, Laurent Meunier, Jean Tarbouriech, Olivier Teytaud, Alessandro Lazaric, and Matteo Pirotta. Adversarial attacks on linear contextual bandits. *Advances in Neural Information Processing Systems*, 33:14362–14373, 2020.

[10] Yuzhe Ma and Zhijin Zhou. Adversarial attacks on adversarial bandits. *arXiv preprint arXiv:2301.12595*, 2023.

[11] Huazheng Wang, Haifeng Xu, and Hongning Wang. When are linear stochastic bandits attackable? In *International Conference on Machine Learning*, pages 23254–23273. PMLR, 2022.

[12] Yinglun Xu, Bhuvesh Kumar, and Jacob D Abernethy. Observation-free attacks on stochastic bandits. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 22550–22561. Curran Associates, Inc., 2021. URL `https://proceedings.neurips.cc/paper_files/paper/2021/file/be315e7f05e9f13629031915fe87ad44-Paper.pdf`.

[13] Zichen Wang, Rishab Balasubramanian, Hui Yuan, Mengdi Wang, Huazheng Wang, et al. Adversarial attacks on online learning to rank with stochastic click models. *Transactions on Machine Learning Research*, 2024.

[14] Sébastien Bubeck and Nicolò Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems, 2012. URL `https://arxiv.org/abs/1204.5721`.

[15] Shipra Agrawal and Navin Goyal. Near-optimal regret bounds for thompson sampling. *J. ACM*, 64(5), September 2017. ISSN 0004-5411. doi: 10.1145/3088510. URL `https://doi.org/10.1145/3088510`.

[16] F. Maxwell Harper and Joseph A. Konstan. The movielens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 5(4):1–19, 2015. doi: 10.1145/2827872. URL `https://doi.org/10.1145/2827872`.

Appendix

A Proofs

A.1 Concentration Results

Suppose that the reward distributions of arms are σ^2-sub-Gaussian. The following concentration result will be useful throughout our analysis.

Recall that

$$\beta(N) = \sqrt{\frac{2\sigma^2}{N} \log \frac{\pi^2 K N^2}{3\delta}}. \tag{4}$$

Then we define event $\mathcal{E}$ as:

$$\mathcal{E} \coloneqq \{\forall i, t, |\hat{\mu}_i(t) - \mu_i| < \beta(N_i(t))\}, \tag{5}$$

where $\hat{\mu}_i(t)$ is the empirical mean reward of arm i and $N_i(t)$ is the number of times arm i has been selected up to round t.

Lemma A.1 (Lemma 1 in [5]). *For $\delta \in (0,1)$, $\mathbb{P}\{\mathcal{E}\} > 1 - \delta$.*

We then define another event $\mathcal{F}$ to bound the sampled value ν_i of arm i in Thompson sampling

$$\mathcal{F} := \left\{\forall i, t, |\nu_i(t) - \hat{\mu}_i(t)| < \frac{\gamma(t)}{N_i(t)}\right\}, \tag{6}$$

where $\gamma(t) \coloneqq \sqrt{2 \log \frac{\pi^2 K t^2}{3\delta}}$.

Lemma A.2 (Lemma 3 in [8]). *For $\delta \in (0,1)$, $\mathbb{P}\{\mathcal{F}\} > 1 - \delta$.*

A.2 Proofs of Injection Attacks with Unbounded Feedback

This section details the proof of Theorem 4.1, Theorem 4.2. First, we present the proof of Lemma 4.1.

A.2.1 Proof of Lemma 4.1

Proof. Under event $\mathcal{E}$ (defined in Equation (5)), since Algorithm 1 never attacks the target arm K, we can establish a lower bound on its estimate. For any two rounds $t_2 > t_1$,

$$\hat{\mu}_K(t_2) \geq \mu_K - \beta(N_K(t_2)) \geq \mu_K - \beta(N_K(t_1)) \geq \hat{\mu}_K(t_1) - 2\beta(N_K(t_1)), \tag{7}$$

where the second inequality follows from the monotonicity of $\beta(N)$.

Suppose that at round t_1, Algorithm 1 injects fake feedback on arm i such that

$$\hat{\mu}_i(t_1) \leq \hat{\mu}_K(t_1) - 2\beta(N_K(t_1)) - 3\sigma\delta_0. \tag{8}$$

This guarantees that in round $t_1 + 1$, the UCB index of arm i satisfies

$$\text{UCB}_i(t_1 + 1) < \text{UCB}_K(t_1 + 1),$$

so arm i is not selected at round $t_1 + 1$.

Now consider any subsequent round t_2 with $t_1 < t_2 < \exp(n_i\delta_0^2)$, where $n_i \coloneqq N_i(t_1)$, and assume that arm i has not been selected in any round between t_1 and t_2. Then $N_i(t_2 + 1) = n_i$ and $\hat{\mu}_i(t_2 + 1) = \hat{\mu}_i(t_1)$, so the UCB index for arm i at round $t_2 + 1$ is

$$\begin{aligned}
\text{UCB}_i(t_2 + 1) &= \hat{\mu}_i(t_1) + 3\sigma\sqrt{\frac{\log(t_2 + 1)}{n_i}} \\
&\leq \hat{\mu}_K(t_1) - 2\beta(N_K(t_1)) - 3\sigma\delta_0 + 3\sigma\sqrt{\frac{\log(t_2 + 1)}{n_i}} \\
&\leq \hat{\mu}_K(t_1) - 2\beta(N_K(t_1)) \quad (\text{since } t_2 < \exp(n_i\delta_0^2)) \\
&\leq \hat{\mu}_K(t_2 + 1) \quad (\text{by (7)}) \\
&\leq \text{UCB}_K(t_2 + 1),
\end{aligned}$$

where the third step uses the bound $\sqrt{\log \frac{t_2+1}{n_i}} < \delta_0$. This argument shows that the UCB index of arm i remains strictly lower than that of the target arm K for all $t_2 < \exp(n_i\delta_0^2)$. By induction, arm i will not be selected again until at least $\exp(n_i\delta_0^2)$ rounds have passed. □

As a direct corollary of Lemma 4.1, if arm i satisfies $\hat{\mu}_i(t) \le \hat{\ell}_K(t)$ for any round t, and has been pulled more than $\frac{\log T}{\delta_0^2}$ times, then arm i will not be selected again before round T.

Meanwhile, we present an analogous result for Thompson sampling in Lemma A.3, which admits a similar corollary under the same condition. For simplicity, we define $\hat{\ell}'_K(t) = \hat{\mu}_K(t) - 2\beta(N_K(t)) - \sqrt{8\log\left(\frac{\pi^2 K}{3\delta}\right)} - 4\sqrt{N_i(t)}\delta_0$

Lemma A.3. *For each non-target arm $i \in [K-1]$, if $\hat{\mu}_i(t) \le \hat{\ell}'_K(t)$, then with probability at least $1-2\delta$, arm i will not be selected again until at least round $\lfloor \exp(N_i(t)\delta_0^2) \rfloor$.*

Proof. Suppose that at round t_1, the following inequality holds:

$$\hat{\mu}_i(t_1) \le \hat{\mu}_K(t_1) - 2\beta(N_K(t_1)) - \left(\sqrt{8\log\frac{\pi^2 K}{3\delta}} + 4\sqrt{N_i(t_1)}\,\delta_0\right). \tag{9}$$

Let $n_i \coloneqq N_i(t_1)$ and consider any round t_2 such that $t_1 < t_2 < \lfloor \exp(n_i\delta_0^2) \rfloor$. Assuming that arm i is not selected from round t_1 to t_2, then $N_i(t_2+1) = n_i$ and $\hat{\mu}_i(t_2+1) = \hat{\mu}_i(t_1)$. Applying the concentration bounds from Lemmas A.1 and A.2, the sampled value $\nu_i(t_2+1)$ for arm i satisfies:

$$\begin{aligned}
\nu_i(t_2+1) &< \hat{\mu}_i(t_2+1) + \gamma(t_2+1) \\
&\le \hat{\mu}_K(t_1) - 2\beta(N_K(t_1)) - \sqrt{8\log\frac{\pi^2 K}{3\delta}} - 4\sqrt{n_i}\,\delta_0 + \gamma(t_2+1) \\
&\le \hat{\mu}_K(t_2+1) - \sqrt{8\log\frac{\pi^2 K}{3\delta}} - 4\sqrt{n_i}\,\delta_0 + \gamma(t_2+1) \\
&\le \hat{\mu}_K(t_2+1) - \sqrt{8\log\frac{\pi^2 K}{3\delta} + 16 n_i\delta_0^2} + \gamma(t_2+1) \\
&= \hat{\mu}_K(t_2+1) - \sqrt{8\log\frac{\pi^2 K t^2}{3\delta}} + \gamma(t_2+1) \\
&\le \hat{\mu}_K(t_2+1) - \gamma(t_2+1) \\
&< \nu_K(t_2+1),
\end{aligned}$$

where the last inequality uses the fact that $\gamma(t_2+1)$ is an upper confidence width and $\nu_K(t_2+1) > \hat{\mu}_K(t_2+1) - \gamma(t_2+1)$ with high probability.

This chain of inequalities implies that the sampled value $\nu_i(t_2+1)$ remains lower than $\nu_K(t_2+1)$ for all $t_2 < \lfloor \exp(n_i\delta_0^2) \rfloor$, with probability at least $1-2\delta$. Therefore, arm i will not be selected again until at least round $\lfloor \exp(n_i\delta_0^2) \rfloor$. □

A.2.2 Proof of Theorem 4.1

Proof. Assume event $\mathcal{E}$ holds. After a single injection on each non-target arm $i \in [K-1]$, Algorithm 1 ensures that

$$\hat{\mu}_i(t) \le \hat{\mu}_K(t) - 2\beta(N_K(t)) - 3\sigma\delta_0 \quad \text{and} \quad N_i(t) \ge \frac{\log T}{\delta_0^2}.$$

By Lemma 4.1, this guarantees that arm i will not be selected before round T with high probability.

Define $\tilde{n}_i$ as the total number of injected data samples for arm i. Since Algorithm 1 performs a single injection per non-target arm, we have $\tilde{n}_i = 1$ for all $i \in [K-1]$. We now analyze the total attack

cost:

$$\begin{aligned}
\sum_{i=1}^{K-1} C_i^F(T) &= \sum_{i=1}^{K-1} \left(\hat{\mu}_i(t)N_i(t) + \mu_i\tilde{n}_i - \hat{\ell}_K(t)(N_i(t)+\tilde{n}_i)\right) \\
&\leq \sum_{i=1}^{K-1} \left(\left(\mu_i + \beta(N_i(t))\right)N_i(t) + \mu_i - (\hat{\mu}_K - 2\beta(N_K(t)) - 3\sigma\delta_0)(N_i(t)+1)\right) \\
&\leq \sum_{i=1}^{K-1} \left(\left(\mu_i + \beta(N_i(t))\right)N_i(t) + \mu_i - (\mu_K - 3\beta(1) - 3\sigma\delta_0)(N_i(t)+1)\right) \\
&\leq \sum_{i=1}^{K-1} \left(\mu_i - \mu_K + \left(\beta\left(\left\lceil\frac{\log T}{\delta_0^2}\right\rceil\right) + 3\beta(1) + 3\sigma\delta_0\right)\right)\left(\left\lceil\frac{\log T}{\delta_0^2}\right\rceil + 1\right) \\
&\leq \sum_{i=1}^{K-1} \left(\Delta_i + 4\beta(1) + 3\sigma\delta_0\right)\left(\left\lceil\frac{\log T}{\delta_0^2}\right\rceil + 1\right) \\
&= \mathcal{O}\left(\sum_{i=1}^{K-1} \left(\Delta_i + 4\beta(1) + 3\sigma\delta_0\right)\frac{\log T}{\delta_0^2}\right).
\end{aligned}$$

Therefore, both the cumulative attack cost and the number of non-target arm pulls are sublinear in T, completing the proof. □

A.2.3 Proof on Theorem 4.2

Proof. Suppose events $\mathcal{E}$ and $\mathcal{F}$ hold. After a single injection on arm i at round t, we ensure that $N_i(t) \geq \frac{\log T}{\delta_0^2}, \hat{\mu}_i(t) \leq \hat{\ell}'_K(t)$, where $\hat{\ell}'_K(t)$ is the adjusted threshold for suppressing arm i under Thompson sampling. By Lemma A.3, arm i will not be selected again before round T with high probability.

Since Algorithm 4 performs a single injection per non-target arm, we have $\tilde{n}_i = 1$ for all $i \in [K-1]$. We now analyze the total attack cost:

$$\begin{aligned}
\sum_{i=1}^{K-1} C_i^F(T) &= \sum_{i=1}^{K-1} \left(\hat{\mu}_i(t)N_i(t) + \mu_i\tilde{n}_i - \hat{\ell}'_K(t)(N_i(t)+\tilde{n}_i)\right) \\
&\leq \sum_{i=1}^{K-1} \left(\left(\mu_i + \beta(N_i(t))\right)N_i(t) + \mu_i\right) \\
&\quad - \sum_{i=1}^{K-1} \left(\hat{\mu}_K(t) - 2\beta(N_K(t)) - \sqrt{8\log\frac{\pi^2 K}{3\delta}} - 4\sqrt{N_i(t)}\delta_0\right)\left(\left\lceil\frac{\log T}{\delta_0^2}\right\rceil + 1\right) \\
&\leq \sum_{i=1}^{K-1} \left(\left(\mu_i + \beta\left(\left\lceil\frac{\log T}{\delta_0^2}\right\rceil\right)\right)\left\lceil\frac{\log T}{\delta_0^2}\right\rceil + \mu_i\right) \\
&\quad - \sum_{i=1}^{K-1} \left(\mu_K - 3\beta(1) - \sqrt{8\log\frac{\pi^2 K}{3\delta}} - 4\sqrt{\left\lceil\frac{\log T}{\delta_0^2}\right\rceil}\delta_0\right)\left(\left\lceil\frac{\log T}{\delta_0^2}\right\rceil + 1\right) \\
&\leq \sum_{i=1}^{K-1} \left(\mu_i - \mu_K + 4\beta(1) + \sqrt{8\log\frac{\pi^2 K}{3\delta}} + 4\sqrt{\left\lceil\frac{\log T}{\delta_0^2}\right\rceil}\delta_0\right)\left(\left\lceil\frac{\log T}{\delta_0^2}\right\rceil + 1\right) \\
&= \mathcal{O}\left(\sum_{i=1}^{K-1} \left(\Delta_i + 4\beta(1) + \sqrt{8\log\frac{\pi^2 K}{3\delta}} + 4\sqrt{\log T}\right)\frac{\log T}{\delta_0^2}\right).
\end{aligned}$$

Therefore, both the cumulative attack cost and the number of non-target arm pulls are sublinear in T, completing the proof. □

Algorithm 4: Least Injection Algorithm on Thompson sampling

Input: Attack parameter $\delta_0 > 0$

for *round* $t = 1, 2, \dots$ **do**
  for *each non-target arm* $i \in [K-1]$ **do**
    if *arm* i *has not been attacked* ***and*** $N_i(t) = \left\lceil \frac{\log T}{\delta_0^2} \right\rceil$ **then**
      Inject fake data sample:
      $(a_i^F, r_i^F) = \left(i, N_i(t) \cdot \left(\hat{\ell}'_K(t) - \hat{\mu}_i(t)\right) + \hat{\ell}'_K(t)\right)$;
      $t \leftarrow t + 1$;
    end
  end
end

A.3 Proofs of Simultaneous Bounded Injection

A.3.1 Proof of Theorem 4.3

Proof. According to Algorithm 2, after injecting $\tilde{n}_i$ samples with value $\tilde{a}$ into arm i at round t, the empirical mean at round $t + \tilde{n}_i$ becomes:

$$\begin{aligned}
\hat{\mu}_i(t+\tilde{n}_i) &= \frac{\hat{\mu}_i(t)N_i(t) + \tilde{n}_i \cdot \tilde{a}}{N_i(t) + \tilde{n}_i} \\
&\leq \frac{\hat{\mu}_i(t)N_i(t) + \frac{\hat{\mu}_i(t)-\hat{\ell}_K(t)}{\hat{\ell}_K(t)-\tilde{a}} N_i(t) \cdot \tilde{a}}{N_i(t) + \frac{\hat{\mu}_i(t)-\hat{\ell}_K(t)}{\hat{\ell}_K(t)-\tilde{a}} N_i(t)} \\
&= \frac{\hat{\mu}_i(t) + \frac{\hat{\mu}_i(t)-\hat{\ell}_K(t)}{\hat{\ell}_K(t)-\tilde{a}} \tilde{a}}{1 + \frac{\hat{\mu}_i(t)-\hat{\ell}_K(t)}{\hat{\ell}_K(t)-\tilde{a}}} \\
&= \frac{\hat{\mu}_i(t)\hat{\ell}_K(t) - \hat{\ell}_K(t)\tilde{a}}{\hat{\mu}_i(t) - \tilde{a}} \\
&\leq \hat{\ell}_K(t),
\end{aligned}$$

where the last step ensures that after injection, arm i's empirical mean is suppressed below the target threshold.

Since event $\mathcal{E}$ holds, and by Lemma 4.1, arm i will not be selected after round $t + \tilde{n}_i$. The total number of pulls for arm i is therefore:

$$\begin{aligned}
N_i(t) + \tilde{n}_i &\leq \left\lceil \frac{\log T}{\delta_0^2} \right\rceil + \frac{\hat{\mu}_i(t) - \hat{\ell}_K(t)}{\hat{\ell}_K(t) - \tilde{a}} \cdot \left\lceil \frac{\log T}{\delta_0^2} \right\rceil \\
&= \frac{\hat{\mu}_i(t) - \tilde{a}}{\hat{\ell}_K(t) - \tilde{a}} \cdot \left\lceil \frac{\log T}{\delta_0^2} \right\rceil \\
&\leq \frac{\mu_i + \beta(N_i(t)) - \tilde{a}}{\hat{\mu}_K(t) - 2\beta(N_K(t)) - 3\sigma\delta_0 - \tilde{a}} \cdot \left\lceil \frac{\log T}{\delta_0^2} \right\rceil \\
&\leq \frac{\mu_i + \beta\left(\left\lceil \frac{\log T}{\delta_0^2} \right\rceil\right) - \tilde{a}}{\mu_K - 3\beta(1) - 3\sigma\delta_0 - \tilde{a}} \cdot \left\lceil \frac{\log T}{\delta_0^2} \right\rceil \\
&= \mathcal{O}\left(\frac{\mu_i + \beta\left(\frac{\log T}{\delta_0^2}\right) - \tilde{a}}{\mu_K - 3\beta(1) - 3\sigma\delta_0 - \tilde{a}} \cdot \frac{\log T}{\delta_0^2} \right),
\end{aligned}$$

where we used concentration bounds for $\hat{\mu}_i(t)$ and $\hat{\mu}_K(t)$ under event $\mathcal{E}$ and the non-decreasing property of $\beta(\cdot)$.

The total attack cost can be calculated as:

$$
\begin{aligned}
\sum_{i=1}^{K-1} C_i^F(T) &= \sum_{i=1}^{K-1} \left(\hat{\mu}_i(t) N_i(t) + \mu_i \tilde{n}_i - \hat{\ell}_K(t)(N_i(t) + \tilde{n}_i) \right) \\
&\leq \sum_{i=1}^{K-1} \left(\beta(N_i(t)) N_i(t) + \mu_i (N_i(t) + \tilde{n}) \right) \\
&\quad - \sum_{i=1}^{K-1} (\mu_K - 3\beta(N_K(t)) - 3\sigma\delta_0)(N_i(t) + \tilde{n}) \\
&= \sum_{i=1}^{K-1} \left(\beta\left(\left\lceil \frac{\log T}{\delta_0^2} \right\rceil \right) \left\lceil \frac{\log T}{\delta_0^2} \right\rceil + (\mu_i - \mu_K + 3\beta(1) + 3\sigma\delta_0)(N_i(t) + \tilde{n}) \right) \\
&\leq \sum_{i=1}^{K-1} \left((\mu_i - \tilde{a}) \frac{\Delta_i + \beta(\lceil (\log T)/\delta_0^2 \rceil) + 3\beta(1) + 3\sigma\delta_0}{\mu_K - 3\beta(1) - 3\sigma\delta_0 - \tilde{a}} \right) \left\lceil \frac{\log T}{\delta_0^2} \right\rceil \\
&\leq \sum_{i=1}^{K-1} \left((\mu_i - \tilde{a}) \frac{\Delta_i + 4\beta(1) + 3\sigma\delta_0}{\mu_K - 3\beta(1) - 3\sigma\delta_0 - \tilde{a}} \right) \left\lceil \frac{\log T}{\delta_0^2} \right\rceil \\
&= \mathcal{O}\left(\sum_{i=1}^{K-1} \left((\mu_i - \tilde{a}) \frac{\Delta_i + 4\beta(1) + 3\sigma\delta_0}{\mu_K - 3\beta(1) - 3\sigma\delta_0 - \tilde{a}} \right) \frac{\log T}{\delta_0^2} \right).
\end{aligned}
$$

Therefore, both the cumulative attack cost and the number of non-target arm pulls are $\mathcal{O}\left(\frac{\log T}{\delta_0^2}\right)$ per arm, and hence sublinear in T, completing the proof. □

A.3.2 Proof of Theorem 4.4

Proof. According to Algorithm 5, after injecting $\tilde{n}_i$ samples with value $\tilde{a}$ into arm i at round t, the empirical mean at round $t + \tilde{n}_i$ becomes:

$$
\begin{aligned}
\hat{\mu}_i(t + \tilde{n}_i) &= \frac{\hat{\mu}_i(t) N_i(t) + \tilde{n}_i \tilde{a}}{N_i(t) + \tilde{n}_i} \\
&\leq \frac{\hat{\mu}_i(t) N_i(t) + \frac{\hat{\mu}_i(t) - \hat{\ell}'_K(t)}{\hat{\ell}'_K(t) - \tilde{a}} N_i(t) \tilde{a}}{N_i(t) + \frac{\hat{\mu}_i(t) - \hat{\ell}'_K(t)}{\hat{\ell}'_K(t) - \tilde{a}} N_i(t)} \\
&= \frac{\hat{\mu}_i(t) + \frac{\hat{\mu}_i(t) - \hat{\ell}'_K(t)}{\hat{\ell}'_K(t) - \tilde{a}} \tilde{a}}{1 + \frac{\hat{\mu}_i(t) - \hat{\ell}'_K(t)}{\hat{\ell}'_K(t) - \tilde{a}}} \\
&= \frac{\hat{\mu}_i(t) \hat{\ell}'_K(t) - \hat{\ell}'_K(t) \tilde{a}}{\hat{\mu}_i(t) - \tilde{a}} \\
&\leq \hat{\ell}'_K(t + n_i),
\end{aligned}
$$

where the last step ensures that after injection, arm i's empirical mean is suppressed below the target threshold.

Since events $\mathcal{E}$ and $\mathcal{F}$ hold, and by Lemma A.3, arm i will not be selected after round $t + \tilde{n}_i$. The total number of pulls for arm i is therefore:

$$
\begin{aligned}
N_i(t)+\tilde{n}_i &= \left\lceil \frac{\log T}{\delta_0^2} \right\rceil + \frac{\hat{\mu}_i(t) - \hat{\ell}'_K(t)}{\hat{\ell}'_K(t) - \tilde{a}} \left\lceil \frac{\log T}{\delta_0^2} \right\rceil \\
&= \frac{\hat{\mu}_i(t) - \tilde{a}}{\hat{\ell}'_K(t) - \tilde{a}} \left\lceil \frac{\log T}{\delta_0^2} \right\rceil \\
&\leq \frac{\mu_i + \beta(N_i(t)) - \tilde{a}}{\hat{\mu}_K(t) - 2\beta(N_K(t)) - \sqrt{8\log\frac{\pi^2 K}{3\delta}} - 4\sqrt{N_i(t)}\delta_0 - \tilde{a}} \left\lceil \frac{\log T}{\delta_0^2} \right\rceil \\
&\leq \frac{\mu_i + \beta(\lceil (\log T)/\delta_0^2 \rceil) - \tilde{a}}{\mu_K - 3\beta(1) - \sqrt{8\log\frac{\pi^2 K}{3\delta}} - 4\sqrt{\lceil (\log T)/\delta_0^2 \rceil}\delta_0 - \tilde{a}} \left\lceil \frac{\log T}{\delta_0^2} \right\rceil \\
&= \mathcal{O}\left(\frac{\mu_i + \beta((\log T)/\delta_0^2) - \tilde{a}}{\mu_K - 3\beta(1) - \sqrt{8\log\frac{\pi^2 K}{3\delta}} - 4\sqrt{\log T} - \tilde{a}} \frac{\log T}{\delta_0^2} \right).
\end{aligned}
$$

Let $\hat{i} = \mu_i + \beta\left(\left\lceil \frac{\log T}{\delta_0^2} \right\rceil\right) - \tilde{a}$ and $\hat{k} = \mu_K - 3\beta(1) - \sqrt{8\log\frac{\pi^2 K}{3\delta}} - 4\sqrt{N_i(t)}\delta_0 - \tilde{a}$. The total attack cost can be calculated as:

$$
\begin{aligned}
\sum_{i=1}^{K-1} C_i^F(T) &= \sum_{i=1}^{K-1} \left(\hat{\mu}_i(t) N_i(t) + \mu_i \tilde{n}_i - \hat{\ell}'_K(t)(N_i(t) + \tilde{n}_i) \right) \\
&\leq \sum_{i=1}^{K-1} \left(\mu_i (N_i(t) + \tilde{n}) + \beta(N_i(t)) N_i(t) \right) \\
&\quad - \sum_{i=1}^{K-1} \left(\hat{\mu}_K(t) - 2\beta(N_K(t)) - \sqrt{8\log\frac{\pi^2 K}{3\delta}} - 4\sqrt{N_i(t)}\delta_0 \right) (N_i(t) + \tilde{n}) \\
&\leq \sum_{i=1}^{K-1} \left(\beta\left(\left\lceil \frac{\log T}{\delta_0^2} \right\rceil\right) \left\lceil \frac{\log T}{\delta_0^2} \right\rceil + \mu_i \left(\frac{\hat{i}}{\tilde{k}} \left\lceil \frac{\log T}{\delta_0^2} \right\rceil \right) \right) \\
&\quad - \sum_{i=1}^{K-1} \left(\mu_K - 3\beta(1) - \sqrt{8\log\frac{\pi^2 K}{3\delta}} - 4\sqrt{N_i(t)}\delta_0 - \tilde{a} + \tilde{a} \right) \left(\frac{\tilde{i}}{\tilde{k}} \left\lceil \frac{\log T}{\delta_0^2} \right\rceil \right) \\
&= \sum_{i=1}^{K-1} \left(\beta\left(\left\lceil \frac{\log T}{\delta_0^2} \right\rceil\right) \left\lceil \frac{\log T}{\delta_0^2} \right\rceil + (\mu_i - \tilde{a}) \left(\frac{\tilde{i}}{\tilde{k}} \left\lceil \frac{\log T}{\delta_0^2} \right\rceil \right) \right) - \sum_{i=1}^{K-1} \left(\tilde{i} \left\lceil \frac{\log T}{\delta_0^2} \right\rceil \right) \\
&= \sum_{i=1}^{K-1} \left((\mu_i - \tilde{a}) \frac{\tilde{i}}{\tilde{k}} - \tilde{i} + \beta\left(\left\lceil \frac{\log T}{\delta_0^2} \right\rceil\right) \right) \left\lceil \frac{\log T}{\delta_0^2} \right\rceil \\
&= \sum_{i=1}^{K-1} \left((\mu_i - \tilde{a}) \frac{\tilde{i}}{\tilde{k}} - (\mu_i - \tilde{a}) \right) \left\lceil \frac{\log T}{\delta_0^2} \right\rceil \\
&= \sum_{i=1}^{K-1} \left((\mu_i - \tilde{a}) \frac{\Delta_i + \beta\left(\left\lceil \frac{\log T}{\delta_0^2} \right\rceil\right) + 3\beta(1) + \sqrt{8\log\frac{\pi^2 K}{3\delta}} + 4\sqrt{N_i(t)}\delta_0}{\mu_K - 3\beta(1) - \sqrt{8\log\frac{\pi^2 K}{3\delta}} - 4\sqrt{N_i(t)}\delta_0 - \tilde{a}} \right) \left\lceil \frac{\log T}{\delta_0^2} \right\rceil \\
&\leq \sum_{i=1}^{K-1} \left((\mu_i - \tilde{a}) \frac{\Delta_i + 4\beta(1) + \sqrt{8\log\frac{\pi^2 K}{3\delta}} + 4\sqrt{\left\lceil \frac{\log T}{\delta_0^2} \right\rceil}\delta_0}{\mu_K - 3\beta(1) - \sqrt{8\log\frac{\pi^2 K}{3\delta}} - 4\sqrt{\left\lceil \frac{\log T}{\delta_0^2} \right\rceil}\delta_0 - \tilde{a}} \right) \left\lceil \frac{\log T}{\delta_0^2} \right\rceil \\
&= \mathcal{O}\left(\sum_{i=1}^{K-1} \left((\mu_i - \tilde{a}) \frac{\Delta_i + 4\beta(1) + \sqrt{8\log\frac{\pi^2 K}{3\delta}} + 4\sqrt{\log T}}{\mu_K - 3\beta(1) - \sqrt{8\log\frac{\pi^2 K}{3\delta}} - 4\sqrt{\log T} - \tilde{a}} \right) \frac{\log T}{\delta_0^2} \right).
\end{aligned}
$$

□

Therefore, both the cumulative attack cost and the number of non-target arm pulls are $\mathcal{O}\left(\frac{\log T}{\delta_0^2}\right)$ per arm, and hence sublinear in T, completing the proof.

Algorithm 5: Simultaneous Bounded Injection Attack on Thompson Sampling

Input: Number of real users R, target arm K, parameter δ_0, lower bound $\tilde{a}$

for *round* $t = 1, 2, \ldots$ **do**
  for *each non-target arm* $i \in [K-1]$ **do**
    if *arm* i *has not been attacked* ***and*** $N_i(t) = \left\lceil \frac{\log T}{\delta_0^2} \right\rceil$ **then**
      $\tilde{n} \leftarrow \left\lceil \frac{\hat{\mu}_i(t) - \hat{\ell}'_K(t)}{\hat{\ell}'_K(t) - \tilde{a}} \cdot \left\lceil \frac{\log T}{\delta_0^2} \right\rceil \right\rceil$;
      Inject $\tilde{n}$ fake samples $(i, \tilde{a})$;
      $t \leftarrow t + \tilde{n}$
    end
  end
end

A.4 Proofs of Periodic Bounded Injection

A.4.1 Proof of Lemma 4.2

Proof. Suppose event $\mathcal{E}$ holds. We begin by estimating the total number of fake samples needed to demote arm i. This quantity is given by

$$\tilde{n}_i = \left\lceil \frac{\hat{\mu}_i(t) - \hat{\ell}_K(t)}{\hat{\ell}_K(t) - \tilde{a}} \cdot \left\lceil \frac{\log T}{\delta_0^2} \right\rceil \right\rceil,$$

where $\hat{\ell}_K(t)$ is a conservative lower bound on arm K's empirical mean, and $\tilde{a}$ is the value of each injected fake sample. Under the batch size constraint f, the attack spans $\left\lceil \frac{\tilde{n}_i}{f} \right\rceil$ periods, with each period injecting f fake samples.

Our goal is to choose an appropriate delay parameter R_i such that after each injection, arm i is not selected again until the next scheduled injection. Specifically, we require that after the c-th batch (for any $c \in \{1, 2, \ldots, \lceil \frac{\tilde{n}_i}{f} \rceil\}$), arm i is not selected for at least R_i rounds.

Let $t_i(c) = t + (f + R_i)c$ denote the round before the $c+1$-th injection. We examine the UCB index of arm i at time $t_i(c)$:

$$\begin{aligned} \mathrm{UCB}_i(t_i(c)) &= \hat{\mu}_i(t_i(c)) + 3\sigma\sqrt{\frac{\log t_i(c)}{N_i(t) + fc}} \\ &= \frac{N_i(t)\hat{\mu}_i(t) + fc\tilde{a}}{N_i(t) + fc} + 3\sigma\sqrt{\frac{\log t_i(c)}{N_i(t) + fc}}. \end{aligned} \tag{10}$$

To ensure arm i is not selected before round $t_i(c)$, we want its UCB index to be no larger than that of arm K. A sufficient condition is

$$\mathrm{UCB}_i(t_i(c)) \le \hat{\mu}_K(t) - 2\beta(N_K(t)) \le \mathrm{UCB}_K(t_i(c)), \tag{11}$$

where we use the lower bound $\hat{\mu}_K(t) - 2\beta(N_K(t))$ to conservatively approximate arm K's UCB.

Define $\tilde{\mu}_i(t_i(c))$ as the post-injection empirical mean of arm i:

$$\tilde{\mu}_i(t_i(c)) = \frac{N_i(t)\hat{\mu}_i(t) + fc\tilde{a}}{N_i(t) + fc}.$$

Then, the condition in (11) implies that, for any c, the delay parameter $R_i(c)$ must satisfy:

$$R_i(c) \leq \frac{\exp\left(\left(\frac{\hat{\mu}_K(t)-2\beta(N_K(t))-\tilde{\mu}_i(t_i(c))}{3\sigma}\right)^2 \cdot (N_i(t)+fc)\right) - t}{c} - f. \tag{12}$$

Finally, to ensure that this condition holds for every injection period, we define the overall delay parameter as the minimum over all c:

$$R_i = \min_{1 \leq c \leq \lceil \frac{\tilde{n}_i}{f} \rceil} \frac{\exp\left(\left(\frac{\hat{\mu}_K(t)-2\beta(N_K(t))-\tilde{\mu}_i(t_i(c))}{3\sigma}\right)^2 \cdot (N_i(t)+fc)\right) - t}{c} - f. \tag{13}$$

This choice of R_i ensures that after each batch of f fake samples, arm i will not be pulled again until the next scheduled injection. □

Discussion of Algorithm 3.

Based on our experimental observations, we find that $c = 1$ typically yields the smallest value of $R_i(c)$ in practice. We provide the following sufficient condition under which $R_i(c = 1)$ is guaranteed to be the minimizer:

$$\frac{\hat{\mu}_K(t) - 2\beta(N_K(t)) - \tilde{\mu}_i(t_i(1))}{3\sigma} > 1.$$

Proof. We simplify the expression for $R_i(c)$ in (12) as follows:

$$R_i(c) = \frac{\exp\big(h(c)g(c)\big) - t}{c} - f,$$

where

$$h(c) = \left(\frac{\hat{\mu}_K(t) - 2\beta(N_K(t)) - \tilde{\mu}_i(t_i(c))}{3\sigma}\right)^2, \quad g(c) = N_i(t) + fc.$$

Note that $h(c) > 0$ and is non-decreasing in c (i.e., $h'(c) \geq 0$), while $g(c)$ is clearly increasing in c.

We now examine the derivative of $R_i(c)$:

$$\begin{aligned}\frac{d}{dc}R_i(c) &= \frac{d}{dc}\left(\frac{\exp(h(c)g(c)) - t}{c}\right) \\ &= \frac{\exp(h(c)g(c))}{c^2}\left[h(c)g(c)\left(h'(c)g(c) + h(c)g'(c)\right) - 1\right] + \frac{t}{c^2}.\end{aligned}$$

Since $g'(c) = f$ and $g(c) = N_i(t) + fc$, we further bound this as:

$$\begin{aligned}\frac{d}{dc}R_i(c) &\geq \frac{\exp(h(c)g(c))}{c^2}\left[h^2(c)g(c)g'(c) - 1\right] \\ &= \frac{\exp(h(c)g(c))}{c^2}\left[h^2(c)f(N_i(t) + fc) - 1\right] \\ &\geq \frac{\exp(h(c)g(c))}{c^2}\left[h^2(1)f(N_i(t) + f) - 1\right].\end{aligned}$$

Therefore, if $h(1) > 1$, then $h^2(1)f(N_i(t) + f) > 1$, which ensures that the derivative is strictly positive for all $c \geq 1$. This implies that $R_i(c)$ is strictly increasing in c, and thus $R_i(1)$ is the minimizer. □

A.5 Proof of Periodic Bounded Injection on Thompson Sampling

Algorithm 6: Periodic Bounded Injection on Thompson Sampling

Input: Attack parameter δ_0, reward bound $[\tilde{a}, \tilde{b}]$, max batch size f

for *round* $t = 1, 2, \dots$ **do**
  for *each non-target arm* $i \in [K-1]$ **do**
    if *arm* i *has not been attacked* ***and*** $N_i(t) = \left\lceil \frac{\log T}{\delta_0^2} \right\rceil$ **then**
      $\tilde{n}_i \leftarrow \left\lceil \frac{\hat{\mu}_i(t) - \hat{\ell}'_K(t)}{\hat{\ell}'_K(t) - \tilde{a}} \cdot \left\lceil \frac{\log T}{\delta_0^2} \right\rceil \right\rceil$;
      $R_i = \min_{1 \le c \le \lceil \frac{\tilde{n}_i}{f} \rceil} \left\{ \frac{1}{c} \left(\sqrt{\frac{3\delta}{\pi^2 K} \cdot \exp\left(\frac{(\hat{\mu}_K(t) - 2\beta(N_K(t)) - \tilde{\mu}_i(t_i(c)))^2}{8} \right)} - t \right) - f \right\}$;
      $\text{next}_i \leftarrow t$;
    end
    if $\tilde{n}_i > 0$ ***and*** *next*$_i \le t$ **then**
      Inject f fake samples $(i, \tilde{a})$;
      $\tilde{n}_i \leftarrow \tilde{n}_i - f$;
      $t \leftarrow t + f$;
      $\text{next}_i \leftarrow \text{next}_i + f + R_i$;
    end
  end
end

Lemma A.4. *The choice of R_i in Algorithm 6 ensures that once a batch of f fake data samples is injected into non-target arm i, the arm will not be selected again for at least the next R_i rounds.*

Proof. We aim to guarantee that arm i is not selected between successive fake sample injections. Let $\tilde{\nu}_i$ and $\tilde{\nu}_K$ denote the Thompson-sampled values of arm i and arm K, respectively, after the c-th injection. Let $t_i(c) = t + (f + R_i)c$ denote the round before the $c+1$-th injection.

After injecting f fake samples with value $\tilde{a}$ for c periods, the empirical mean of arm i becomes

$$\tilde{\mu}_i(t_i(c)) = \frac{N_i(t)\hat{\mu}_i(t) + fc\tilde{a}}{N_i(t) + fc}.$$

We want to ensure that arm i is unlikely to be selected before time $t_i(c)$ by ensuring:

$$\begin{aligned} \tilde{\nu}_i(t_i(c)) &\le \tilde{\mu}_i(t_i(c)) + \gamma(t_i(c)) \\ &\le \hat{\mu}_K(t) - 2\beta(N_K(t)) - \gamma(t_i(c)) \le \tilde{\nu}_K(t_i(c)), \end{aligned} \tag{14}$$

where $\gamma(t) = \sqrt{8 \log\left(\frac{\pi^2 K t^2}{3\delta}\right)}$ bounds the Thompson sampling deviation under event $\mathcal{F}$.

Rearranging the middle inequality in (14), we require:

$$\gamma(t_i(c)) \le \hat{\mu}_K(t) - 2\beta(N_K(t)) - \tilde{\mu}_i(t_i(c)). \tag{15}$$

Solving (15) for $t_i(c)$ gives:

$$t_i(c) \le \sqrt{\frac{3\delta}{\pi^2 K} \cdot \exp\left(\frac{1}{8} \left(\hat{\mu}_K(t) - 2\beta(N_K(t)) - \tilde{\mu}_i(t_i(c)) \right)^2 \right)}.$$

Therefore, we define:

$$R_i = \min_{1 \le c \le \lceil \frac{\tilde{n}_i}{f} \rceil} \frac{\sqrt{\frac{3\delta}{\pi^2 K} \cdot \exp\left(\frac{(\hat{\mu}_K(t) - 2\beta(N_K(t)) - \tilde{\mu}_i(t_i(c)))^2}{8} \right)} - t}{c} - f. \tag{16}$$

This ensures that after each batch of f fake samples, arm i is suppressed for at least R_i rounds under events $\mathcal{E}$ and $\mathcal{F}$. Hence, the learner will not select arm i between consecutive injections. □

B Additional Experiments

B.1 Experimental Setup

Our attack strategies were evaluated on both synthetic data and real-world user-item interaction data derived from the MovieLens dataset [16]. We considered a 10-armed stochastic bandit setup for for all experiments.

For the synthetic setting, each arm's reward distribution was modeled as a Gaussian with mean in the range $[0, 1]$ and fixed standard deviation $\sigma = 1$. We designated the arm with the lowest mean as the target arm to be attacked. Simulations were run for up to 10^6 rounds using either the UCB algorithm or Thompson sampling, with our attack strategies applied at predefined intervals.

For the real-world experiments, we used the MovieLens 25M dataset. Ratings were binarized into a sparse user-item interaction matrix, where each entry indicates whether a user interacted with a movie. We then extracted a submatrix comprising the 1000 most active users and the 1000 most interacted-with movies. In each trial, 10 movies were randomly selected as arms, with the movie having the fewest interactions chosen as the target arm. The reward of each arm was defined as the average interaction rate (i.e., the mean of the corresponding binary column). This setup provides a realistic approximation of reward feedback in a recommender system, enabling us to evaluate the attack algorithms in a practical and data-driven context.

We measured the effectiveness of the attacks by tracking the cumulative pull ratio of the target arm over T rounds. To assess both robustness and cost-efficiency, we further analyzed how the total attack cost varies with respect to different values of δ_0 and the time horizon T.

B.2 Attacks on Thompson Sampling

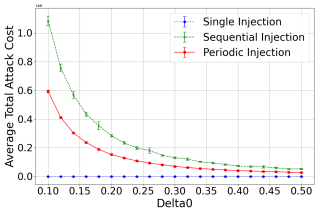

Figure 4: Attack Costs vs δ_0

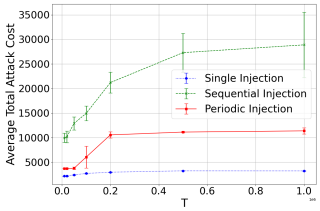

Figure 5: Attack Costs vs T

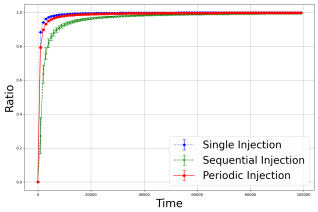

Figure 6: Target Arm Selection Ratios

We evaluate our attack strategies in a simulated environment using synthetic data. Specifically, we consider a multi-armed bandit setting with $K = 10$ arms, whose mean rewards follow a descending sequence $\{0.9, 0.85, \ldots, 0.45\}$ to ensure clear arm differentiation. We compare the performance of three attack methods—single injection, SBI, and PBI—against a Thompson Sampling learner. The time horizon is set to $T = 100{,}000$ steps, and the per-round injection cap for PBI is fixed at $f = 10$. To understand the trade-off between attack cost and effectiveness, we systematically vary both the confidence parameter δ_0 and the time horizon T.

Figure 4 illustrates how the average total attack cost varies with the confidence parameter δ_0. As δ_0 increases, the statistical threshold for suppressing non-target arms becomes more lenient, resulting in significantly fewer fake data injections. Consequently, both the SBI and PBI strategies exhibit a marked reduction in cost, while the Single Injection maintains a consistently low cost due to its one-shot nature. Figure 5 examines the total attack cost as a function of the time horizon T. The cost of SBI continues to grow with T, whereas PBI flattens out, highlighting its efficiency in long-term scenarios. As expected, the Single Injection strategy incurs the lowest cost overall. Figure 6 tracks the target arm selection ratio over time. All three strategies successfully induce the learner to select the target arm in over 95% of rounds after approximately 20,000 steps and maintain this dominance throughout the remaining horizon. These results confirm that all proposed strategies are effective in attacking Thompson sampling, with PBI offering a favorable balance between cost and control.